

CarND-Vehicle-Detection-Copy9

July 15, 2018

1 Vehicle Detection and Tracking

In this project, your goal is to write a software pipeline to detect vehicles in a video (start with the test_video.mp4 and later implement on full project_video.mp4), but the main output or product we want you to create is a detailed writeup of the project.

1.1 The Project

The goals / steps of this project are the following:

- Perform a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set of images and train a classifier Linear SVM classifier
- Optionally, you can also apply a color transform and append binned color features, as well as histograms of color, to your HOG feature vector.
- Note: for those first two steps don't forget to normalize your features and randomize a selection for training and testing.
- Implement a sliding-window technique and use your trained classifier to search for vehicles in images.
- Run your pipeline on a video stream (start with the test_video.mp4 and later implement on full project_video.mp4) and create a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles.
- Estimate a bounding box for vehicles detected.

In [1]: # All imports necessary for the project.

```
import cv2
import time
import glob
import math
import pickle
import random
import string
import platform
import pixiedust
import numpy as np
```

```

import multiprocessing as mp
from skimage.feature import hog
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from sklearn.svm import LinearSVC
from mpl_toolkits.mplot3d import Axes3D
from scipy.ndimage.measurements import label
from sklearn.preprocessing import StandardScaler
# NOTE: the next import is only valid for scikit-learn version <= 0.17
# for scikit-learn >= 0.18 use:
from sklearn.model_selection import train_test_split
# from sklearn.cross_validation import train_test_split

from moviepy.editor import VideoFileClip
from IPython.display import HTML
%matplotlib inline

print(platform.python_version())

Pixiedust database opened successfully

<IPython.core.display.HTML object>

```

3.6.5

1.2 Step 0: Exploration of methods: Drawing boxes

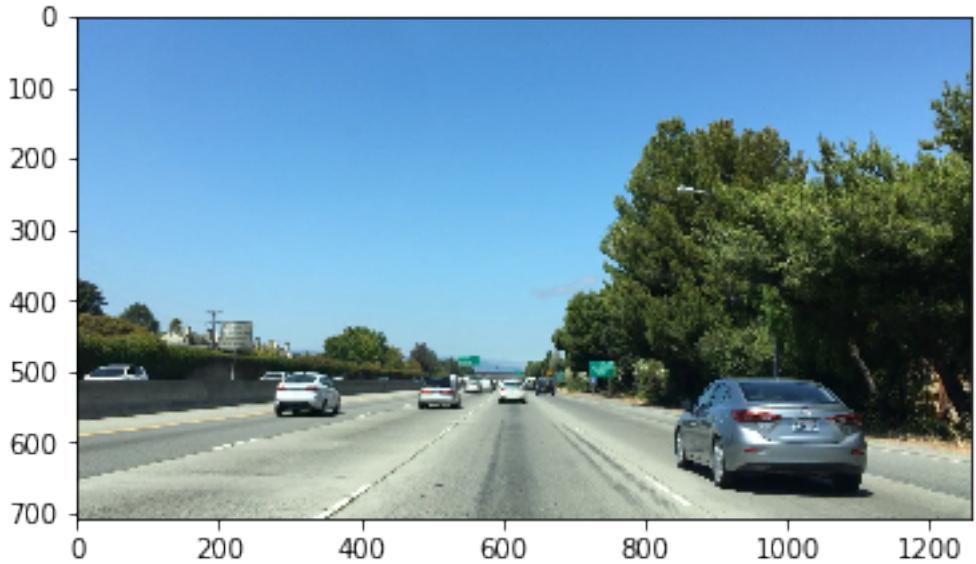
```

In [2]: image = mpimg.imread('/home/gvenkat/Downloads/bbox-example-image.jpg')

In [3]: plt.imshow(image)

Out[3]: <matplotlib.image.AxesImage at 0x7f912714f390>

```



```
In [4]: # Define a function that takes an image, a list of bounding boxes,
# and optional color tuple and line thickness as inputs
# then draws boxes in that color on the output

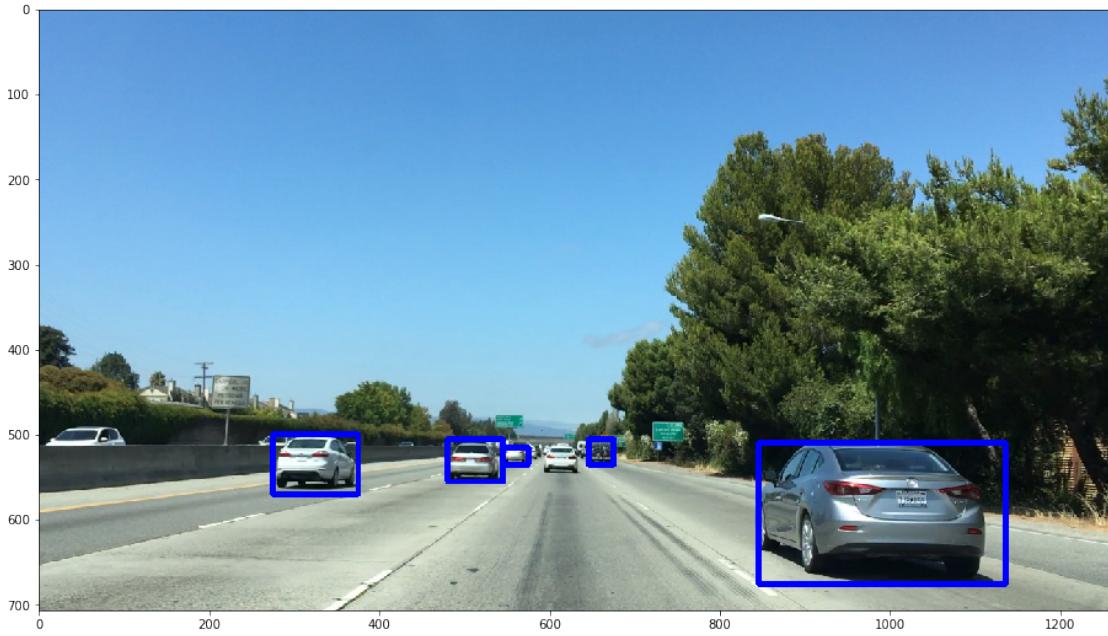
def draw_boxes(img, bboxes, color=(0, 0, 255), thick=6):
    # make a copy of the image
    draw_img = np.copy(img)
    # draw each bounding box on your image copy using cv2.rectangle()
    for box in bboxes:
        draw_img = cv2.rectangle(draw_img, (box[0][0], box[0][1]), (box[1][0], box[1][1]),
                               color, thick)
    # return the image copy with boxes drawn
    return draw_img # Change this line to return image copy with boxes

In [5]: # Add bounding boxes in this format, these are just example coordinates.
bboxes = [((275, 500), (375, 570)), ((845, 510), (1135, 675)), ((480, 505), (545, 555)),

result = draw_boxes(image, bboxes)

plt.figure(figsize=(16, 9))
plt.imshow(result)

Out[5]: <matplotlib.image.AxesImage at 0x7f91270f33c8>
```



```
In [6]: # Define a function that takes an image and a list of templates as inputs
# then searches the image and returns the a list of bounding boxes
# for matched templates
def find_matches(img, template_list):
    # Make a copy of the image to draw on
    # Define an empty list to take bbox coords
    bbox_list = []
    #method = eval('cv2.TM_SQDIFF')
    method = cv2.TM_CCOEFF_NORMED

    # Apply template Matching
    for temp in template_list:
        template = mpimg.imread(temp)
        res = cv2.matchTemplate(img,template,method)
        min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)
        w, h = (template.shape[1], template.shape[0])
        if method in [cv2.TM_SQDIFF, cv2.TM_SQDIFF_NORMED]:
            top_left = min_loc
        else:
            top_left = max_loc
        bottom_right = (top_left[0] + w, top_left[1] + h)
        '''
        print(min_loc)
        print(max_loc)
        '''

        bbox_list.append((top_left, bottom_right))
    return bbox_list
```

```
In [7]: image_template_list = glob.glob('/home/gvenkat/Downloads/cutouts/cut*.jpg')
print(image_template_list)

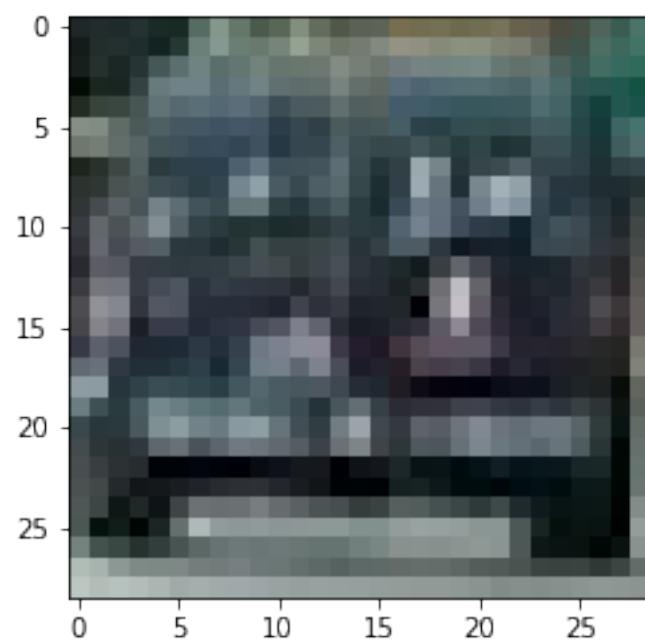
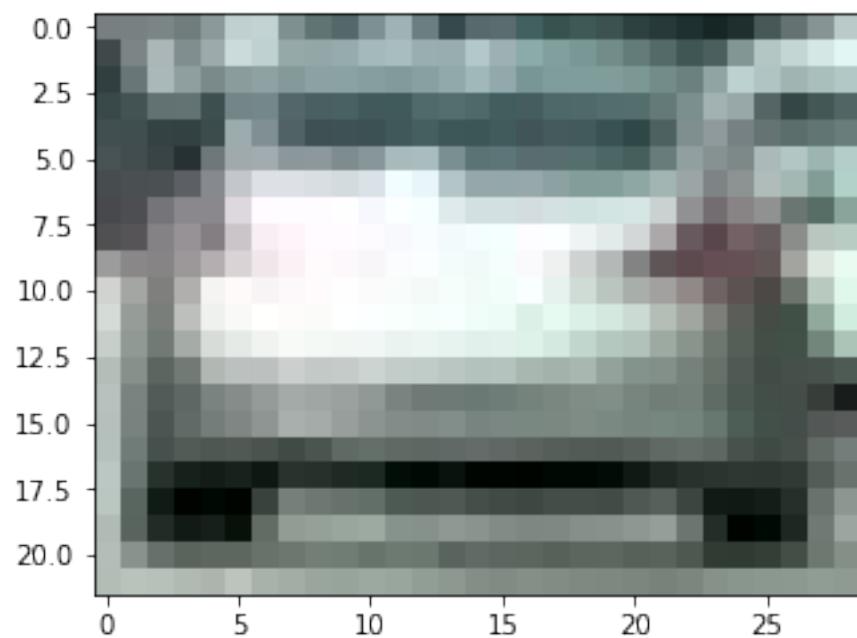
['/home/gvenkat/Downloads/cutouts/cutout2.jpg', '/home/gvenkat/Downloads/cutouts/cutout1.jpg', ...]

In [8]: template_list = []
for image_name in image_template_list:
    img = mpimg.imread(image_name)
    template_list.append(img)

In [9]: for i in range(len(template_list)):
    plt.figure()
    plt.imshow(template_list[i])
```



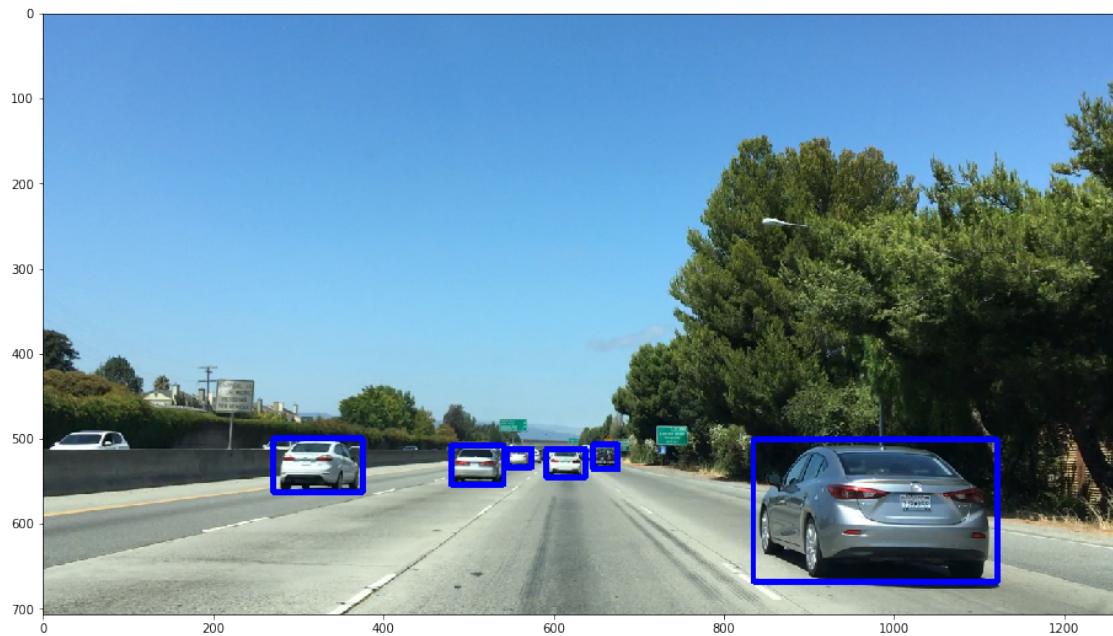






```
In [10]: bboxes = find_matches(image, image_template_list)
         result = draw_boxes(image, bboxes)
         plt.figure(figsize=(16,9))
         plt.imshow(result)
```

```
Out[10]: <matplotlib.image.AxesImage at 0x7f9115b42828>
```



```
In [11]: image2 = mpimg.imread('/home/gvenkat/Downloads/temp-matching-example-2.jpg')

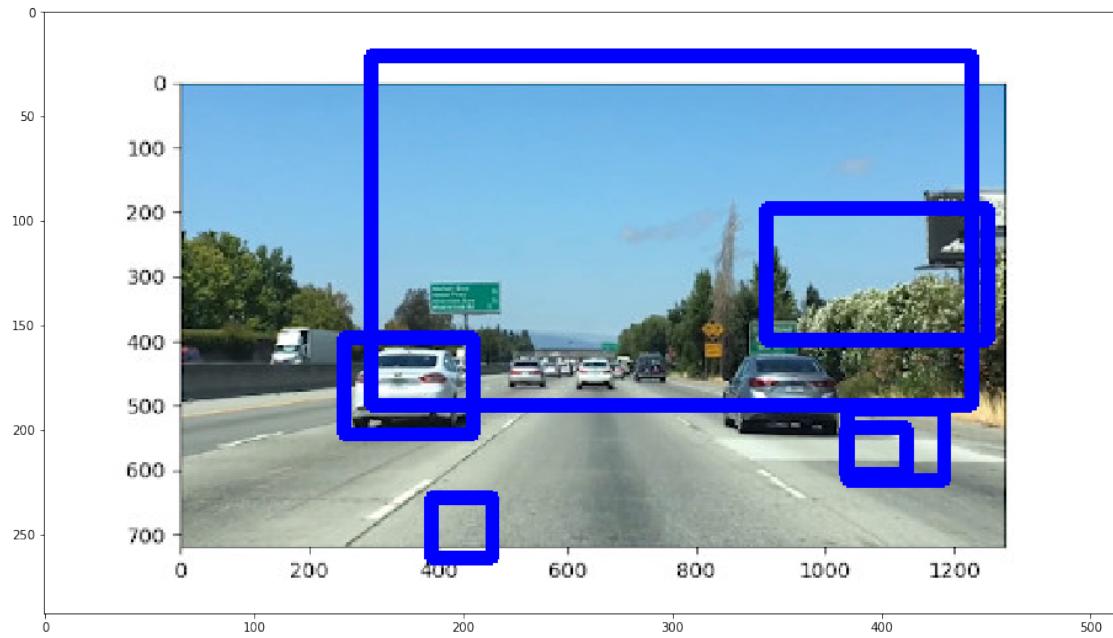
plt.figure(figsize=(16,9))
plt.imshow(image2)
```

```
Out[11]: <matplotlib.image.AxesImage at 0x7f9115b27748>
```



```
In [12]: bboxes = find_matches(image2, image_template_list)
result = draw_boxes(image2, bboxes)
plt.figure(figsize=(16,9))
plt.imshow(result)
```

```
Out[12]: <matplotlib.image.AxesImage at 0x7f9115a7f8d0>
```



```
In [13]: image = mpimg.imread(image_template_list[1])
print(image_template_list[1])
```

```
/home/gvenkat/Downloads/cutouts/cutout1.jpg
```

```
In [14]: plt.figure(figsize=(16,9))
plt.imshow(image)
```

```
Out[14]: <matplotlib.image.AxesImage at 0x7f9115a5d828>
```



1.3 Step 0: Exploration of methods: Histograms

In [15]: # Define a function to compute color histogram features

```
def color_hist(img, nbins=32, bins_range=(0, 256)):
    # Compute the histogram of the RGB channels separately
    rhist = np.histogram(img[:, :, 0], bins = nbins, range = bins_range)
    ghist = np.histogram(img[:, :, 1], bins = nbins, range = bins_range)
    bhist = np.histogram(img[:, :, 2], bins = nbins, range = bins_range)

    # Generating bin centers

    bin_edges = rhist[1]
    bin_centers = (bin_edges[1:] + bin_edges[0:len(bin_edges) - 1])/2

    # Concatenate the histograms into a single feature vector
    hist_features = np.concatenate((rhist[0], ghist[0], bhist[0]))

    # Return the individual histograms, bin_centers and feature vector
    return rhist, ghist, bhist, bin_centers, hist_features
```

In [16]: rh, gh, bh, bincen, feature_vec = color_hist(image, nbins = 32, bins_range=(0, 256))

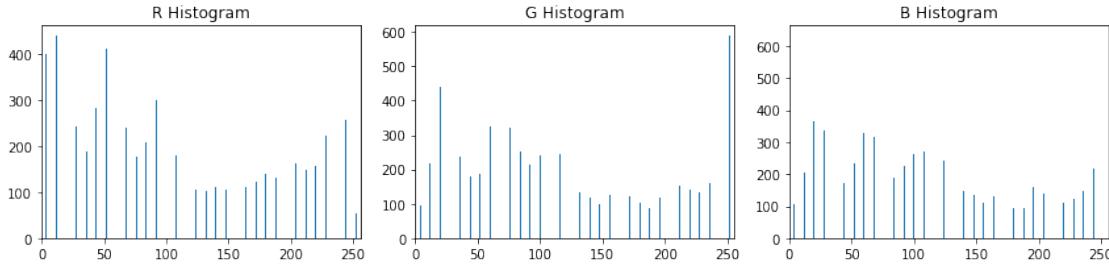
In [17]: # Plot a figure with all three bar charts

```
if rh is not None:
```

```

    fig = plt.figure(figsize=(12,3))
    plt.subplot(131)
    plt.bar(bincen, rh[0])
    plt.xlim(0, 256)
    plt.title('R Histogram')
    plt.subplot(132)
    plt.bar(bincen, gh[0])
    plt.xlim(0, 256)
    plt.title('G Histogram')
    plt.subplot(133)
    plt.bar(bincen, bh[0])
    plt.xlim(0, 256)
    plt.title('B Histogram')
    fig.tight_layout()
else:
    print('Your function is returning None for at least one variable...')


```



In [18]: `image2 = mpimg.imread(image_template_list[0])`

In [19]: `rh, gh, bh, bincen, feature_vec = color_hist(image2, nbins = 32, bins_range=(0, 256))`

In [20]: *# Plot a figure with all three bar charts*

```

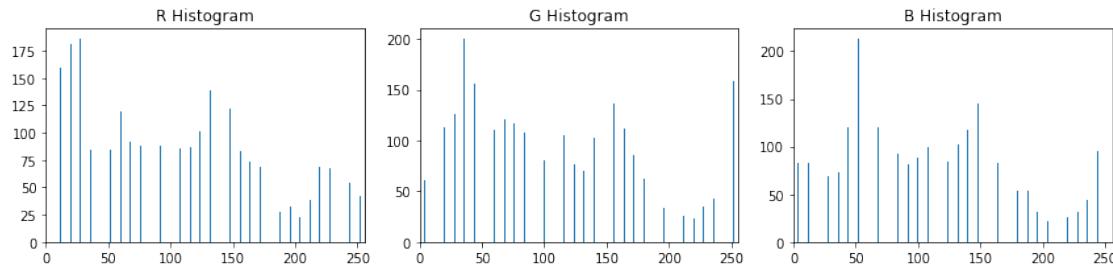
if rh is not None:
    fig = plt.figure(figsize=(12,3))
    plt.subplot(131)
    plt.bar(bincen, rh[0])
    plt.xlim(0, 256)
    plt.title('R Histogram')
    plt.subplot(132)
    plt.bar(bincen, gh[0])
    plt.xlim(0, 256)
    plt.title('G Histogram')
    plt.subplot(133)
    plt.bar(bincen, bh[0])
    plt.xlim(0, 256)
    plt.title('B Histogram')
    fig.tight_layout()


```

```

else:
    print('Your function is returning None for at least one variable...')


```



```

In [21]: def plot3d(pixels, colors_rgb,
                 axis_labels=list("RGB"), axis_limits=((0, 255), (0, 255), (0, 255))):
    """Plot pixels in 3D."""

    # Create figure and 3D axes
    fig = plt.figure(figsize=(8, 8))
    ax = Axes3D(fig)

    # Set axis limits
    ax.set_xlim(*axis_limits[0])
    ax.set_ylim(*axis_limits[1])
    ax.set_zlim(*axis_limits[2])

    # Set axis labels and sizes
    ax.tick_params(axis='both', which='major', labelsize=14, pad=8)
    ax.set_xlabel(axis_labels[0], fontsize=16, labelpad=16)
    ax.set_ylabel(axis_labels[1], fontsize=16, labelpad=16)
    ax.set_zlabel(axis_labels[2], fontsize=16, labelpad=16)

    # Plot pixel values with colors given in colors_rgb
    ax.scatter(
        pixels[:, :, 0].ravel(),
        pixels[:, :, 1].ravel(),
        pixels[:, :, 2].ravel(),
        c=colors_rgb.reshape((-1, 3)), edgecolors='none')

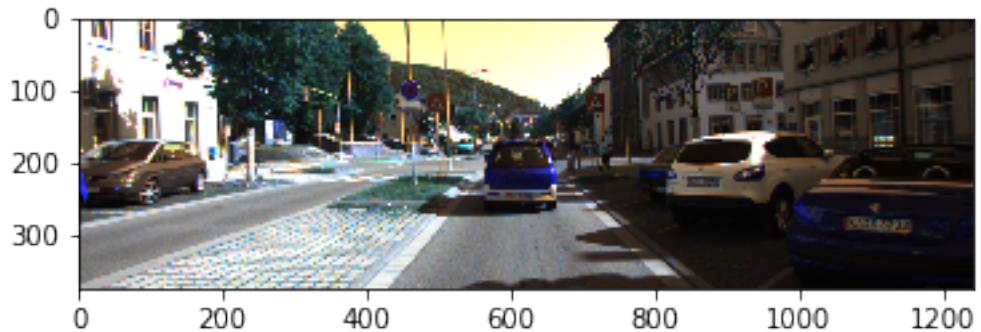
    return ax # return Axes3D object for further manipulation

```

```
In [22]: # Read a color image
```

```
img = cv2.imread("/home/gvenkat/Downloads/000275.png")
plt.imshow(img)
```

```
Out[22]: <matplotlib.image.AxesImage at 0x7f9126e87a58>
```

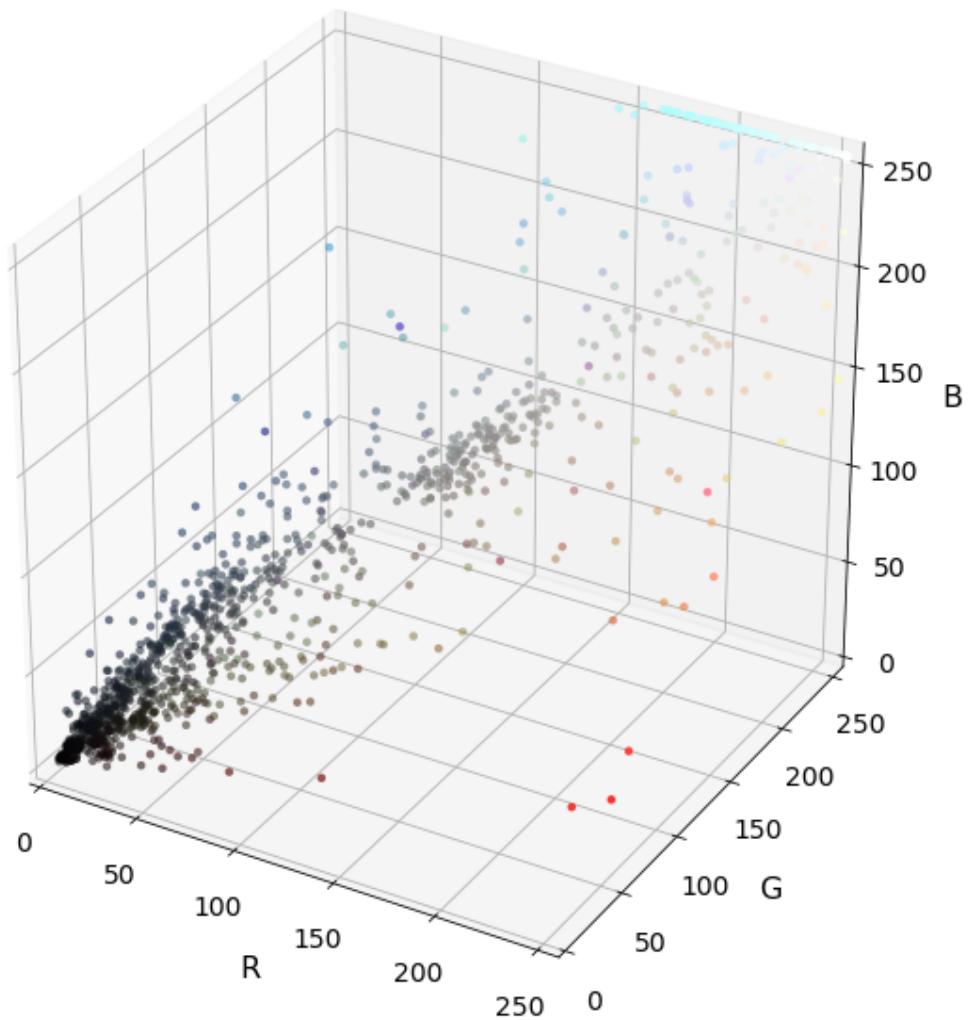


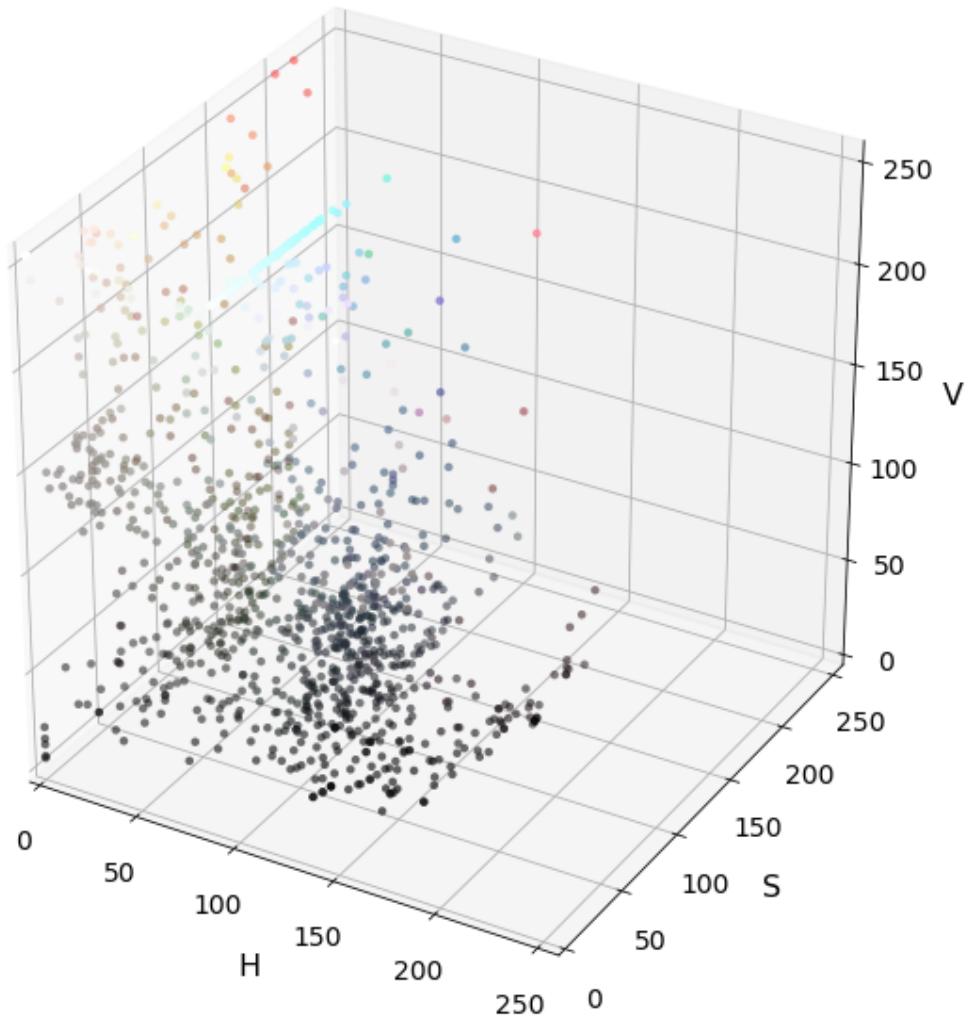
```
In [23]: # Select a small fraction of pixels to plot by subsampling it
scale = max(img.shape[0], img.shape[1], 64) / 64 # at most 64 rows and columns
img_small = cv2.resize(img, (np.int(img.shape[1] / scale),
                             np.int(img.shape[0] / scale)),
                        interpolation=cv2.INTER_NEAREST)
```

```
In [24]: # Convert subsampled image to desired color space(s)
img_small_RGB = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB) # OpenCV uses BGR, matplotlib uses RGB
img_small_HSV = cv2.cvtColor(img_small, cv2.COLOR_BGR2HSV)
img_small_rgb = img_small_RGB / 255. # scaled to [0, 1], only for plotting
```

```
In [25]: # Plot and show
plot3d(img_small_RGB, img_small_rgb)
plt.show()

plot3d(img_small_HSV, img_small_rgb, axis_labels=list("HSV"))
plt.show()
```

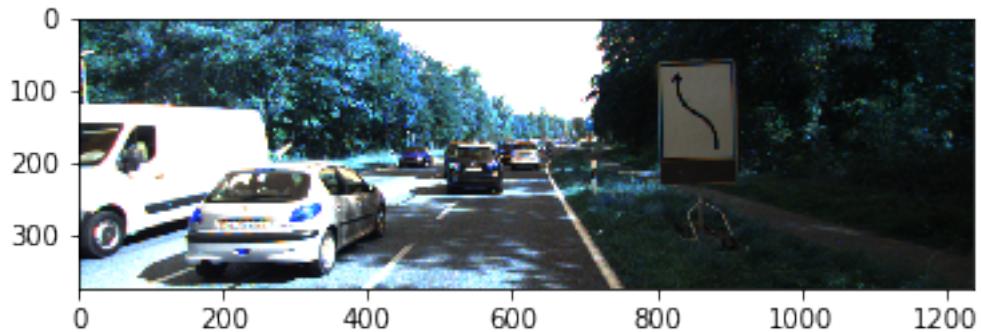




In [26]: # Read a color image

```
img = cv2.imread("/home/gvenkat/Downloads/001240.png")
plt.imshow(img)
```

Out[26]: <matplotlib.image.AxesImage at 0x7f9126c37e10>

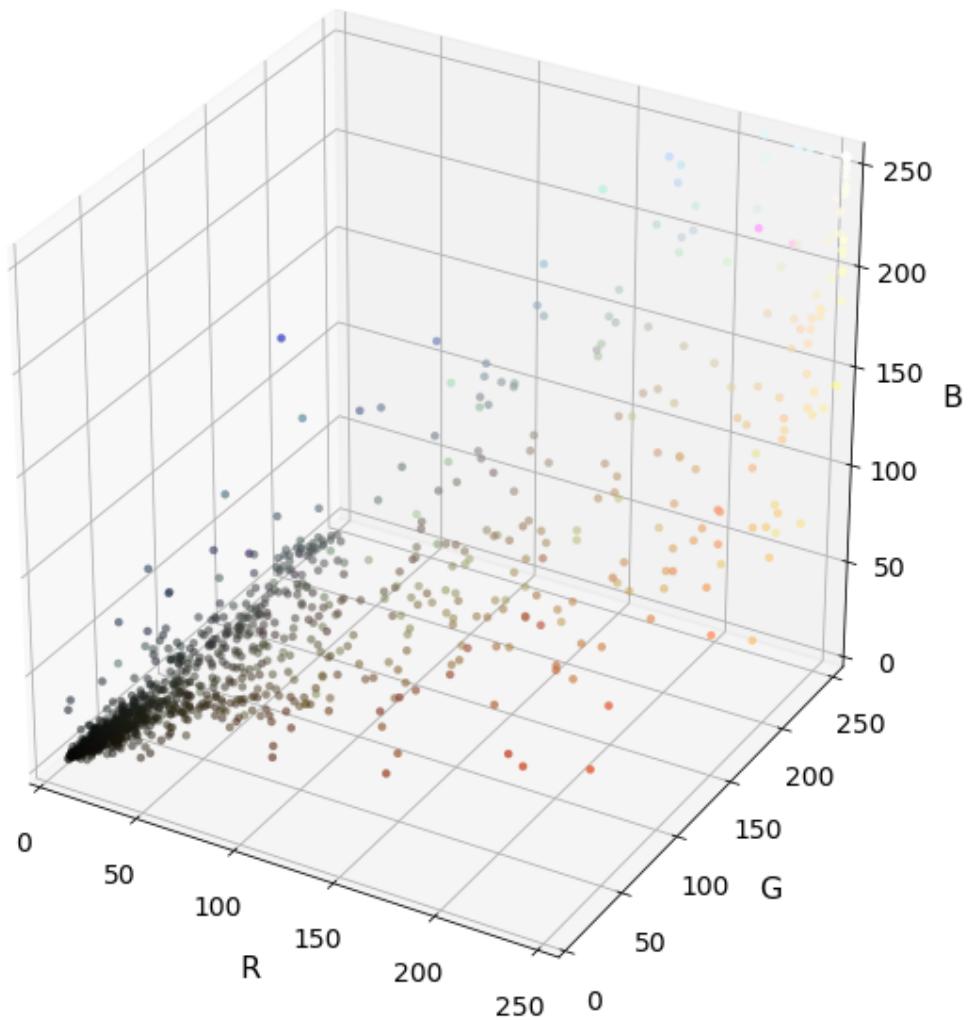


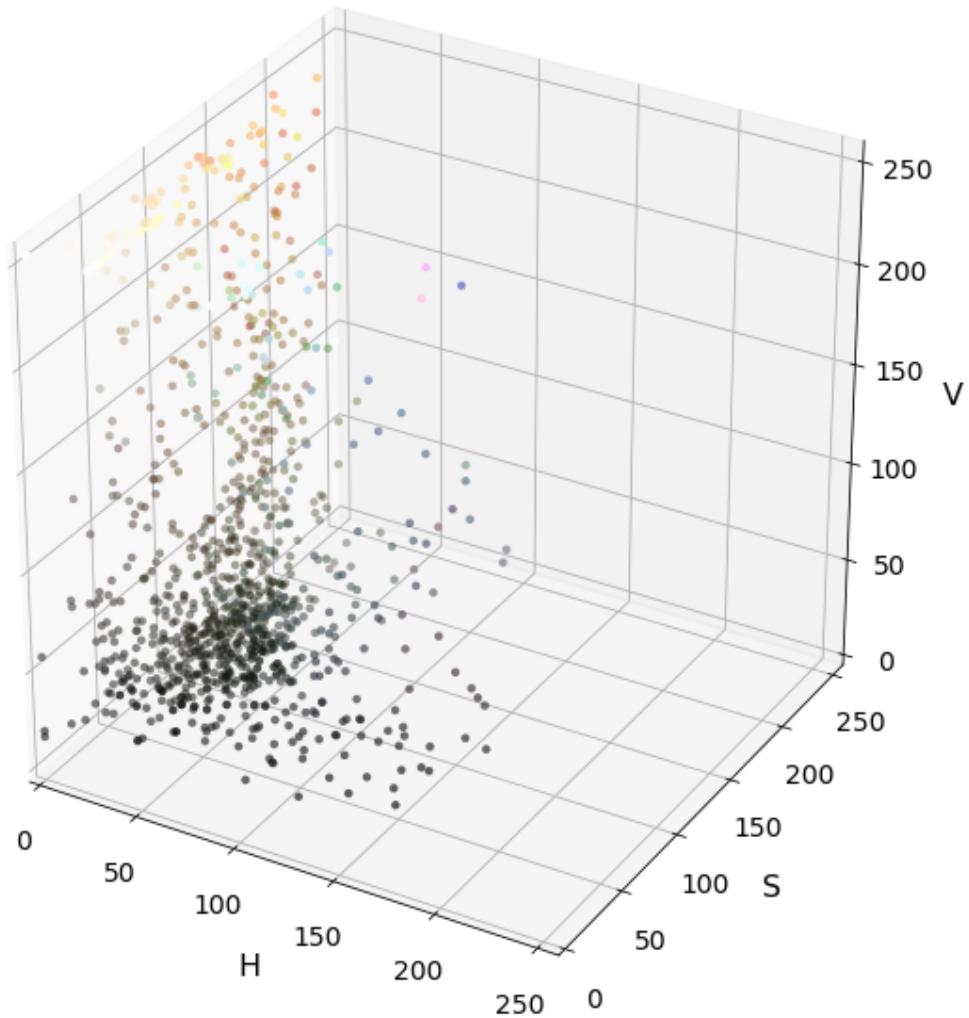
```
In [27]: # Select a small fraction of pixels to plot by subsampling it
scale = max(img.shape[0], img.shape[1], 64) / 64 # at most 64 rows and columns
img_small = cv2.resize(img, (np.int(img.shape[1] / scale),
                             np.int(img.shape[0] / scale)),
                        interpolation=cv2.INTER_NEAREST)
```

```
In [28]: # Convert subsampled image to desired color space(s)
img_small_RGB = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB) # OpenCV uses BGR, matplotlib uses RGB
img_small_HSV = cv2.cvtColor(img_small, cv2.COLOR_BGR2HSV)
img_small_rgb = img_small_RGB / 255. # scaled to [0, 1], only for plotting
```

```
In [29]: # Plot and show
plot3d(img_small_RGB, img_small_rgb)
plt.show()

plot3d(img_small_HSV, img_small_rgb, axis_labels=list("HSV"))
plt.show()
```

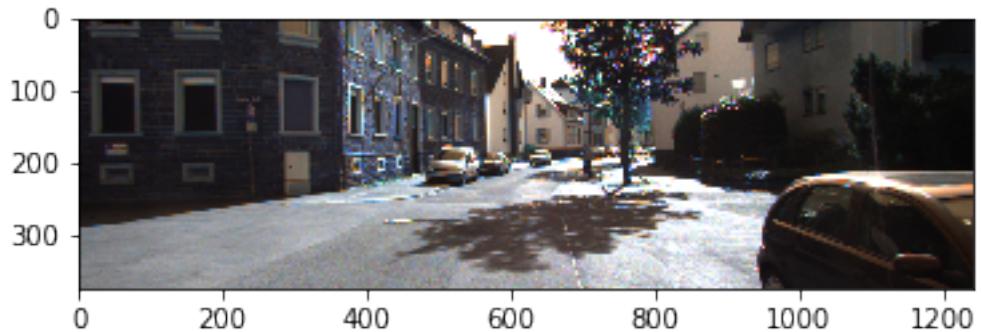




In [30]: # Read a color image

```
img = cv2.imread("/home/gvenkat/Downloads/000528.png")
plt.imshow(img)
```

Out[30]: <matplotlib.image.AxesImage at 0x7f911581ba20>

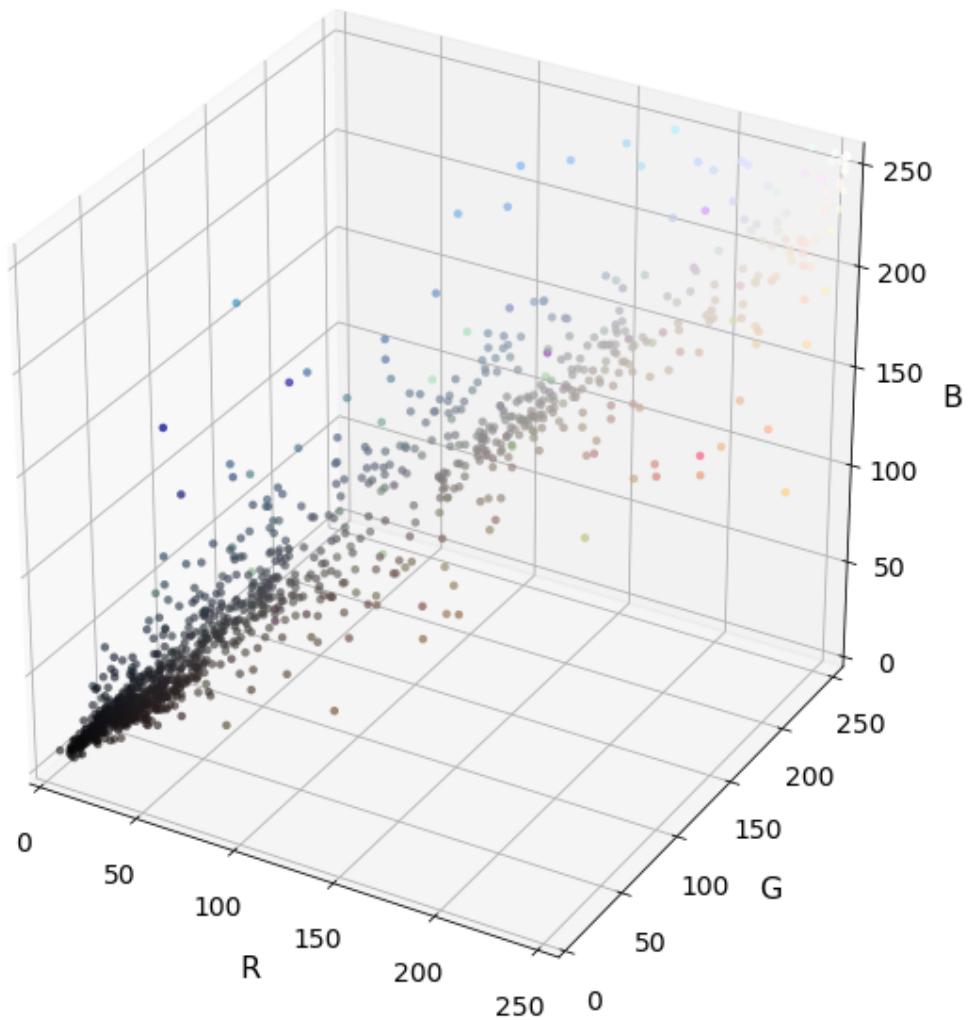


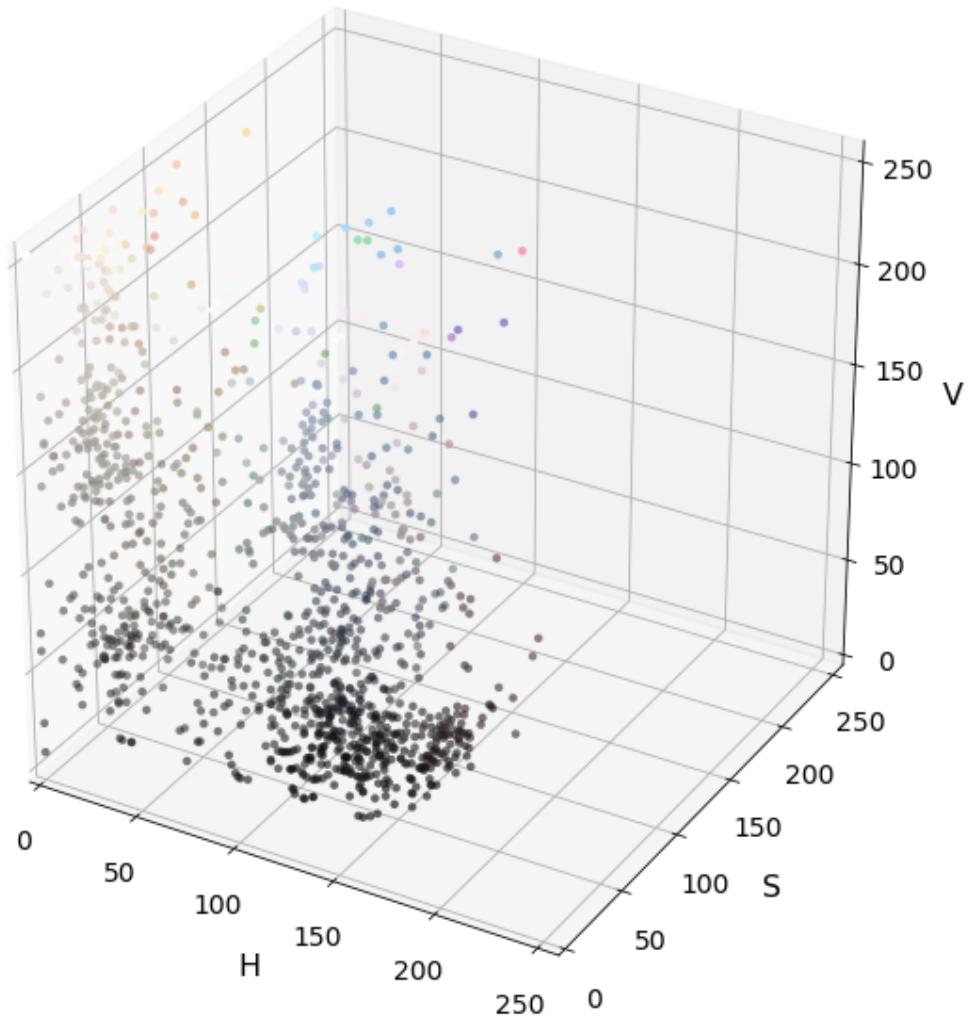
```
In [31]: # Select a small fraction of pixels to plot by subsampling it
scale = max(img.shape[0], img.shape[1], 64) / 64 # at most 64 rows and columns
img_small = cv2.resize(img, (np.int(img.shape[1] / scale),
                             np.int(img.shape[0] / scale)),
                        interpolation=cv2.INTER_NEAREST)
```

```
In [32]: # Convert subsampled image to desired color space(s)
img_small_RGB = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB) # OpenCV uses BGR, matplotlib uses RGB
img_small_HSV = cv2.cvtColor(img_small, cv2.COLOR_BGR2HSV)
img_small_rgb = img_small_RGB / 255. # scaled to [0, 1], only for plotting
```

```
In [33]: # Plot and show
plot3d(img_small_RGB, img_small_rgb)
plt.show()

plot3d(img_small_HSV, img_small_rgb, axis_labels=list("HSV"))
plt.show()
```

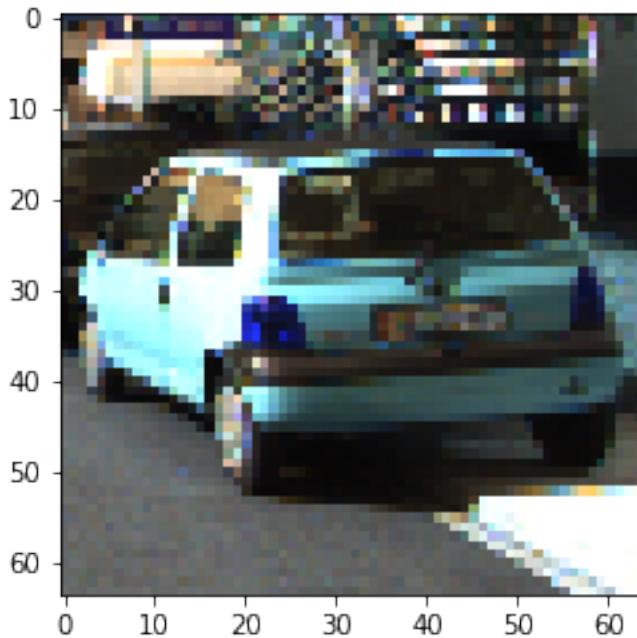




In [34]: # Read a color image

```
img = cv2.imread("/home/gvenkat/Downloads/25.png")
plt.imshow(img)
```

Out[34]: <matplotlib.image.AxesImage at 0x7f911556e908>

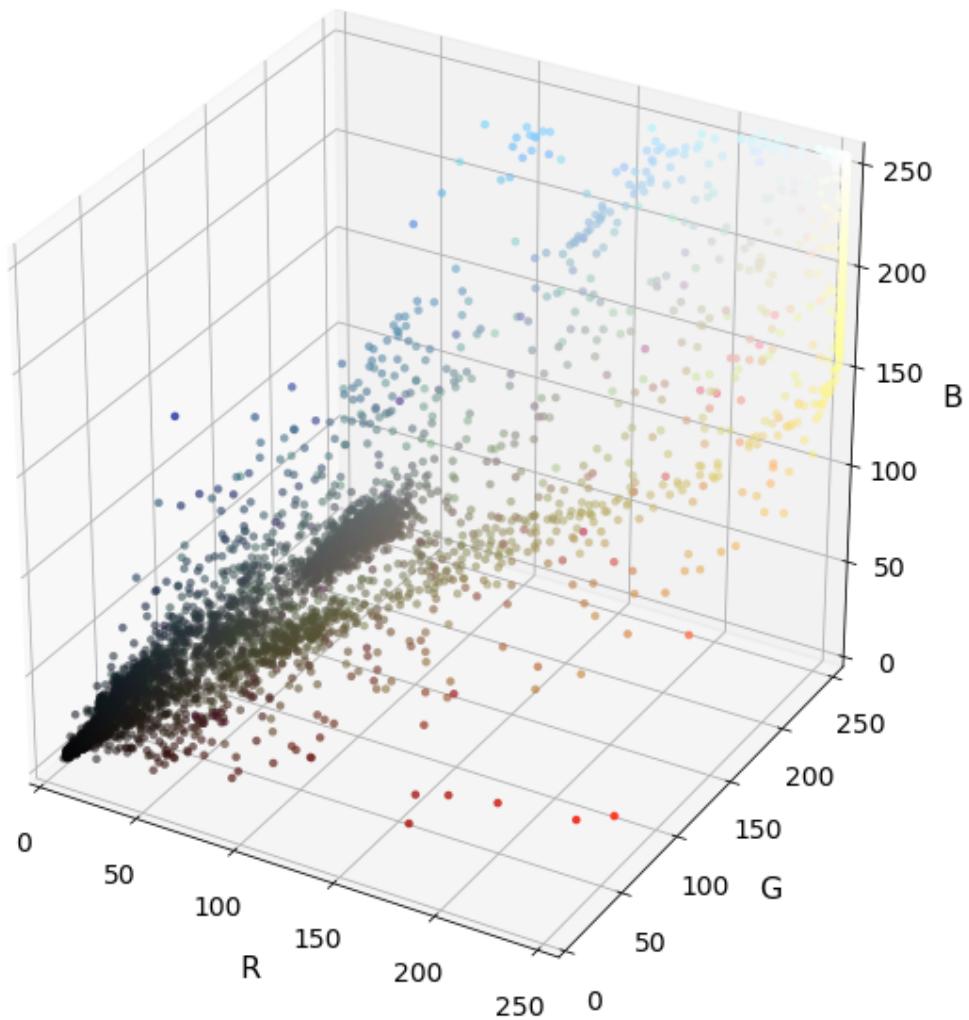


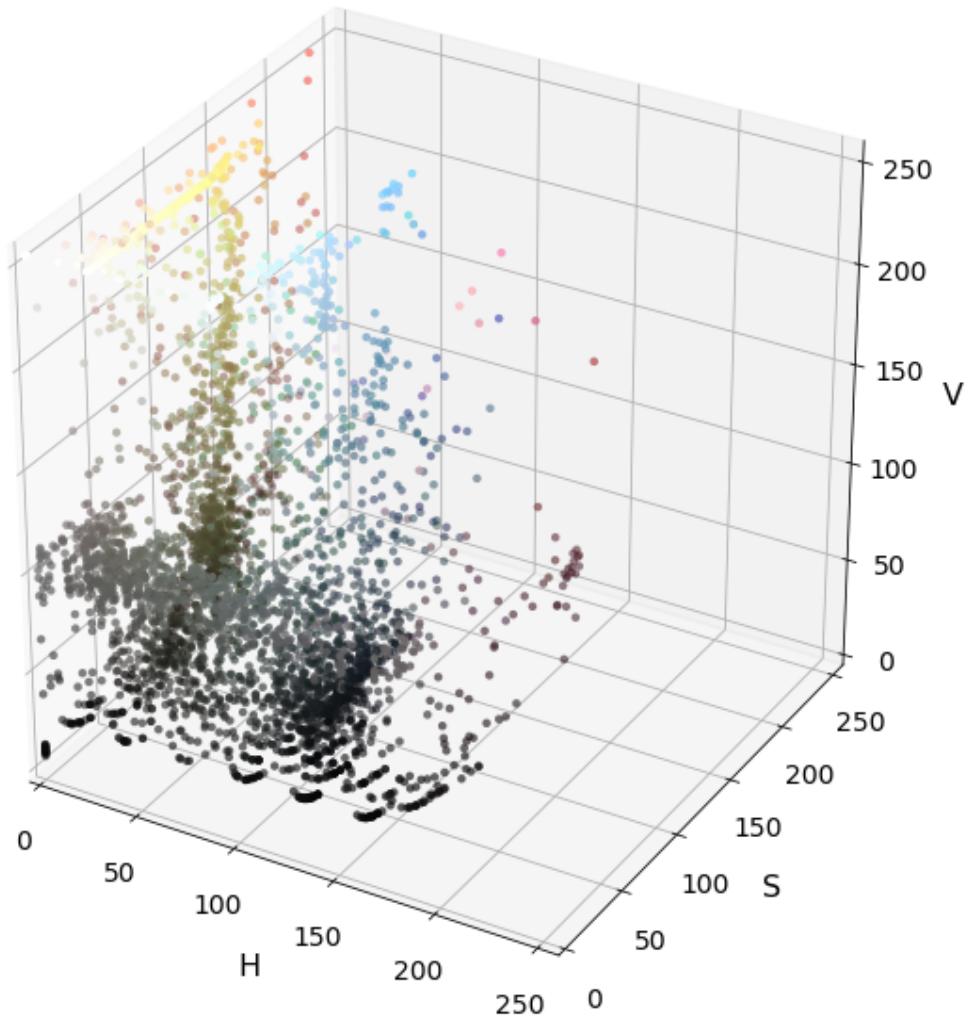
```
In [35]: # Select a small fraction of pixels to plot by subsampling it
scale = max(img.shape[0], img.shape[1], 64) / 64 # at most 64 rows and columns
img_small = cv2.resize(img, (np.int(img.shape[1] / scale),
                             np.int(img.shape[0] / scale)),
                       interpolation=cv2.INTER_NEAREST)

In [36]: # Convert subsampled image to desired color space(s)
img_small_RGB = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB) # OpenCV uses BGR, matplotlib uses RGB
img_small_HSV = cv2.cvtColor(img_small, cv2.COLOR_BGR2HSV)
img_small_rgb = img_small_RGB / 255. # scaled to [0, 1], only for plotting

In [37]: # Plot and show
plot3d(img_small_RGB, img_small_rgb)
plt.show()

plot3d(img_small_HSV, img_small_rgb, axis_labels=list("HSV"))
plt.show()
```

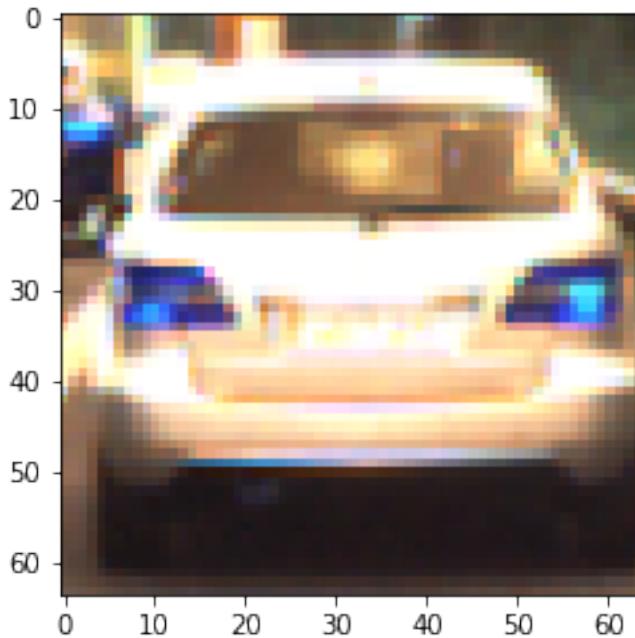




In [38]: # Read a color image

```
img = cv2.imread("/home/gvenkat/Downloads/31.png")
plt.imshow(img)
```

Out[38]: <matplotlib.image.AxesImage at 0x7f91155b9198>

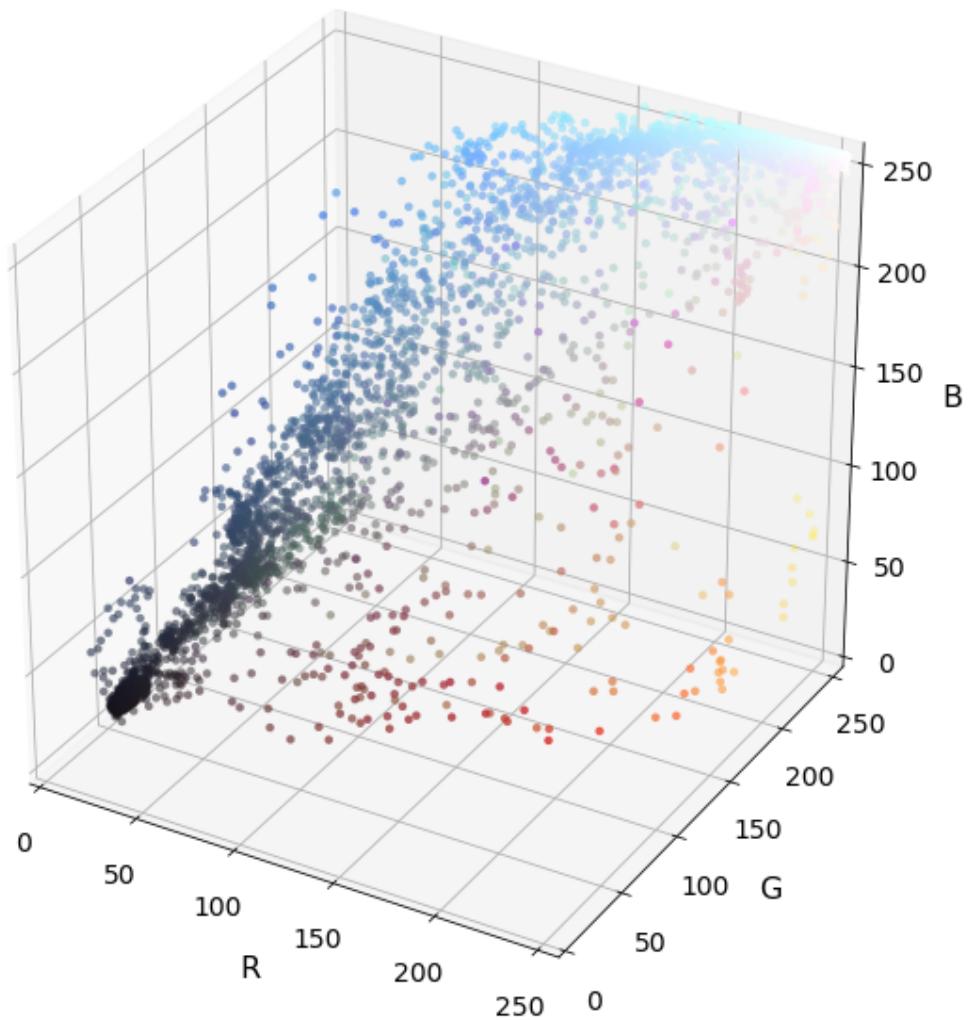


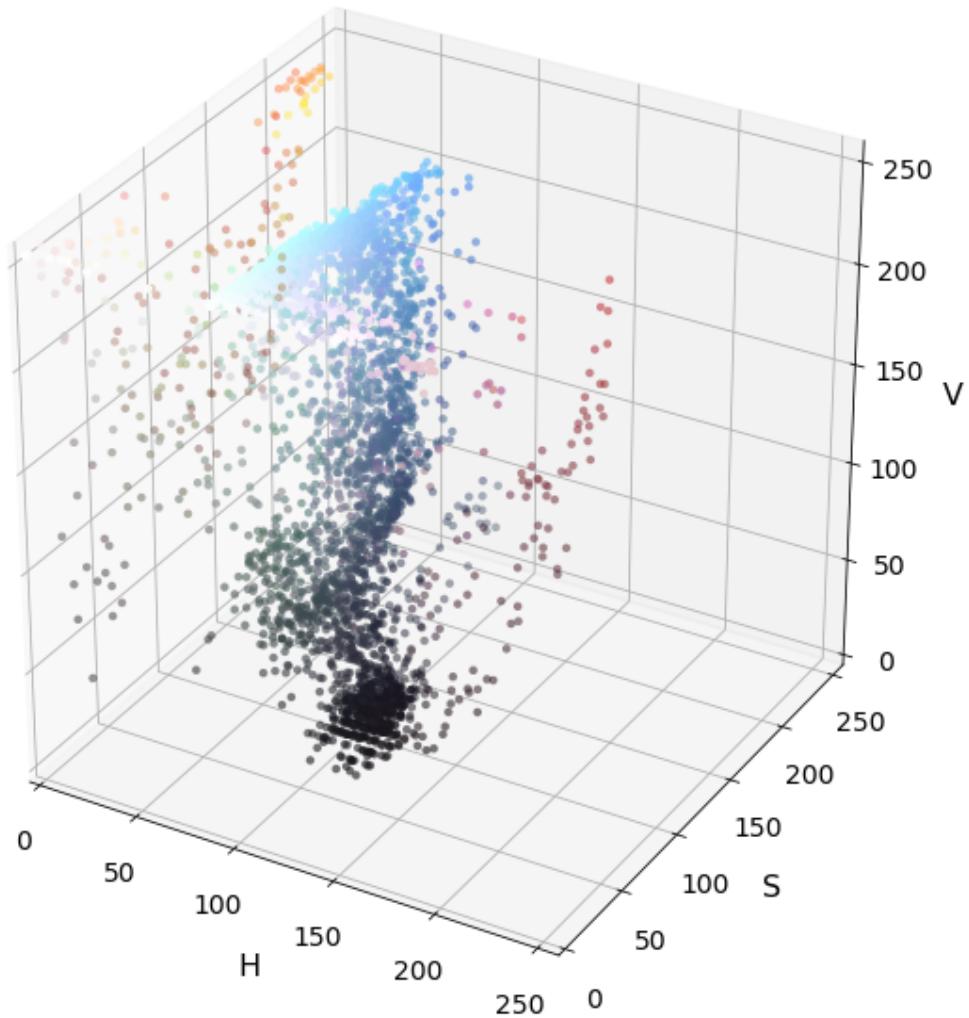
```
In [39]: # Select a small fraction of pixels to plot by subsampling it
scale = max(img.shape[0], img.shape[1], 64) / 64 # at most 64 rows and columns
img_small = cv2.resize(img, (np.int(img.shape[1] / scale),
                             np.int(img.shape[0] / scale)),
                       interpolation=cv2.INTER_NEAREST)
```

```
In [40]: # Convert subsampled image to desired color space(s)
img_small_RGB = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB) # OpenCV uses BGR, matplotlib uses RGB
img_small_HSV = cv2.cvtColor(img_small, cv2.COLOR_BGR2HSV)
img_small_rgb = img_small_RGB / 255. # scaled to [0, 1], only for plotting
```

```
In [41]: # Plot and show
plot3d(img_small_RGB, img_small_rgb)
plt.show()

plot3d(img_small_HSV, img_small_rgb, axis_labels=list("HSV"))
plt.show()
```

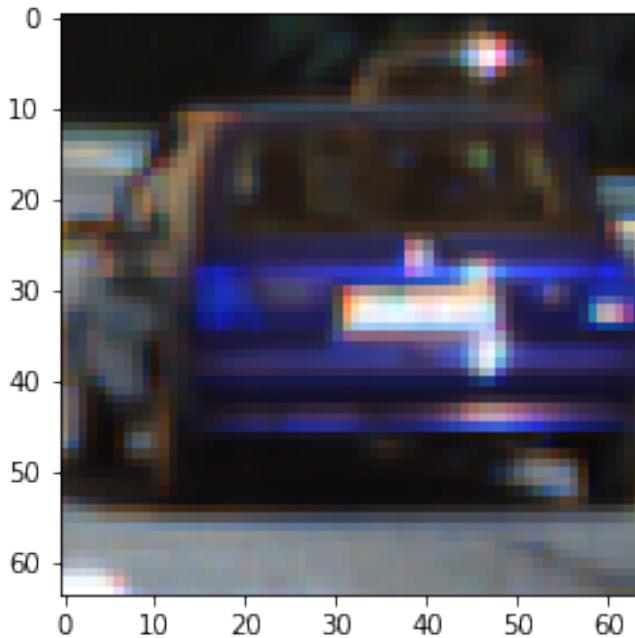




In [42]: # Read a color image

```
img = cv2.imread("/home/gvenkat/Downloads/53.png")
plt.imshow(img)
```

Out[42]: <matplotlib.image.AxesImage at 0x7f91159cb0f0>

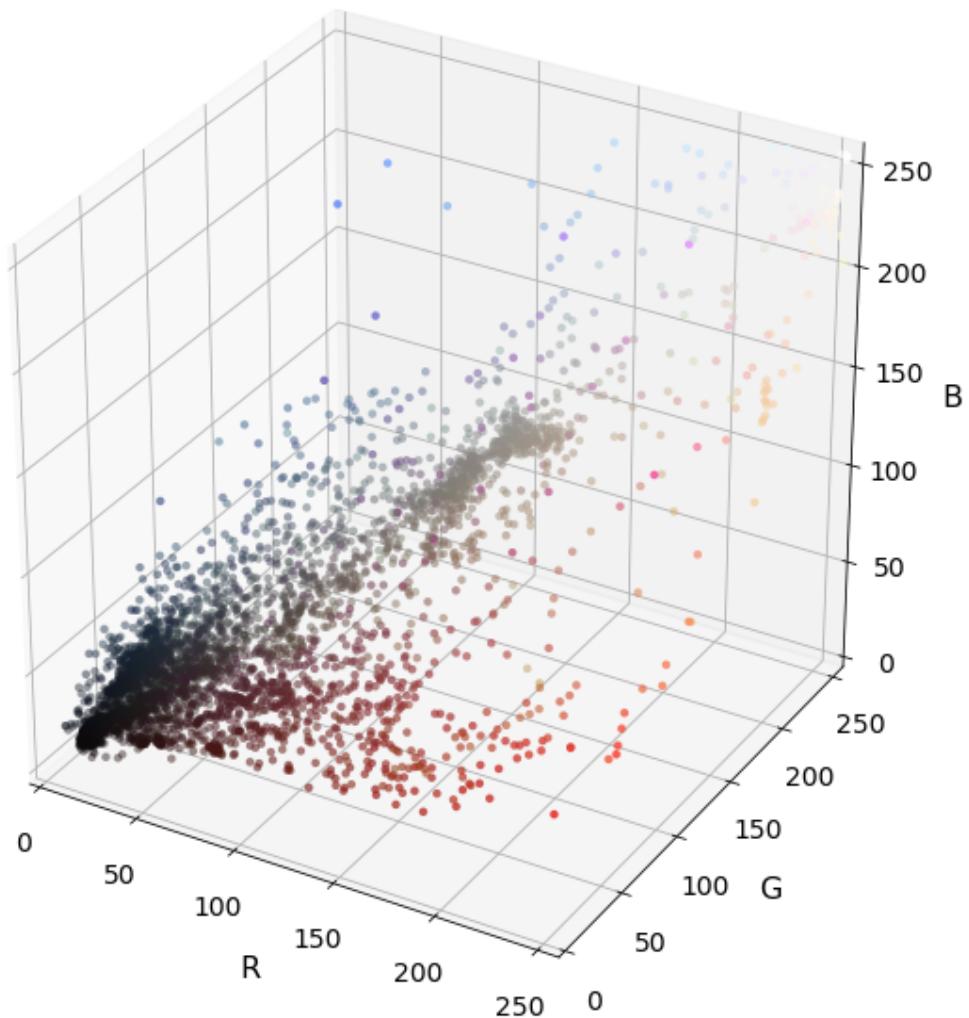


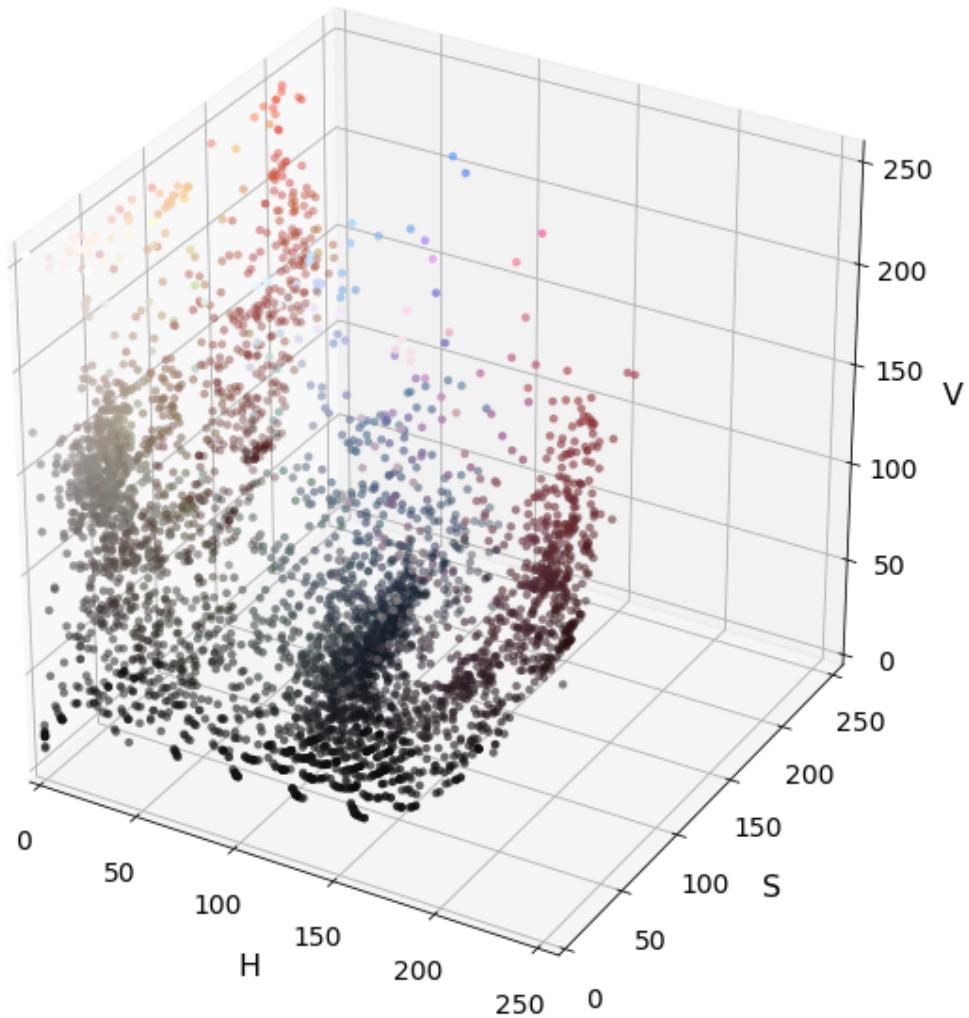
```
In [43]: # Select a small fraction of pixels to plot by subsampling it
scale = max(img.shape[0], img.shape[1], 64) / 64 # at most 64 rows and columns
img_small = cv2.resize(img, (np.int(img.shape[1] / scale),
                             np.int(img.shape[0] / scale)),
                       interpolation=cv2.INTER_NEAREST)
```

```
In [44]: # Convert subsampled image to desired color space(s)
img_small_RGB = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB) # OpenCV uses BGR, matplotlib uses RGB
img_small_HSV = cv2.cvtColor(img_small, cv2.COLOR_BGR2HSV)
img_small_rgb = img_small_RGB / 255. # scaled to [0, 1], only for plotting
```

```
In [45]: # Plot and show
plot3d(img_small_RGB, img_small_rgb)
plt.show()

plot3d(img_small_HSV, img_small_rgb, axis_labels=list("HSV"))
plt.show()
```

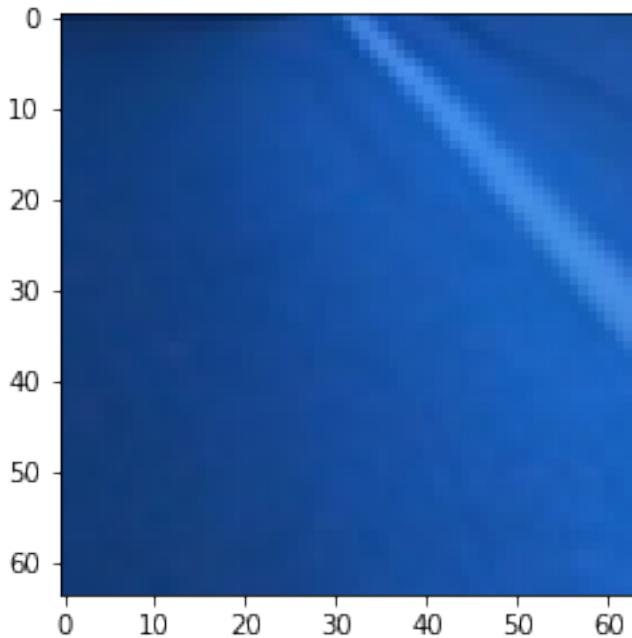




In [46]: # Read a color image

```
img = cv2.imread("/home/gvenkat/Downloads/8.png")
plt.imshow(img)
```

Out[46]: <matplotlib.image.AxesImage at 0x7f9114b42550>

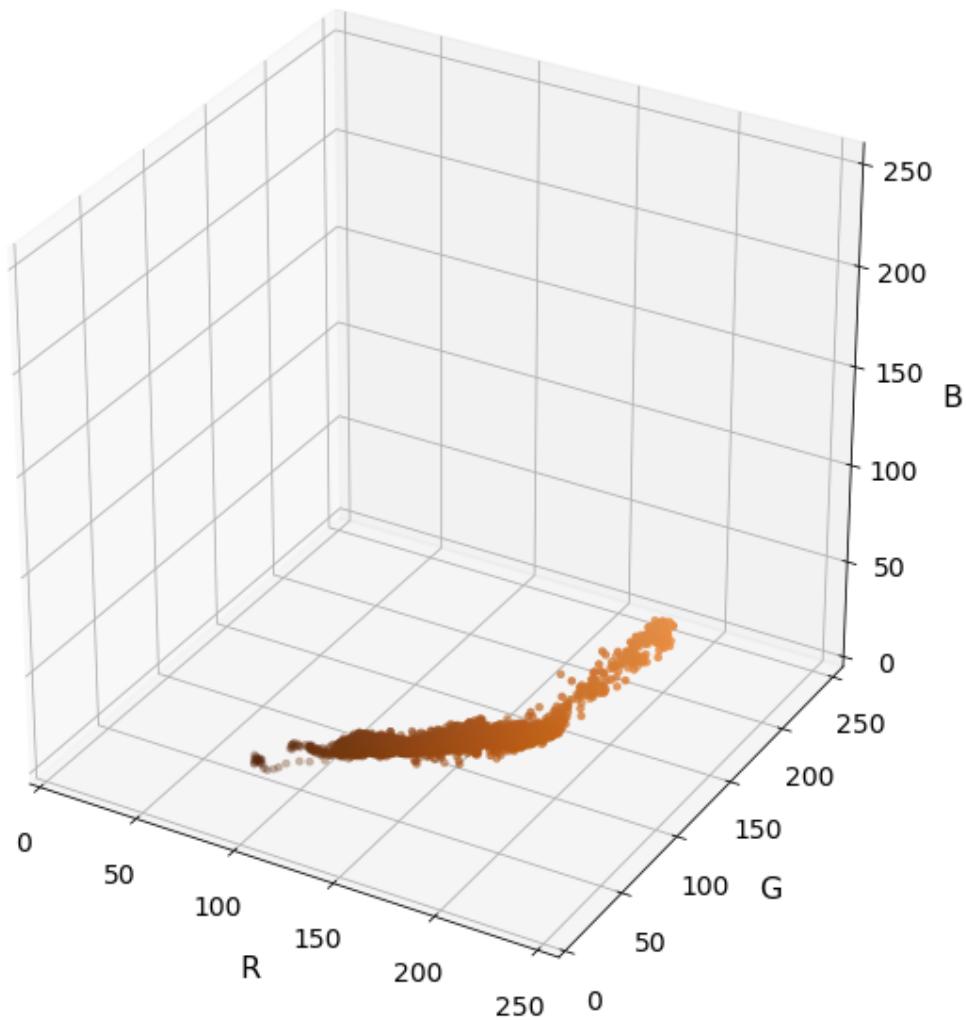


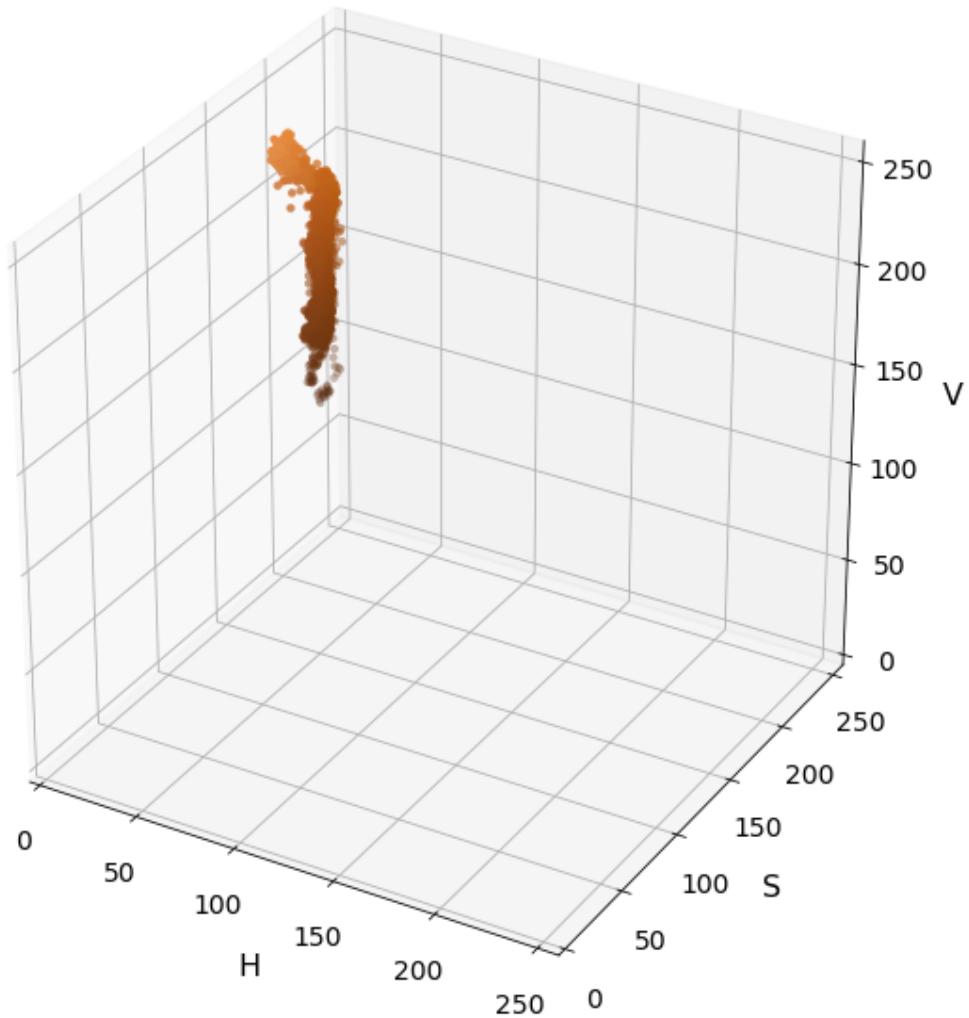
```
In [47]: # Select a small fraction of pixels to plot by subsampling it
scale = max(img.shape[0], img.shape[1], 64) / 64 # at most 64 rows and columns
img_small = cv2.resize(img, (np.int(img.shape[1] / scale),
                             np.int(img.shape[0] / scale)),
                       interpolation=cv2.INTER_NEAREST)
```

```
In [48]: # Convert subsampled image to desired color space(s)
img_small_RGB = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB) # OpenCV uses BGR, matplotlib uses RGB
img_small_HSV = cv2.cvtColor(img_small, cv2.COLOR_BGR2HSV)
img_small_rgb = img_small_RGB / 255. # scaled to [0, 1], only for plotting
```

```
In [49]: # Plot and show
plot3d(img_small_RGB, img_small_rgb)
plt.show()

plot3d(img_small_HSV, img_small_rgb, axis_labels=list("HSV"))
plt.show()
```

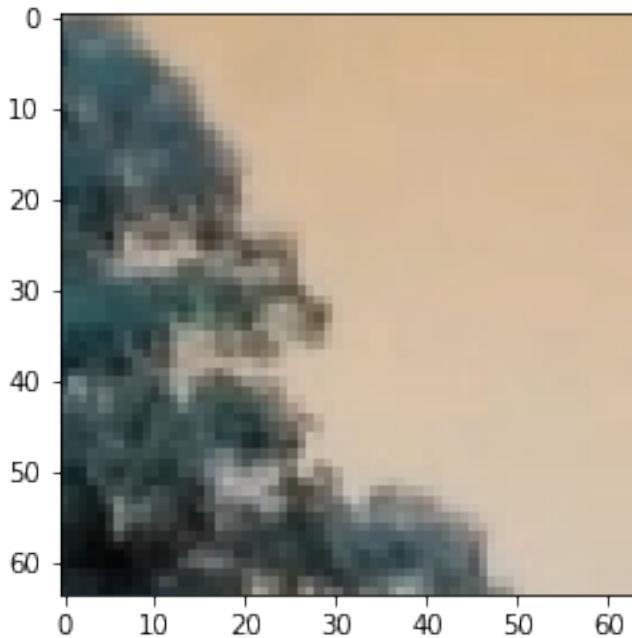




In [50]: # Read a color image

```
img = cv2.imread("/home/gvenkat/Downloads/2.png")
plt.imshow(img)
```

Out[50]: <matplotlib.image.AxesImage at 0x7f91148161d0>

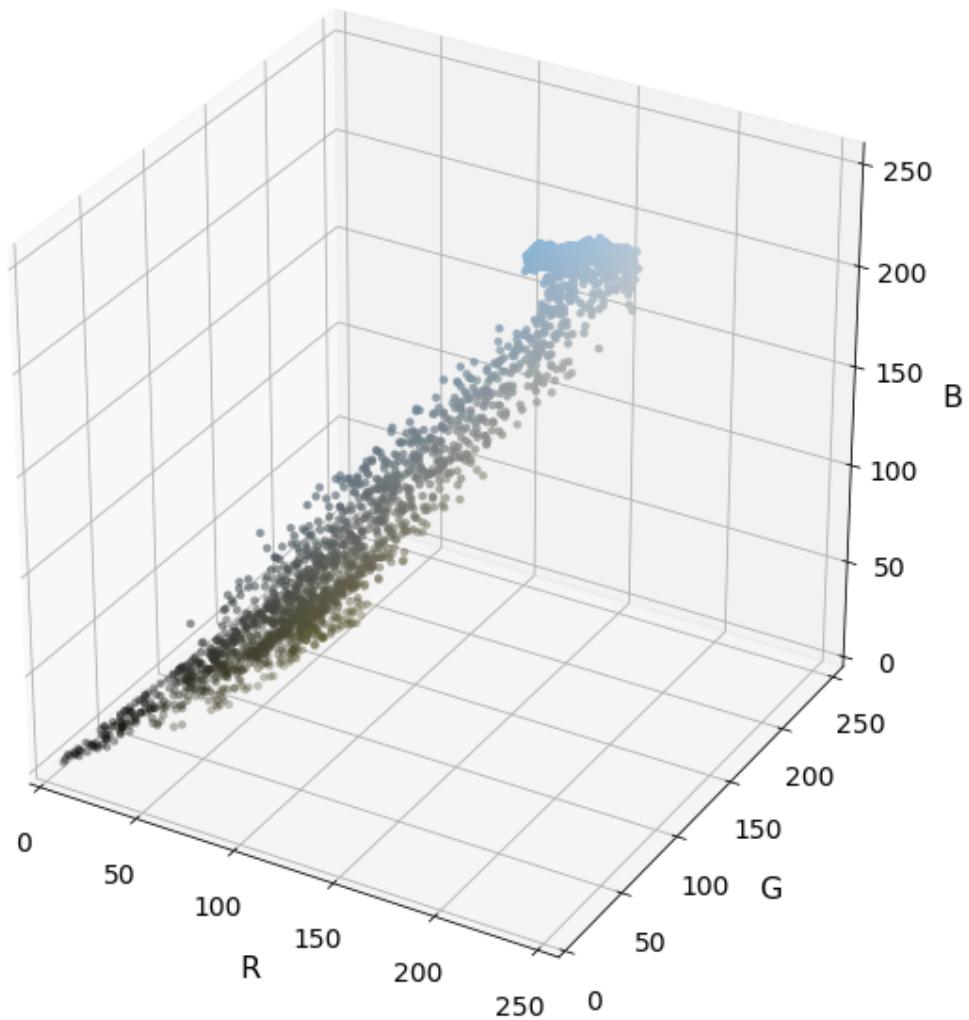


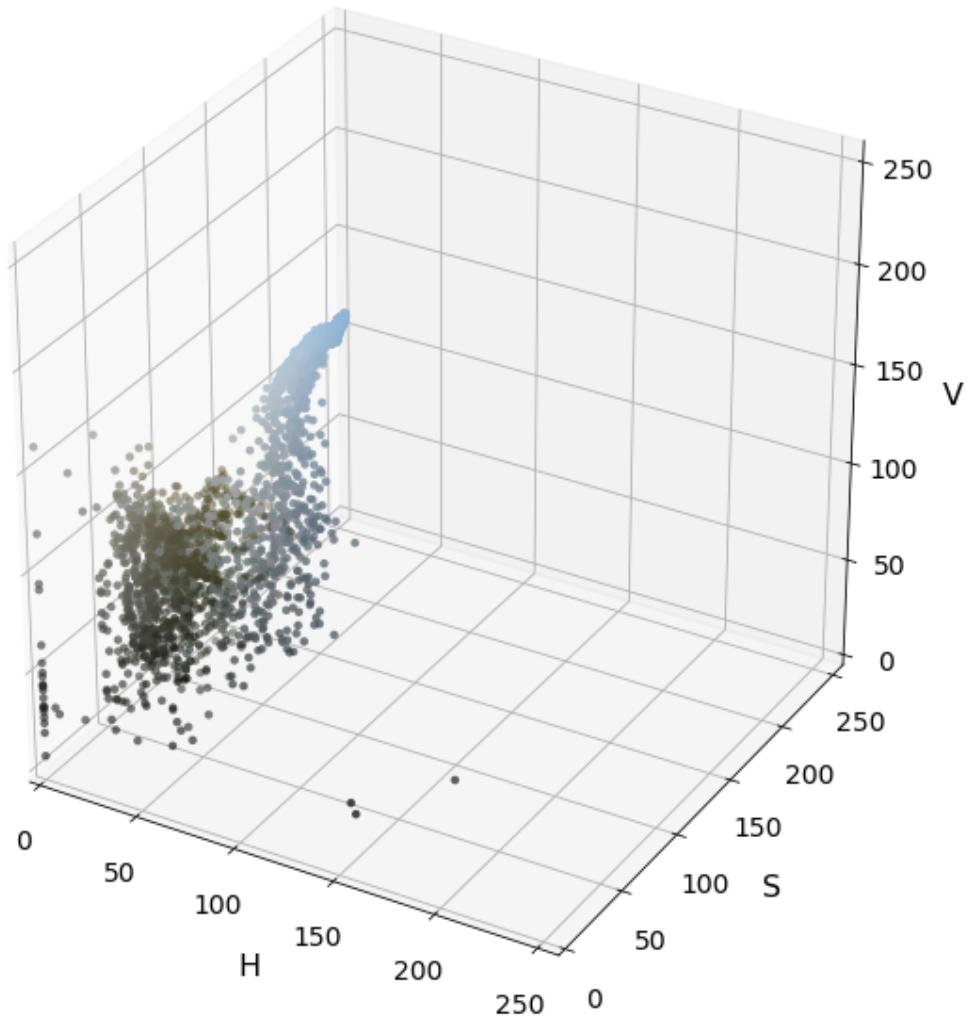
```
In [51]: # Select a small fraction of pixels to plot by subsampling it
scale = max(img.shape[0], img.shape[1], 64) / 64 # at most 64 rows and columns
img_small = cv2.resize(img, (np.int(img.shape[1] / scale),
                             np.int(img.shape[0] / scale)),
                       interpolation=cv2.INTER_NEAREST)
```

```
In [52]: # Convert subsampled image to desired color space(s)
img_small_RGB = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB) # OpenCV uses BGR, matplotlib uses RGB
img_small_HSV = cv2.cvtColor(img_small, cv2.COLOR_BGR2HSV)
img_small_rgb = img_small_RGB / 255. # scaled to [0, 1], only for plotting
```

```
In [53]: # Plot and show
plot3d(img_small_RGB, img_small_rgb)
plt.show()

plot3d(img_small_HSV, img_small_rgb, axis_labels=list("HSV"))
plt.show()
```

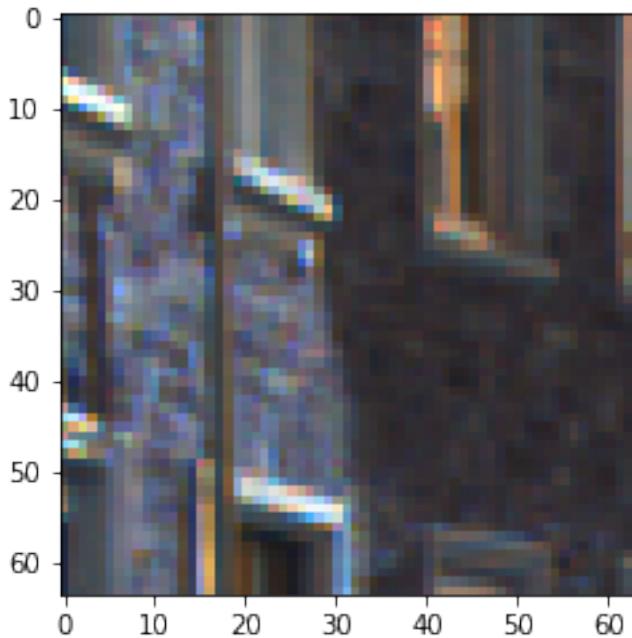




In [54]: # Read a color image

```
img = cv2.imread("/home/gvenkat/Downloads/3.png")
plt.imshow(img)
```

Out[54]: <matplotlib.image.AxesImage at 0x7f911449ada0>

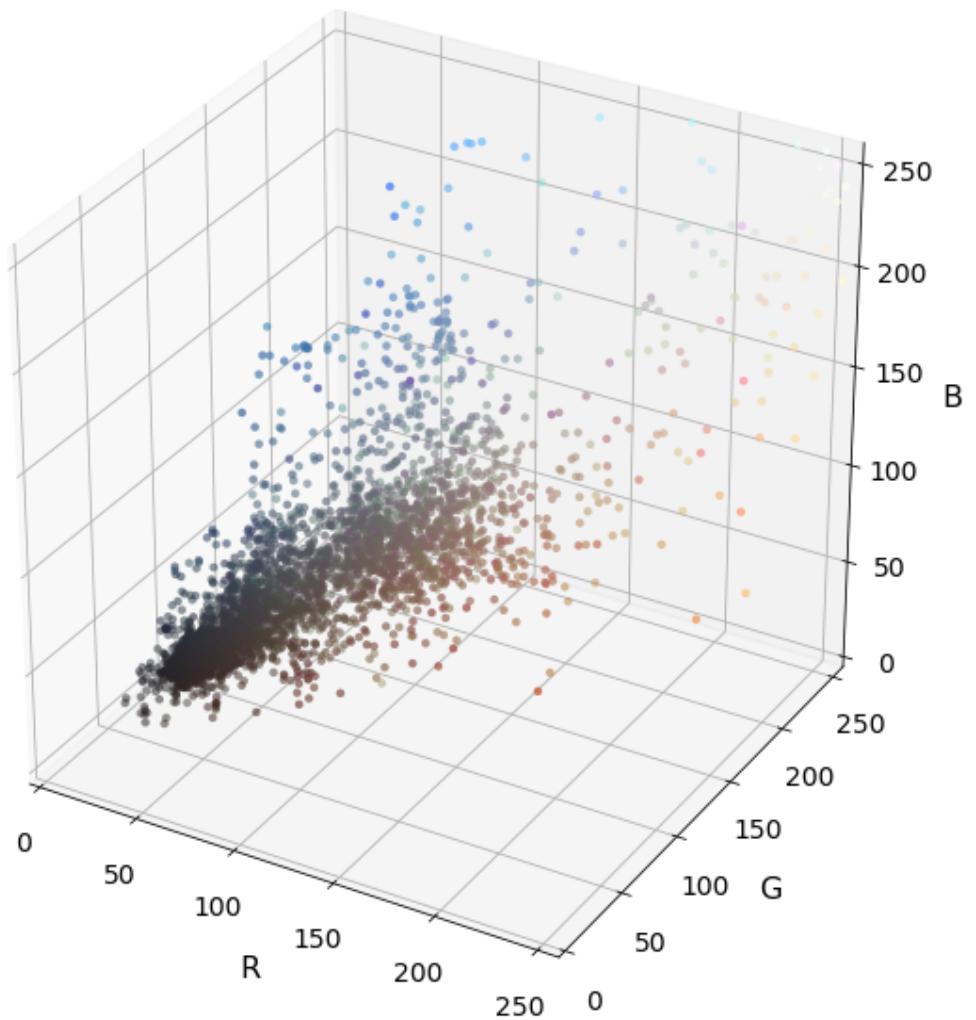


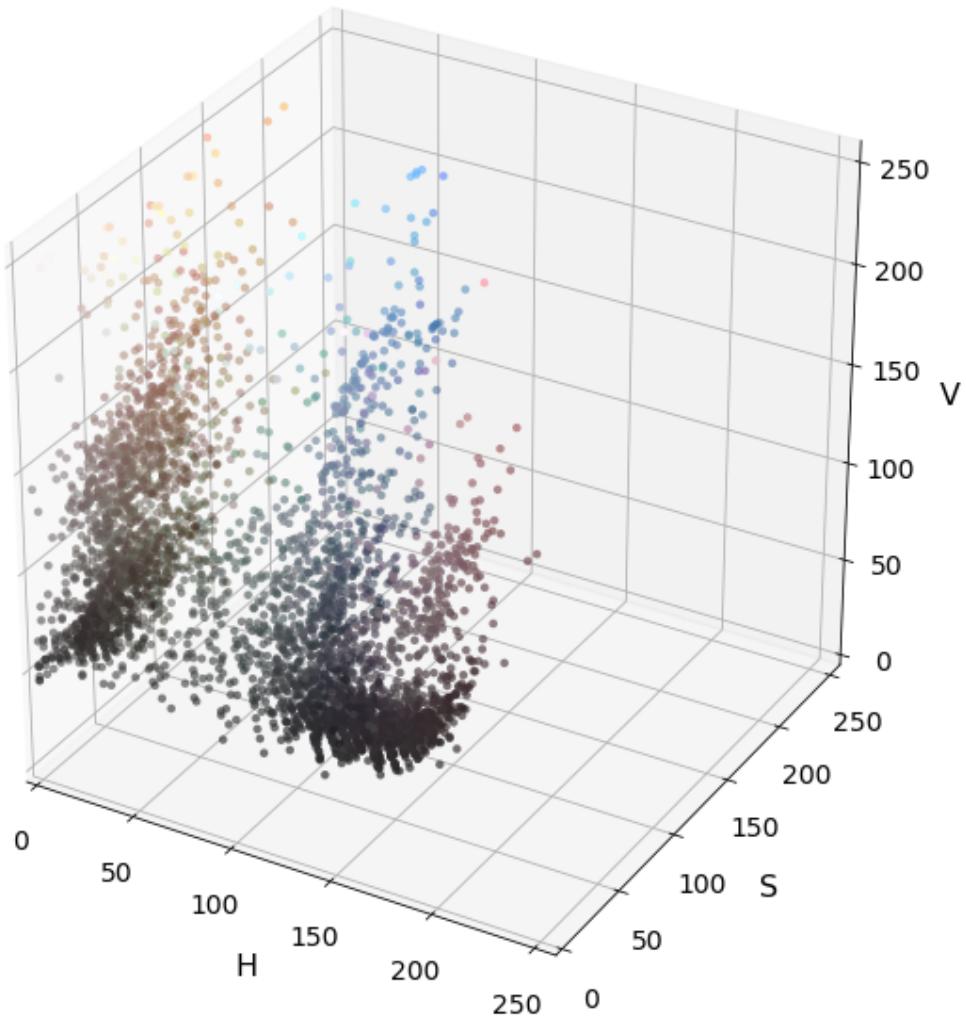
```
In [55]: # Select a small fraction of pixels to plot by subsampling it
scale = max(img.shape[0], img.shape[1], 64) / 64 # at most 64 rows and columns
img_small = cv2.resize(img, (np.int(img.shape[1] / scale),
                             np.int(img.shape[0] / scale)),
                        interpolation=cv2.INTER_NEAREST)
```

```
In [56]: # Convert subsampled image to desired color space(s)
img_small_RGB = cv2.cvtColor(img_small, cv2.COLOR_BGR2RGB) # OpenCV uses BGR, matplotlib uses RGB
img_small_HSV = cv2.cvtColor(img_small, cv2.COLOR_BGR2HSV)
img_small_rgb = img_small_RGB / 255. # scaled to [0, 1], only for plotting
```

```
In [57]: # Plot and show
plot3d(img_small_RGB, img_small_rgb)
plt.show()

plot3d(img_small_HSV, img_small_rgb, axis_labels=list("HSV"))
plt.show()
```





```
In [58]: image = mpimg.imread('/home/gvenkat/Downloads/test_img.jpg')
small_img = cv2.resize(image, (32, 32))
print(small_img.shape)
```

```
(32, 32, 3)
```

```
In [59]: feature_vec = small_img.ravel()
print(feature_vec.shape)
```

```
(3072,)
```

```
In [60]: image = mpimg.imread('/home/gvenkat/Downloads/cutouts/cutout1.jpg')
plt.imshow(image)
```

```
Out[60]: <matplotlib.image.AxesImage at 0x7f910ffe4828>
```



```
In [61]: # Define a function to compute color histogram features
# Pass the color_space flag as 3-letter all caps string
# like 'HSV' or 'LUV' etc.
# KEEP IN MIND IF YOU DECIDE TO USE THIS FUNCTION LATER
# IN YOUR PROJECT THAT IF YOU READ THE IMAGE WITH
# cv2.imread() INSTEAD YOU START WITH BGR COLOR!
def bin_spatial(img, color_space='RGB', size=(32, 32)):
    # Convert image to new color space (if specified)
    if color_space != 'RGB':
        if color_space == 'HSV':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)
        elif color_space == 'LUV':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2LUV)
        elif color_space == 'HLS':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2HLS)
        elif color_space == 'YUV':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2YUV)
        elif color_space == 'YCrCb':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2YCrCb)
    else: feature_image = np.copy(img)

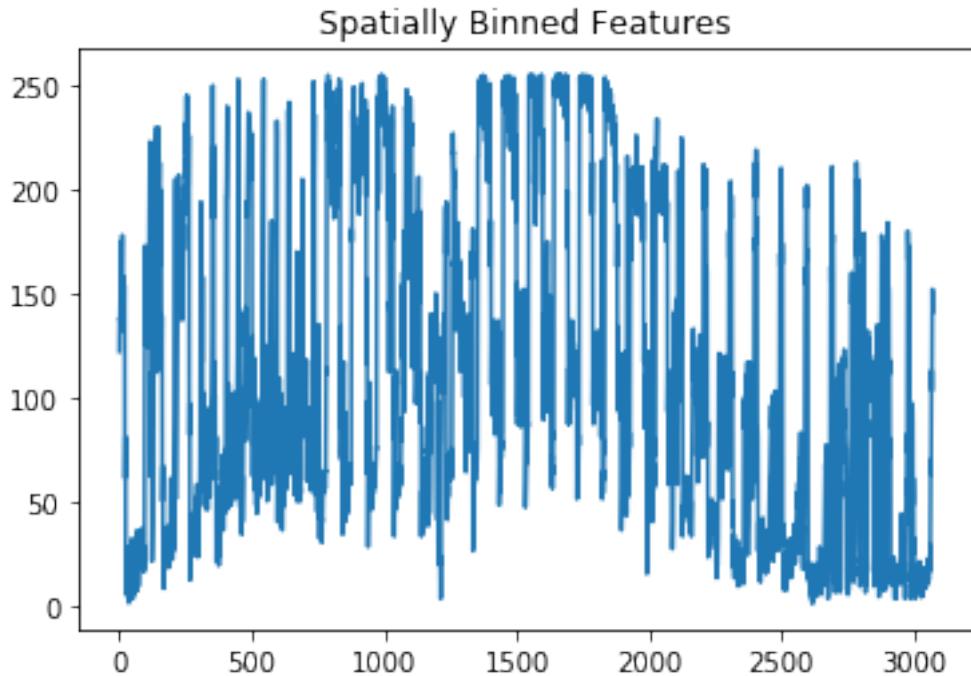
    # Use cv2.resize().ravel() to create the feature vector

    features = cv2.resize(feature_image, size).ravel() # Remove this line!
    # Return the feature vector
    return features
```

```
In [62]: feature_vec = bin_spatial(image, color_space='RGB', size=(32, 32))
```

```
In [63]: # Plot features
    plt.plot(feature_vec)
    plt.title('Spatially Binned Features')
```

```
Out[63]: Text(0.5,1,'Spatially Binned Features')
```



```
In [64]: car_images = glob.glob('data/v/vehicles/**/*.png')
noncar_images = glob.glob('data/nv/non-vehicles/**/*.png')
print(len(car_images), len(noncar_images))
```

```
8792 8968
```

```
In [65]: # Define a function to return some characteristics of the dataset
def data_look(car_list, notcar_list):
    data_dict = {}
    # Define a key in data_dict "n_cars" and store the number of car images
    data_dict["n_cars"] = len(car_list)
    # Define a key "n_notcars" and store the number of notcar images
    data_dict["n_notcars"] = len(notcar_list)
    # Read in a test image, either car or notcar
    # Define a key "image_shape" and store the test image shape 3-tuple
    sample_image = mpimg.imread(car_list[0])
```

```

data_dict["image_shape"] = sample_image.shape
# Define a key "data_type" and store the data type of the test image.
data_dict["data_type"] = sample_image.dtype
# Return data_dict
return data_dict

In [66]: data_info = data_look(car_images, noncar_images)

'''val = 1

print(val)
print(type(val))
print(type.dtype)'''

print('Your function returned a count of',
      data_info["n_cars"], ' cars and',
      data_info["n_notcars"], ' non-cars')
print('of size: ', data_info["image_shape"], ' and data type:',
      data_info["data_type"])
# Just for fun choose random car / not-car indices and plot example images
car_ind = np.random.randint(0, len(car_images))
notcar_ind = np.random.randint(0, len(noncar_images))

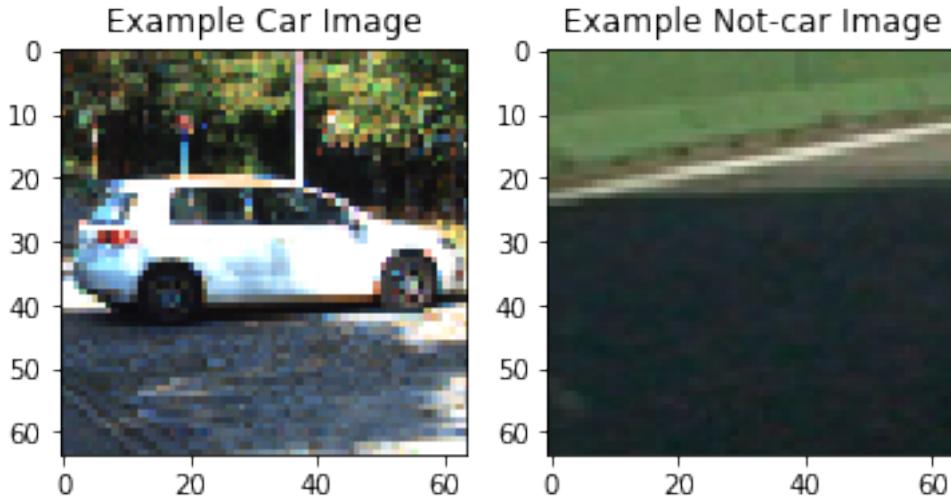
# Read in car / not-car images
car_image = mpimg.imread(car_images[car_ind])
notcar_image = mpimg.imread(noncar_images[notcar_ind])

# Plot the examples
fig = plt.figure()
plt.subplot(121)
plt.imshow(car_image)
plt.title('Example Car Image')
plt.subplot(122)
plt.imshow(notcar_image)
plt.title('Example Not-car Image')

Your function returned a count of 8792 cars and 8968 non-cars
of size: (64, 64, 3) and data type: float32

Out[66]: Text(0.5,1,'Example Not-car Image')

```



1.4 Step 0: Exploration of methods: HOG

```
In [67]: def get_hog_features(img, orient, pix_per_cell, cell_per_block, vis=True,
                           feature_vec=True):

    # TODO: Complete the function body and returns
    try:
        return_list = hog(img, orientations=orient, pixels_per_cell=(pix_per_cell, pix_
                           cells_per_block=(cell_per_block, cell_per_block),
                           block_norm= 'L2-Hys', transform_sqrt=False,
                           visualise= vis, feature_vector= feature_vec)

    except cv2.error as e:
        print('unable to obtain hog features')
        raise
    # name returns explicitly
    hog_features = return_list[0]
    if vis:
        hog_image = return_list[1]
        return hog_features, hog_image
    else:
        return hog_features
```

```
In [68]: # Generate a random index to look at a car image
ind = np.random.randint(0, len(car_images))
# Read in the image
image = mpimg.imread(car_images[ind])
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

# Call our function with vis=True to see an image output
```

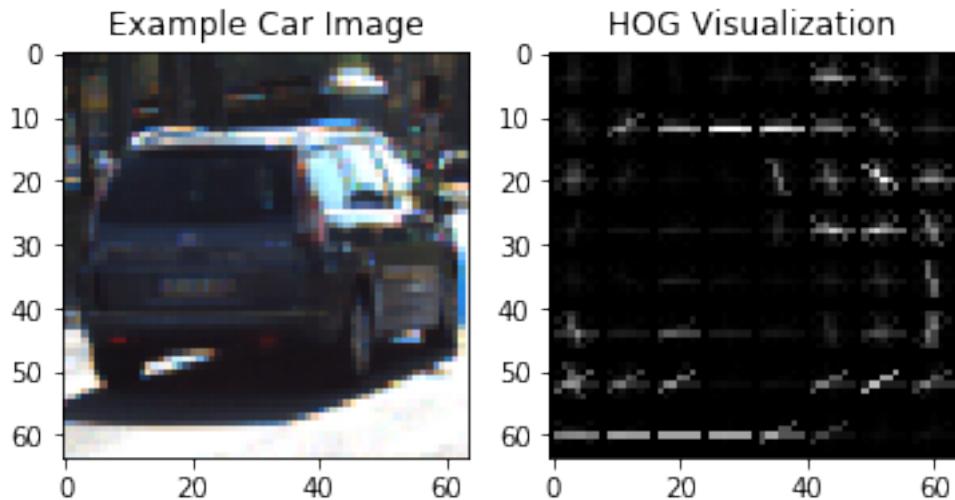
```

features, hog_image = get_hog_features(gray, orient= 9,
                                         pix_per_cell= 8, cell_per_block= 2,
                                         vis=True, feature_vec=True)

In [69]: # Plot the examples
fig = plt.figure()
plt.subplot(121)
plt.imshow(image, cmap='gray')
plt.title('Example Car Image')
plt.subplot(122)
plt.imshow(hog_image, cmap='gray')
plt.title('HOG Visualization')

Out[69]: Text(0.5,1,'HOG Visualization')

```



1.5 Step 0: Exploration of methods: Combining methods(color histograms and spacial binning) for feature extraction and normalization

```

In [70]: # Define a function to compute binned color features
def bin_spatial(img, size=(32, 32)):
    # Use cv2.resize().ravel() to create the feature vector
    features = cv2.resize(img, size).ravel()
    # Return the feature vector
    return features

# Define a function to compute color histogram features
def color_hist(img, nbins=32, bins_range=(0, 256)):
    # Compute the histogram of the color channels separately
    channel1_hist = np.histogram(img[:, :, 0], bins=nbins, range=bins_range)
    channel2_hist = np.histogram(img[:, :, 1], bins=nbins, range=bins_range)

```

```

channel3_hist = np.histogram(img[:, :, 2], bins=nbins, range=bins_range)
# Concatenate the histograms into a single feature vector
hist_features = np.concatenate((channel1_hist[0], channel2_hist[0], channel3_hist[0]))
# Return the individual histograms, bin_centers and feature vector
return hist_features

In [71]: # Define a function to extract features from a list of images
        # Have this function call bin_spatial() and color_hist()
def extract_features(imgs, cspace='RGB', spatial_size=(32, 32),
                     hist_bins=32, hist_range=(0, 256)):
    # Create a list to append feature vectors to
    features = []
    # Iterate through the list of images
    for img in imgs:
        # Read in each one by one
        image = mpimg.imread(img)
        # apply color conversion if other than 'RGB'
        if cspace != 'RGB':
            if cspace == 'HSV':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
            elif cspace == 'LUV':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2LUV)
            elif cspace == 'HLS':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2HLS)
            elif cspace == 'YUV':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2YUV)
            else: feature_image = np.copy(image)
        # Apply bin_spatial() to get spatial color features
        spatial_features = bin_spatial(feature_image, size=spatial_size)
        #print(len(spatial_features))
        # Apply color_hist() to get color histogram features
        hist_features = color_hist(feature_image, nbins=hist_bins, bins_range=hist_range)
        #print(len(hist_features))

        #print('spatial_features type', type(spatial_features))
        #print('hist_features type', type(hist_features))

        # Append the new feature vector to the features list
        features.append(np.concatenate((spatial_features, hist_features)))
    # Return list of feature vectors
    return features

In [72]: car_features = extract_features(car_images, cspace='RGB', spatial_size=(32, 32),
                                         hist_bins=32, hist_range=(0, 256))
notcar_features = extract_features(noncar_images, cspace='RGB', spatial_size=(32, 32),
                                   hist_bins=32, hist_range=(0, 256))

print(len(car_features))

```

```

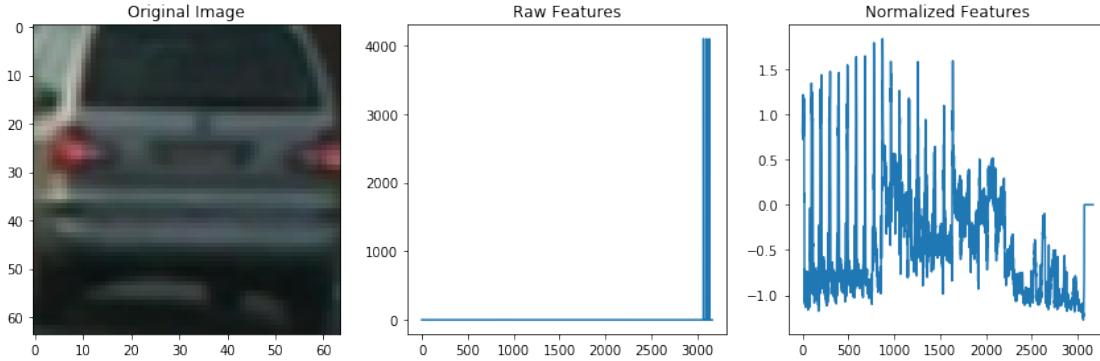
print(len(notcar_features))
car_features = np.asarray(car_features)
notcar_features = np.asarray(notcar_features)
print(len(car_features[0]))
print((car_features[0].shape))
print((notcar_features[0].shape))
'''print('=====')'''

for i in range(len(car_features[0])):
    print(car_features[0][i])
    print()
print('=====')'''

if len(car_features) > 0:
    # Create an array stack of feature vectors
    X = np.vstack((car_features, notcar_features)).astype(np.float64)
    # Fit a per-column scaler
    X_scaler = StandardScaler().fit(X)
    # Apply the scaler to X
    scaled_X = X_scaler.transform(X)
    car_ind = np.random.randint(0, len(car_images))
    # Plot an example of raw and scaled features
    fig = plt.figure(figsize=(12,4))
    plt.subplot(131)
    plt.imshow(mpimg.imread(car_images[car_ind]))
    plt.title('Original Image')
    plt.subplot(132)
    plt.plot(X[car_ind])
    plt.title('Raw Features')
    plt.subplot(133)
    plt.plot(scaled_X[car_ind])
    plt.title('Normalized Features')
    fig.tight_layout()
else:
    print('Your function only returns empty feature vectors...')

8792
8968
3168
(3168,)
(3168,)


```



1.6 Step 0: Exploration of methods: Prediction using color histograms and spatial binning and LinearSVC as classifier

```
In [73]: spatial = 32
histbin = 32

# Create an array stack of feature vectors
X = np.vstack((car_features, notcar_features)).astype(np.float64)

# Define the labels vector
y = np.hstack((np.ones(len(car_features)), np.zeros(len(notcar_features))))

# Split up data into randomized training and test sets
rand_state = np.random.randint(0, 100)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=rand_state)

# Fit a per-column scaler only on the training data
X_scaler = StandardScaler().fit(X_train)
# Apply the scaler to X_train and X_test
X_train = X_scaler.transform(X_train)
X_test = X_scaler.transform(X_test)

print('Using spatial binning of:', spatial, 'and', histbin, 'histogram bins')
print('Feature vector length:', len(X_train[0]))
# Use a linear SVC
svc = LinearSVC()
# Check the training time for the SVC
t=time.time()
svc.fit(X_train, y_train)
t2 = time.time()
print(round(t2-t, 2), 'Seconds to train SVC...')

# Check the score of the SVC
print('Test Accuracy of SVC = ', round(svc.score(X_test, y_test), 4))
```

```

# Check the prediction time for a single sample
t=time.time()
n_predict = 10
print('My SVC predicts: ', svc.predict(X_test[0:n_predict]))
print('For these',n_predict, 'labels: ', y_test[0:n_predict])
t2 = time.time()
print(round(t2-t, 5), 'Seconds to predict', n_predict,'labels with SVC')

```

Using spatial binning of: 32 and 32 histogram bins
Feature vector length: 3168
49.37 Seconds to train SVC...
Test Accuracy of SVC = 0.9122
My SVC predicts: [1. 1. 1. 0. 0. 0. 1. 0. 1. 0.]
For these 10 labels: [1. 1. 1. 0. 0. 0. 0. 0. 1. 0.]
0.00283 Seconds to predict 10 labels with SVC

1.7 Step 0: Exploration of methods: Using HOG for feature extraction

```

In [74]: # Define a function to return HOG features and visualization
def get_hog_features(img, orient, pix_per_cell, cell_per_block,
                     vis=False, feature_vec=True):
    # Call with two outputs if vis==True
    if vis == True:
        try:
            features, hog_image = hog(img, orientations=orient, pixels_per_cell=(pix_per_cell,
                                                                           cells_per_block=(cell_per_block, cell_per_block), block_norm='L2-Hausdorff',
                                                                           transform_sqrt=True,
                                                                           visualise=vis, feature_vector=feature_vec)
        except cv2.error as e:
            print('unable to obtain hog features')
            raise
        return features, hog_image
    # Otherwise call with one output
    else:
        try:
            features = hog(img, orientations=orient, pixels_per_cell=(pix_per_cell, pix_per_cell),
                           cells_per_block=(cell_per_block, cell_per_block), block_norm='L2-Hausdorff',
                           transform_sqrt=True,
                           visualise=vis, feature_vector=feature_vec)
        except cv2.error as e:
            print('unable to obtain hog features')
            raise

    return features

# Define a function to extract features from a list of images
# Have this function call bin_spatial() and color_hist()

```

```

def extract_features(imgs, cspace='RGB', orient=9,
                     pix_per_cell=8, cell_per_block=2, hog_channel=1):
    # Create a list to append feature vectors to
    features = []
    # Iterate through the list of images
    for file in imgs:
        # Read in each one by one
        image = mpimg.imread(file)
        # apply color conversion if other than 'RGB'
        if cspace != 'RGB':
            if cspace == 'HSV':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
            elif cspace == 'LUV':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2LUV)
            elif cspace == 'HLS':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2HLS)
            elif cspace == 'YUV':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2YUV)
            elif cspace == 'YCrCb':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2YCrCb)
            else: feature_image = np.copy(image)

        # Call get_hog_features() with vis=False, feature_vec=True
        if hog_channel == 'ALL':
            hog_features = []
            for channel in range(feature_image.shape[2]):
                hog_features.append(get_hog_features(feature_image[:, :, channel],
                                                     orient, pix_per_cell, cell_per_block,
                                                     vis=False, feature_vec=True))
            hog_features = np.ravel(hog_features)
        else:
            hog_features = get_hog_features(feature_image[:, :, hog_channel], orient,
                                             pix_per_cell, cell_per_block, vis=False, feature_vec=True)
        # Append the new feature vector to the features list
        features.append(hog_features)
    # Return list of feature vectors
    return features

```

1.8 Step 0: Exploration of methods: Using HOG and Linear SVC for prediction

```

In [75]: # Reduce the sample size because HOG features are slow to compute
          # The quiz evaluator times out after 13s of CPU time
sample_size = 500
cars = car_images
notcars = noncar_images

### TODO: Tweak these parameters and see how the results change.
colorspace = 'RGB' # Can be RGB, HSV, LUV, HLS, YUV, YCrCb

```

```

orient = 9
pix_per_cell = 8
cell_per_block = 2
hog_channel = 0 # Can be 0, 1, 2, or "ALL"

t=time.time()
car_features = extract_features(cars, cspace=colorspace, orient=orient,
                                 pix_per_cell=pix_per_cell, cell_per_block=cell_per_block,
                                 hog_channel=hog_channel)
notcar_features = extract_features(notcars, cspace=colorspace, orient=orient,
                                    pix_per_cell=pix_per_cell, cell_per_block=cell_per_block,
                                    hog_channel=hog_channel)
t2 = time.time()
print(round(t2-t, 2), 'Seconds to extract HOG features...')

# Create an array stack of feature vectors
X = np.vstack((car_features, notcar_features)).astype(np.float64)

# Define the labels vector
y = np.hstack((np.ones(len(car_features)), np.zeros(len(notcar_features)))))

# Split up data into randomized training and test sets
rand_state = np.random.randint(0, 100)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=rand_state)

# Fit a per-column scaler
X_scaler = StandardScaler().fit(X_train)
# Apply the scaler to X
X_train = X_scaler.transform(X_train)
X_test = X_scaler.transform(X_test)

print('Using:',orient,'orientations',pix_per_cell,
      'pixels per cell and', cell_per_block,'cells per block')
print('Feature vector length:', len(X_train[0]))
# Use a linear SVC
svc = LinearSVC()
# Check the training time for the SVC
t=time.time()
svc.fit(X_train, y_train)
t2 = time.time()
print(round(t2-t, 2), 'Seconds to train SVC...')

# Check the score of the SVC
print('Test Accuracy of SVC = ', round(svc.score(X_test, y_test), 4))
# Check the prediction time for a single sample
t=time.time()
n_predict = 10
print('My SVC predicts: ', svc.predict(X_test[0:n_predict]))

```

```

print('For these',n_predict, 'labels: ', y_test[0:n_predict])
t2 = time.time()
print(round(t2-t, 5), 'Seconds to predict', n_predict,'labels with SVC')

41.84 Seconds to extract HOG features...
Using: 9 orientations 8 pixels per cell and 2 cells per block
Feature vector length: 1764
9.27 Seconds to train SVC...
Test Accuracy of SVC = 0.9396
My SVC predicts: [ 0.  1.  1.  0.  1.  1.  0.  1.  1.  1.]
For these 10 labels: [ 0.  1.  0.  0.  1.  1.  0.  1.  1.  0.]
0.00255 Seconds to predict 10 labels with SVC

```

1.9 Step 0: Sliding window Implementation

In [76]: `image = mpimg.imread('/home/gvenkat/Downloads/bbox-example-image.jpg')`

```

In [77]: def slide_window(img, x_start_stop=[None, None], y_start_stop=[None, None],
                      xy_window=(64, 64), xy_overlap=(0.5, 0.5)):
    # If x and/or y start/stop positions not defined, set to image size
    if x_start_stop[0] == None:
        x_start_stop[0] = 0
    if x_start_stop[1] == None:
        x_start_stop[1] = img.shape[1]
    if y_start_stop[0] == None:
        y_start_stop[0] = 0
    if y_start_stop[1] == None:
        y_start_stop[1] = img.shape[0]
    # Compute the span of the region to be searched
    xspan = x_start_stop[1] - x_start_stop[0]
    yspan = y_start_stop[1] - y_start_stop[0]
    # Compute the number of pixels per step in x/y
    nx_pix_per_step = np.int(xy_window[0]*(1 - xy_overlap[0]))
    ny_pix_per_step = np.int(xy_window[1]*(1 - xy_overlap[1]))
    # Compute the number of windows in x/y
    nx_buffer = np.int(xy_window[0]*(xy_overlap[0]))
    ny_buffer = np.int(xy_window[1]*(xy_overlap[1]))
    nx_windows = np.int((xspan-nx_buffer)/nx_pix_per_step)
    ny_windows = np.int((yspan-ny_buffer)/ny_pix_per_step)
    # Initialize a list to append window positions to
    window_list = []
    # Loop through finding x and y window positions
    # Note: you could vectorize this step, but in practice
    # you'll be considering windows one by one with your
    # classifier, so looping makes sense
    for ys in range(ny_windows):
        for xs in range(nx_windows):
            # Extract HOG for window in img
           hog = hog_image(img, window=(xs, ys), Feature_vec=True)
            # Append window position to list
            window_list.append((xs, ys, hog))

```

```

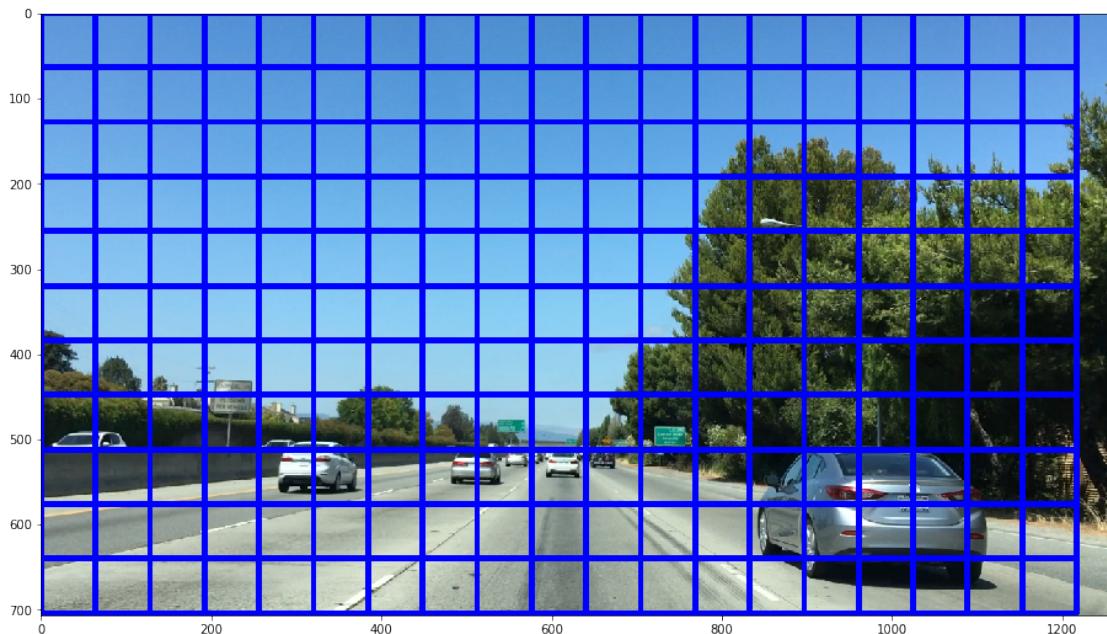
# Calculate window position
startx = xs*nx_pix_per_step + x_start_stop[0]
endx = startx + xy_window[0]
starty = ys*ny_pix_per_step + y_start_stop[0]
endy = starty + xy_window[1]
# Append window position to list
window_list.append(((startx, starty), (endx, endy)))
# Return the list of windows
return window_list

```

In [78]: windows = slide_window(image, x_start_stop=[None, None], y_start_stop=[None, None],
 xy_window=(128, 128), xy_overlap=(0.5, 0.5))

window_img = draw_boxes(image, windows, color=(0, 0, 255), thick=6)
plt.figure(figsize=(16,9))
plt.imshow(window_img)

Out[78]: <matplotlib.image.AxesImage at 0x7f911595a160>



1.10 Step 0: Finalizing all the methods for pipeline

In [79]: # Define a function to return HOG features and visualization
def get_hog_features(img, orient, pix_per_cell, cell_per_block,
 vis=False, feature_vec=True):
Call with two outputs if vis==True
if vis == True:
 try:

```

        features, hog_image = hog(img, orientations=orient,
                                   pixels_per_cell=(pix_per_cell, pix_per_cell),
                                   block_norm= 'L2-Hys',
                                   cells_per_block=(cell_per_block, cell_per_block),
                                   transform_sqrt=True,
                                   visualise=vis, feature_vector=feature_vec)

    except cv2.error as e:
        print('unable to obtain hog features')
        raise

    return features, hog_image
# Otherwise call with one output
else:
    try:
        features = hog(img, orientations=orient,
                       pixels_per_cell=(pix_per_cell, pix_per_cell),
                       cells_per_block=(cell_per_block, cell_per_block),
                       block_norm= 'L2-Hys',
                       transform_sqrt=False,
                       visualise=vis, feature_vector=feature_vec)

    except cv2.error as e:
        print('unable to obtain hog features')
        raise

return features

# Define a function to compute binned color features
def bin_spatial(img, size=(32, 32)):
    # Use cv2.resize().ravel() to create the feature vector
    color1 = cv2.resize(img[:, :, 0], size).ravel()
    color2 = cv2.resize(img[:, :, 1], size).ravel()
    color3 = cv2.resize(img[:, :, 2], size).ravel()
    return np.hstack((color1, color2, color3))

# Define a function to compute color histogram features
# NEED TO CHANGE bins_range if reading .png files with mpimg!
def color_hist(img, nbins=32, bins_range=(0, 256)):
    # Compute the histogram of the color channels separately
    channel1_hist = np.histogram(img[:, :, 0], bins=nbins, range=bins_range)
    channel2_hist = np.histogram(img[:, :, 1], bins=nbins, range=bins_range)
    channel3_hist = np.histogram(img[:, :, 2], bins=nbins, range=bins_range)
    # Concatenate the histograms into a single feature vector
    hist_features = np.concatenate((channel1_hist[0], channel2_hist[0], channel3_hist[0]))
    # Return the individual histograms, bin_centers and feature vector
    return hist_features

# Define a function to extract features from a list of images
# Have this function call bin_spatial() and color_hist()

```

```

def extract_features(imgs, color_space='RGB', spatial_size=(32, 32),
                     hist_bins=32, orient=9,
                     pix_per_cell=8, cell_per_block=2, hog_channel=0,
                     spatial_feat=True, hist_feat=True, hog_feat=True):
    # Create a list to append feature vectors to
    features = []
    # Iterate through the list of images
    for file in imgs:
        file_features = []
        # Read in each one by one
        image = mpimg.imread(file)
        # apply color conversion if other than 'RGB'
        if color_space != 'RGB':
            if color_space == 'HSV':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
            elif color_space == 'LUV':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2LUV)
            elif color_space == 'HLS':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2HLS)
            elif color_space == 'YUV':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2YUV)
            elif color_space == 'YCrCb':
                feature_image = cv2.cvtColor(image, cv2.COLOR_RGB2YCrCb)
            else: feature_image = np.copy(image)

        if spatial_feat == True:
            spatial_features = bin_spatial(feature_image, size=spatial_size)
            file_features.append(spatial_features)
        if hist_feat == True:
            # Apply color_hist()
            hist_features = color_hist(feature_image, nbins=hist_bins)
            file_features.append(hist_features)
        if hog_feat == True:
            # Call get_hog_features() with vis=False, feature_vec=True
            if hog_channel == 'ALL':
                hog_features = []
                for channel in range(feature_image.shape[2]):
                    hog_features.append(get_hog_features(feature_image[:, :, channel],
                                                         orient, pix_per_cell, cell_per_block,
                                                         vis=False, feature_vec=True))
                hog_features = np.ravel(hog_features)
            else:
                hog_features = get_hog_features(feature_image[:, :, hog_channel], orient,
                                                pix_per_cell, cell_per_block, vis=False, feature_vec=True)
            # Append the new feature vector to the features list
            file_features.append(hog_features)
        features.append(np.concatenate(file_features))
    # Return list of feature vectors

```

```

    return features

# Define a function that takes an image,
# start and stop positions in both x and y,
# window size (x and y dimensions),
# and overlap fraction (for both x and y)
def slide_window(img, x_start_stop=[None, None], y_start_stop=[None, None],
                 xy_window=(64, 64), xy_overlap=(0.5, 0.5)):
    # If x and/or y start/stop positions not defined, set to image size
    if x_start_stop[0] == None:
        x_start_stop[0] = 0
    if x_start_stop[1] == None:
        x_start_stop[1] = img.shape[1]
    if y_start_stop[0] == None:
        y_start_stop[0] = 0
    if y_start_stop[1] == None:
        y_start_stop[1] = img.shape[0]
    # Compute the span of the region to be searched
    xspan = x_start_stop[1] - x_start_stop[0]
    yspan = y_start_stop[1] - y_start_stop[0]
    # Compute the number of pixels per step in x/y
    nx_pix_per_step = np.int(xy_window[0]*(1 - xy_overlap[0]))
    ny_pix_per_step = np.int(xy_window[1]*(1 - xy_overlap[1]))
    # Compute the number of windows in x/y
    nx_buffer = np.int(xy_window[0]*(xy_overlap[0]))
    ny_buffer = np.int(xy_window[1]*(xy_overlap[1]))
    nx_windows = np.int((xspan-nx_buffer)/nx_pix_per_step)
    ny_windows = np.int((yspan-ny_buffer)/ny_pix_per_step)
    # Initialize a list to append window positions to
    window_list = []
    # Loop through finding x and y window positions
    # Note: you could vectorize this step, but in practice
    # you'll be considering windows one by one with your
    # classifier, so looping makes sense
    for ys in range(ny_windows):
        for xs in range(nx_windows):
            # Calculate window position
            startx = xs*nx_pix_per_step + x_start_stop[0]
            endx = startx + xy_window[0]
            startY = ys*ny_pix_per_step + y_start_stop[0]
            endY = startY + xy_window[1]

            # Append window position to list
            window_list.append(((startx, startY), (endx, endY)))
    # Return the list of windows
    return window_list

# Define a function to draw bounding boxes

```

```

def draw_boxes(img, bboxes, color=(0, 0, 255), thick=6):
    # Make a copy of the image
    imcopy = np.copy(img)
    # Iterate through the bounding boxes
    for bbox in bboxes:
        # Draw a rectangle given bbox coordinates
        cv2.rectangle(imcopy, bbox[0], bbox[1], color, thick)
    # Return the image copy with boxes drawn
    return imcopy

In [80]: def single_img_features(img, color_space='RGB', spatial_size=(32, 32),
                           hist_bins=32, orient=9,
                           pix_per_cell=8, cell_per_block=2, hog_channel=0,
                           spatial_feat=True, hist_feat=True, hog_feat=True):
    #1) Define an empty list to receive features
    img_features = []
    #2) Apply color conversion if other than 'RGB'
    if color_space != 'RGB':
        if color_space == 'HSV':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)
        elif color_space == 'LUV':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2LUV)
        elif color_space == 'HLS':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2HLS)
        elif color_space == 'YUV':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2YUV)
        elif color_space == 'YCrCb':
            feature_image = cv2.cvtColor(img, cv2.COLOR_RGB2YCrCb)
    else: feature_image = np.copy(img)
    #3) Compute spatial features if flag is set
    if spatial_feat == True:
        spatial_features = bin_spatial(feature_image, size=spatial_size)
        #4) Append features to list
        img_features.append(spatial_features)
    #5) Compute histogram features if flag is set
    if hist_feat == True:
        hist_features = color_hist(feature_image, nbins=hist_bins)
        #6) Append features to list
        img_features.append(hist_features)
    #7) Compute HOG features if flag is set
    if hog_feat == True:
        if hog_channel == 'ALL':
            hog_features = []
            for channel in range(feature_image.shape[2]):
                hog_features.extend(get_hog_features(feature_image[:, :, channel],
                                                     orient, pix_per_cell, cell_per_block,
                                                     vis=False, feature_vec=True))
        else:

```

```

        hog_features = get_hog_features(feature_image[:, :, hog_channel], orient,
                                         pix_per_cell, cell_per_block, vis=False, feature_vec=True)
    #8) Append features to list
    img_features.append(hog_features)

    #9) Return concatenated array of features
    return np.concatenate(img_features)

In [81]: # Define a function you will pass an image
         # and the list of windows to be searched (output of slide_windows())
def search_windows(img, windows, clf, scaler, color_space='RGB',
                    spatial_size=(32, 32), hist_bins=32,
                    hist_range=(0, 256), orient=9,
                    pix_per_cell=8, cell_per_block=2,
                    hog_channel=0, spatial_feat=True,
                    hist_feat=True, hog_feat=True):

    #1) Create an empty list to receive positive detection windows
    on_windows = []
    #2) Iterate over all windows in the list
    for window in windows:
        #3) Extract the test window from original image
        test_img = cv2.resize(img[window[0][1]:window[1][1], window[0][0]:window[1][0]])
        #4) Extract features for that window using single_img_features()
        features = single_img_features(test_img, color_space=color_space,
                                        spatial_size=spatial_size, hist_bins=hist_bins,
                                        orient=orient, pix_per_cell=pix_per_cell,
                                        cell_per_block=cell_per_block,
                                        hog_channel=hog_channel, spatial_feat=spatial_feat,
                                        hist_feat=hist_feat, hog_feat=hog_feat)
        #5) Scale extracted features to be fed to classifier
        test_features = scaler.transform(np.array(features).reshape(1, -1))
        #6) Predict using your classifier
        prediction = clf.predict(test_features)
        #7) If positive (prediction == 1) then save the window
        if prediction == 1:
            on_windows.append(window)
    #8) Return windows for positive detections
    return on_windows

def convert_color(img, conv='RGB2YCrCb'):
    #print(type(img))
    if conv == 'RGB2YCrCb':
        #print('here')
        return cv2.cvtColor(img, cv2.COLOR_RGB2YCrCb)
    if conv == 'BGR2YCrCb':
        return cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb)
    if conv == 'RGB2LUV':

```

```
        return cv2.cvtColor(img, cv2.COLOR_RGB2LUV)
```

1.11 Step1: Perform a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set of images and train a classifier Linear SVM classifier

```
In [82]: ## Reduce the sample size because
# The quiz evaluator times out after 13s of CPU time
sample_size = 500
cars = car_images
notcars = noncar_images

orient_s = [9, 11, 13, 19, 31]
pix_per_cell_s = [8, 16, 6, ]
### TODO: Tweak these parameters and see how the results change.
color_space = 'YCrCb' # Can be RGB, HSV, LUV, HLS, YUV, YCrCb
orient = 31 # HOG orientations
pix_per_cell = 5 # HOG pixels per cell
cell_per_block = 2 # HOG cells per block
hog_channel = 'ALL' # Can be 0, 1, 2, or "ALL"
spatial_size = (32, 32) # Spatial binning dimensions
hist_bins = 32 # Number of histogram bins
spatial_feat = False # Spatial features on or off
hist_feat = False
hog_feat = True # HOG features on or off
y_start_stop = [None, None] # Min and max in y to search in slide_window()

print("Extracting features...")

t0 = time.time()
car_features = extract_features(cars, color_space=color_space,
                                 spatial_size=spatial_size, hist_bins=hist_bins,
                                 orient=orient, pix_per_cell=pix_per_cell,
                                 cell_per_block=cell_per_block,
                                 hog_channel=hog_channel, spatial_feat=spatial_feat,
                                 hist_feat=hist_feat, hog_feat=hog_feat)
t1 = time.time()
print("Extracting car features took ", round(t1-t0, 4), " seconds")
notcar_features = extract_features(notcars, color_space=color_space,
                                    spatial_size=spatial_size, hist_bins=hist_bins,
                                    orient=orient, pix_per_cell=pix_per_cell,
                                    cell_per_block=cell_per_block,
                                    hog_channel=hog_channel, spatial_feat=spatial_feat,
                                    hist_feat=hist_feat, hog_feat=hog_feat)
t2 = time.time()
print("Extracting noncar features took ", round(t2-t1, 4), " seconds")

# Create an array stack of feature vectors
X = np.vstack((car_features, notcar_features)).astype(np.float64)
```

```

# Define the labels vector
y = np.hstack((np.ones(len(car_features)), np.zeros(len(notcar_features)))))

# Split up data into randomized training and test sets
rand_state = np.random.randint(0, 100)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=rand_state)

# Fit a per-column scaler
X_scaler = StandardScaler().fit(X_train)
# Apply the scaler to X
X_train = X_scaler.transform(X_train)
X_test = X_scaler.transform(X_test)

print('Using:', orient, 'orientations', pix_per_cell,
      'pixels per cell and', cell_per_block, 'cells per block')
print('Feature vector length:', len(X_train[0]))
# Use a linear SVC
svc = LinearSVC()
# Check the training time for the SVC
t=time.time()
svc.fit(X_train, y_train)
t2 = time.time()
print(round(t2-t, 2), 'Seconds to train SVC...')
# Check the score of the SVC
print('Test Accuracy of SVC = ', round(svc.score(X_test, y_test), 4))
# Check the prediction time for a single sample
t = time.time()

# Saving the model to a file

with open('model.p', 'wb') as file:
    pickle.dump(svc, file)

```

Extracting features...
Extracting car features took 114.3593 seconds
Extracting noncar features took 115.5019 seconds
Using: 31 orientations 5 pixels per cell and 2 cells per block
Feature vector length: 45012
31.55 Seconds to train SVC...
Test Accuracy of SVC = 0.9803

In [83]: image = mpimg.imread('test_images/test1.jpg')
draw_image = np.copy(image)

Uncomment the following line if you extracted training

```

# data from .png images (scaled 0 to 1 by mpimg) and the
# image you are searching is a .jpg (scaled 0 to 255)
image = image.astype(np.float32)/255

windows = slide_window(image, x_start_stop=[None, None], y_start_stop=y_start_stop,
                       xy_window=(96, 96), xy_overlap=(0.5, 0.5))

hot_windows = search_windows(image, windows, svc, X_scaler, color_space=color_space,
                             spatial_size=spatial_size, hist_bins=hist_bins,
                             orient=orient, pix_per_cell=pix_per_cell,
                             cell_per_block=cell_per_block,
                             hog_channel=hog_channel, spatial_feat=spatial_feat,
                             hist_feat=hist_feat, hog_feat=hog_feat)

window_img = draw_boxes(draw_image, hot_windows, color=(0, 0, 255), thick=6)

plt.imshow(window_img)

```

Out[83]: <matplotlib.image.AxesImage at 0x7f911530fac8>



In [84]: # Define a single function that can extract features using hog sub-sampling and make predictions
find_car_debug = 0
def find_cars(img, ystart, ystop, scale, svc, X_scaler, orient, pix_per_cell, cell_per_block):
 rectangles = []
 if find_car_debug:
 find_cars_start_time = time.time()
 total_rectangle_draw_time = 0
 preprocessing_time = 0

```

    preprocessing_start_time = time.time()

    draw_img = np.copy(img)
    img = img.astype(np.float32)/255
    #print(ystart, ystop)
    img_tosearch = img[ystart:ystop,:,:]
    #plt.figure(figsize=(16,9))
    #plt.imshow(img_tosearch)

    #print(type(img_tosearch))

    ctrans_tosearch = convert_color(img_tosearch, conv='RGB2YCrCb')
    #print((ctrans_tosearch))
    print('Shape of image: ', ctrans_tosearch.shape)

    #print(type(ctrans_tosearch))
    #ctrans_tosearch = img_tosearch
    if scale != 1:
        imshape = ctrans_tosearch.shape
        ctrans_tosearch = cv2.resize(ctrans_tosearch, (np.int(imshape[1]/scale), np.int(imshape[0]/scale)))

    ch1 = ctrans_tosearch[:, :, 0]
    ch2 = ctrans_tosearch[:, :, 1]
    ch3 = ctrans_tosearch[:, :, 2]

    if find_car_debug:
        preprocessing_end_time = time.time()
        preprocessing_time = round((preprocessing_end_time - preprocessing_start_time), 2)
        print()
        print('Preprocessing time for image', preprocessing_time)
        print()

    # Define blocks and steps as above
    nxblocks = (ch1.shape[1] // pix_per_cell) - cell_per_block + 1
    nyblocks = (ch1.shape[0] // pix_per_cell) - cell_per_block + 1
    nfeat_per_block = orient*cell_per_block**2

    if find_car_debug:
        print('Shape of scale: ', ch1.shape)
        print('NX Blocks: ', nxblocks)
        print('NY Blocks: ', nyblocks)
        print()

    # 64 was the orginal sampling rate, with 8 cells and 8 pix per cell
    window = 64

    nblocks_per_window = (window // pix_per_cell) - cell_per_block + 1
    cells_per_step = 2 # Instead of overlap, define how many cells to step
    nxsteps = (nxblocks - nblocks_per_window) // cells_per_step + 1
    nysteps = (nyblocks - nblocks_per_window) // cells_per_step + 1

```

```

if find_car_debug:
    print('n_blocks_per_window', nblocks_per_window)
    print('NX Steps: ', nxsteps)
    print('NY Steps: ', nysteps)
    print()

# Compute individual channel HOG features for the entire image
hog1 = get_hog_features(ch1, orient, pix_per_cell, cell_per_block, feature_vec=False)
hog2 = get_hog_features(ch2, orient, pix_per_cell, cell_per_block, feature_vec=False)
hog3 = get_hog_features(ch3, orient, pix_per_cell, cell_per_block, feature_vec=False)
iterations = 0
t0 = time.time()

x_positions = []
y_positions = []
x_real_positions = []
y_real_positions = []
window_sizes = []
time_at_start_for_loop = time.time()
hog_features_start_times = []
hog_features_end_times = []
hog_feature_times_per_iteration = []
test_feature_transform_times = []
for xb in range(nxsteps):
    for yb in range(nysteps):
        iterations += 1
        if find_car_debug:
            iterations += 1
            if not iterations % 1000:
                print("Iteration: ", iterations, "nx: ", xb, "ny: ", yb)
                t1 = time.time()
                print(round(t1-t0, 4), 'seconds elapsed.')
                t0 = t1
        ypos = yb*cells_per_step
        xpos = xb*cells_per_step
        x_positions.append(xpos)
        y_positions.append(ypos)
        # Extract HOG for this patch
        hog_feature_start_time = time.time()
        hog_feat1 = hog1[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window]
        hog_feat2 = hog2[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window]
        hog_feat3 = hog3[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window]
        hog_features = np.hstack((hog_feat1, hog_feat2, hog_feat3))
        #hog_features = np.hstack((hog_feat1))
        hog_feature_end_time = time.time()
        hog_feature_time = hog_feature_end_time - hog_feature_start_time
        hog_feature_times_per_iteration.append(round(hog_feature_time, 6))

```

```

xleft = xpos*pix_per_cell
ytop = ypos*pix_per_cell

# Extract the image patch
#subimg = cv2.resize(ctrans_tosearch[ytop:ytop+window, xleft:xleft+window],

# Get color features
#spatial_features = bin_spatial(subimg, size=spatial_size)
#hist_features = color_hist(subimg, nbins=hist_bins)

# Scale features and make a prediction
#test_features = X_scaler.transform(np.hstack((spatial_features, hist_features)))
if find_car_debug:
    test_feature_transform_start_time = time.time()
    #test_features = X_scaler.transform(np.hstack((hog_features)).reshape(1, -1))
    test_feature_transform_end_time = time.time()
    test_feature_transform_times.append(round((test_feature_transform_end_time - test_feature_transform_start_time)))
    #test_features = X_scaler.transform(np.hstack((shape_feat, hist_feat)).reshape(1, -1))
    #test_prediction = svc.predict(test_features) # commented

#print("test features shape", test_features.shape)
print('hog features shape', hog_features.shape)

hog_features = hog_features.reshape(1, -1)
if find_car_debug:
    print('hog features shape after reshape', hog_features.shape)

t2 = time.time()

test_prediction = svc.predict(hog_features) # added
t3 = time.time()
if find_car_debug:
    if not iterations % 1000:
        print('Prediction time: ', t1-t0, ' with prediction: ', test_prediction)
if test_prediction == 1:
    xbox_left = np.int(xleft*scale)
    ytop_draw = np.int(ytop*scale)
    win_draw = np.int(window*scale)

    if find_car_debug:
        x_real_positions.append(xbox_left)
        y_real_positions.append(ytop_draw)
        window_sizes.append(win_draw)
        rectangle_draw_start_time = time.time()
        #cv2.rectangle(draw_img, (xbox_left, ytop_draw+ystart), (xbox_left+win_draw, ytop_draw+win_draw+ystart), (0, 255, 0), 2)

    if find_car_debug:
        rectangle_draw_end_time = time.time()

```

```

        print('Time spent in drawing a rectangle', round((rectangle_draw_end_time - rectangle_start_time)))
        total_rectangle_draw_time += (rectangle_draw_end_time - rectangle_start_time)

    rectangles.append(((xbox_left, ytop_draw+ystart),(xbox_left+win_draw,yt
    if find_car_debug:
        print()
        print('X positions: ', x_positions)
        print('Len X: ', len(x_positions))
        print()
        print('Y positions: ', y_positions)
        print('Len Y: ', len(y_positions))
        print()

        print()
        print('X real positions: ', x_real_positions)
        print('Len real X: ', len(x_real_positions))
        print()
        print('Y real positions: ', y_real_positions)
        print('Len real Y: ', len(y_real_positions))
        print()
        print('Window sizes: ', window_sizes)
        print('Len Window sizes: ', len(window_sizes))
        print()

        print(hog_feature_times_per_iteration)
        print('Sum of hog times: ', sum(hog_feature_times_per_iteration))
        print()

        print("Test features transform time: ", sum(test_feature_transform_times))
        print()

    time_at_end_for_loop = time.time()
    #print('Time spent in for loop', round((time_at_end_for_loop - time_at_start_for_loop)))
    #print('Average time per iteration', round((time_at_end_for_loop - time_at_start_for_loop)/len(x_positions)))
    if find_car_debug:
        print('Time spent in for loop', round((time_at_end_for_loop - time_at_start_for_loop)))
        print()
        print("Total time spent in drawing cars: ", round(total_rectangle_draw_time, 6))
        print()

    find_cars_end_time = time.time()
    if find_car_debug:
        print('Total time spent in find cars', round((find_cars_end_time - find_cars_start_time)))
        print()

return draw_img, rectangles

```

```
In [93]: # Define a single function that can extract features using hog sub-sampling and make pr
find_car_debug = 1
find_car_debug2 = 0
window = 64
cells_per_step = 4 # Instead of overlap, define how many cells to step

def find_cars_new(img, ystart, ystop, xstart, xstop, scale, window, cells_per_step,\n                 svc, X_scaler, orient, pix_per_cell, cell_per_block, spatial_size, hi
hog_parallelism_enabled = 0
time.sleep(0.5*random.random())
#print(process_name + ', in the called function')
rectangles = []
#print(process_name + 'Xstart', xstart, 'Xstop:', xstop, 'Y start:', ystart, 'Y sto
find_cars_start_time = time.time()
#import pdb; pdb.set_trace()

if xstart > 1280 - window*scale or xstop > 1280 or ystart > 660 - window*scale or y
    print(process_name + ': returning for wrong sizes')
    return rectangles

#print(process_name + ', entering step 1...')

#draw_img = np.copy(img)
# Step 1: Divide with 255
step1_start_time = time.time()
try:
    img = img.astype(np.float32)/255
except:
    print("Step 1: failed... could not initialize image as float!")
    raise

step1_end_time = time.time()
#print(process_name + ', exiting step 1...')

step1_time = round(step1_end_time - step1_start_time, 6)

# Step 2: Resize if scale is not 1
#print(process_name + ', entering step 2...')

step2_start_time = time.time()

if xstop - xstart < scale*window or ystop - ystart < scale*window:
    print(process_name + ': step2: returning for bad sizes')
    print(process_name + ':Step2: Xstart', xstart, 'Xstop:', xstop, 'Y start:', yst
    print("Step 2: failed... inappropriate start and stop sizes!")
    raise
```

```

try:
    img_tosearch = img[ystart:ystop,xstart:xstop]
except:
    raise

#print(process_name + ', patch shape...', img_tosearch.shape)
try:
    ctrans_tosearch = convert_color(img_tosearch, conv='RGB2YCrCb')
except:
    raise
#print(process_name + ', patch shape post color conversion...', ctrans_tosearch.shape)

try:
    if scale != 1:
        imshape = ctrans_tosearch.shape
        ctrans_tosearch = cv2.resize(ctrans_tosearch, (np.int(imshape[1]/scale), np.int(imshape[0]/scale)))
except:
    raise

#print(process_name + ', patch shape post resize...', ctrans_tosearch.shape)
step2_end_time = time.time()
#print(process_name + ', exiting step 2...')

step2_time = round(step2_end_time - step2_start_time, 6)

# Step 3: Get HOG channels
#print(process_name + ', entering step 3...')
step3_start_time = time.time()

try:
    if hog_channel == 'ALL':
        ch1 = ctrans_tosearch[:, :, 0]
        ch2 = ctrans_tosearch[:, :, 1]
        ch3 = ctrans_tosearch[:, :, 2]
    else:
        ch1 = ctrans_tosearch[:, :, hog_channel]
except:
    raise

step3_end_time = time.time()
#print(process_name + ', exiting step 3...')

step3_time = round(step3_end_time - step3_start_time, 6)

# Step 4: Define blocks and steps as above
#print(process_name + ', entering step 4...')
#print(process_name + ', channel shape st begin of step4...', ch1.shape)

```

```

step4_start_time = time.time()

nxblocks = (ch1.shape[1] // pix_per_cell) - cell_per_block + 1
nyblocks = (ch1.shape[0] // pix_per_cell) - cell_per_block + 1

nfeat_per_block = orient*cell_per_block**2

nblocks_per_window = (window // pix_per_cell) - cell_per_block + 1

nxsteps = (nxblocks - nblocks_per_window) // cells_per_step + 1
nysteps = (nyblocks - nblocks_per_window) // cells_per_step + 1

#print(process_name + ', step4, nx blocks', nxblocks)
#print(process_name + ', step4, ny blocks', nyblocks)
#print(process_name + ', step4, nx steps', nxsteps)
#print(process_name + ', step4, ny steps', nysteps)
#print(process_name + ', step4, nblocks_per_window', nblocks_per_window)

step4_end_time = time.time()
#print(process_name + ', exiting step 4...')

step4_time = round(step4_end_time - step4_start_time, 6)

# Step 5: Compute individual channel HOG features for the entire image
#print(process_name + ', entering step 5...')

step5_start_time = time.time()

if hog_channel == 'ALL':
    if hog_parallelism_enabled:
        pool = mp.Pool(processes=3)
        channels = [ch1, ch2, ch3]
        feature_vec = False
        hog = [pool.apply(get_hog_features, args=(ch, orient, pix_per_cell, cell_per_block,
                                                    for ch in channels)]
        #print(hog)
        hog1 = hog[0]
        hog2 = hog[1]
        hog3 = hog[2]
        #hog1_1 = get_hog_features(ch1, orient, pix_per_cell, cell_per_block, feature_vec)
        #hog2_1 = get_hog_features(ch2, orient, pix_per_cell, cell_per_block, feature_vec)
        #hog3_1 = get_hog_features(ch3, orient, pix_per_cell, cell_per_block, feature_vec)

        #print(hog1_1.shape, hog1.shape)

        print('channel shape', channels[0].shape)
        print(nyblocks, nxblocks, cell_per_block, cell_per_block, orient)
        hog1 = hog[0].reshape(nyblocks, nxblocks, cell_per_block, cell_per_block, orient)
        hog2 = hog[1].reshape(nyblocks, nxblocks, cell_per_block, cell_per_block, orient)
        hog3 = hog[2].reshape(nyblocks, nxblocks, cell_per_block, cell_per_block, orient)

```

```

        hog3 = hog[2].reshape(nyblocks, nxblocks, cell_per_block, cell_per_block, cell_per_block)
    else:
        hog1 = get_hog_features(ch1, orient, pix_per_cell, cell_per_block, feature_vec)
        hog2 = get_hog_features(ch2, orient, pix_per_cell, cell_per_block, feature_vec)
        hog3 = get_hog_features(ch3, orient, pix_per_cell, cell_per_block, feature_vec)
    else:
        hog1 = get_hog_features(ch1, orient, pix_per_cell, cell_per_block, feature_vec)

step5_end_time = time.time()
#print(process_name + ', exiting step 5...')

step5_time = round(step5_end_time - step5_start_time, 6)

# Step 6: Misc initializations
#print(process_name + ', entering step 6...')

step6_start_time = time.time()
iterations = 0
x_positions = []
y_positions = []
x_real_positions = []
y_real_positions = []
window_sizes = []

step6_end_time = time.time()
#print(process_name + ', exiting step 6...')

step6_time = round(step6_end_time - step6_start_time, 6)

# Step 7: for loop
#print(process_name + ', entering step 7...')

step7_start_time = time.time()
for xb in range(nxsteps):
    for yb in range(nysteps):
        ypos = yb*cells_per_step
        xpos = xb*cells_per_step
        x_positions.append(xpos)
        y_positions.append(ypos)

        # Extract HOG for this patch
        hog_feat1 = hog1[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window]
        if hog_channel == 'ALL':
            hog_feat2 = hog2[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window]
            hog_feat3 = hog3[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window]
            hog_features = np.hstack((hog_feat1, hog_feat2, hog_feat3))
        else:
            hog_features = np.hstack((hog_feat1))

```

```

xleft = xpos*pix_per_cell
ytop = ypos*pix_per_cell

hog_features = hog_features.reshape(1, -1)

test_prediction = svc.predict(hog_features) # added

if test_prediction == 1:
    xbox_left = np.int(xleft*scale)
    ytop_draw = np.int(ytop*scale)
    win_draw = np.int(window*scale)

    x_real_positions.append(xbox_left)
    y_real_positions.append(ytop_draw)
    window_sizes.append(win_draw)

    #cv2.rectangle(draw_img, (xbox_left+xstart, ytop_draw+ystart), (xbox_left+win_size+xstart, ytop_draw+win_size+ystart), (0, 255, 0), 2)
    rectangles.append(((xbox_left+xstart, ytop_draw+ystart),(xbox_left+win_size+xstart, ytop_draw+win_size+ystart)))

step7_end_time = time.time()
#print(process_name + ', exiting step 7...')

step7_time = round(step7_end_time - step7_start_time, 6)

find_cars_end_time = time.time()
'''

print(process_name + ', Step 1: Divide with 255, processing time:', step1_time, 'seconds')

print(process_name + ', Step 2: Resize if scale is not 1:', step2_time, 'seconds')

print(process_name + ', Step 3: Get HOG channels:', step3_time, 'seconds')
print(process_name + ', Step 4: Define blocks and steps as above:', step4_time, 'seconds')
print(process_name + ', Step 5: Compute individual channel HOG features for the entire image:', step5_time, 'seconds')
print(process_name + ', Step 6: Misc initializations:', step6_time, 'seconds')
print(process_name + ', Step 7: for loop:', step7_time, 'seconds')
print()

'''

all_steps_time = step1_time + step2_time + step3_time + step4_time + step5_time + step6_time + step7_time
#print(process_name + 'All steps time:', all_steps_time, 'seconds')
#print()

total_find_cars_time = round(find_cars_end_time - find_cars_start_time, 6)

```

```

#print(process_name + ', Find cars processing time:', total_find_cars_time, 'second')
#print()

return (rectangles, total_find_cars_time)

In [94]: def calculate_patches(xstart, xstop, window, scale, step, cells_per_step):
    xstarts = []
    xstops = []

    step_value = int(step - window*scale + cells_per_step)
    steps = int((xstop - xstart)/(step - window*scale + cells_per_step))
    if xstop - xstart > step:
        for i in range(steps):
            if i == 0:
                xstarts.append(int(xstart))
                if xstart + step > xstop:
                    xstops.append(int(xstop))
                else:
                    xstops.append(int(xstarts[i] + step))
            else:
                xstarts.append(int(xstarts[-1] + step_value))
                if xstarts[-1] + step > xstop:
                    xstops.append(int(xstop))
                else:
                    xstops.append(int(xstarts[i] + step))
    '''if xstops[-1] < xstop:
        xstarts.append(int(xstarts[-1] + step_value))
        xstops.append(int(xstop))'''
    else:
        xstarts.append(int(xstart))
        xstops.append(int(xstop))
    #print('starts returned', xstarts)
    #print('stops returned', xstops)
    return xstarts, xstops

```

```

In [95]: import time
import random
import multiprocessing as mp
from multiprocessing import Process, Queue, current_process, freeze_support

print(mp.cpu_count())
#
# Function run by worker processes
#

def worker(iput, oput):
    #print(current_process().name + ' in worker')
    for func, args in iter(iput.get, 'STOP'):

```

```

#print(current_process().name + ' in worker for loop')
result = calculate(func, args)
#print(current_process().name + ' in worker after calculate')
output.put(result)
#print(current_process().name + ' after put result')

#
# Function used to calculate result
#

def calculate(func, args):
    #print(current_process().name + ' in calculate')
    result = func(*args, current_process().name)
    #print()
    #return '%s done at %s returns that %s = %s' % \
    #       (current_process().name, str(time.time()), func.__name__, result)
    return result

```

40

```

In [96]: #parallelizing find_cars
scales1 = [0.95]
scales2 = [1.15, 1.35, 1.55]
scales3 = [0.75, 0.95, 1.15, 1.35, 1.55]
scales4 = [1.15, 1.25, 1.35, 1.45, 1.55]
scales5 = [0.75, 0.8, 0.85, 0.9, 0.95, 1.05]
scales6 = [0.75, 0.85, 0.95, 1.05, 1.15, 1.25, 1.35, 1.45, 1.55]
scales7 = [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.10, 1.15, 1.2, 1.25, 1.3, 1.35, 1.40, 1.

scaleslist = [scales1, scales2, scales3, scales4, scales5, scales6, scales7]

use_default_number_of_processes = 0

y_start = 400
y_stop = 660
x_start = 395
x_stop = 1280
step = 260

use_sub_windows = 0
print_task_times = 0
def parallelize_find_cars(img, processes, scaleslistId=2):
    global scaleslist
    global use_default_number_of_processes
    global print_task_times
    global y_start, y_stop, x_start, x_stop
    global step

```

```

rects = []
task_times = []
#import pdb; pdb.set_trace()

TASK_LIST = []
allXstarts = []
allXstops = []
allYstarts = []
allYstops = []

removedXstarts = []
removedXstops = []
removedYstarts = []
removedYstops = []
for scale in scaleslist[scaleslistId]:
    xstarts = []
    xstops = []
    ystarts = []
    ystops = []

    if use_sub_windows:
        xstarts, xstops = calculate_patches(x_start, x_stop, window, scale, step, c)
        ystarts, ystops = calculate_patches(y_start, y_stop, window, scale, step, c)
    else:
        xstarts = [400]
        xstops = [1280]
        ystarts = [400]
        ystops = [600]

    allXstarts += xstarts
    allXstops += xstops
    allYstarts += ystarts
    allYstops += ystops
    #print('X Lengths: ', len(xstarts), len(xstops))
    #print('Y Lengths: ', len(ystarts), len(ystops))
    #print('Step:', step)
    #print('Scale', scale)
    for i in range(len(xstarts)):
        for j in range(len(ystarts)):
            TASK_LIST += [(find_cars_new, (img, ystarts[j], ystops[j], xstarts[i], xstops[i]))]
print('Length of task list: ', len(TASK_LIST))

```

```

    ...
    TASK_LIST +=[(find_cars_new, (img, ystart, ystop, xstart, xstop, scale, window,
                                print('TASK LEN per iteration', len(TASK_LIST))
                                #print(TASK_LIST[0])
    ...
    #print(allXstarts)
    #print(allXstops)
    #print(allYstarts)
    #print(allYstops)
    #print(img, ystart, ystop, xstart, xstop, scale, window, cells_per_step, suc, X_sca
    # Create queues
    task_queue = Queue()
    done_queue = Queue()

    for task in TASK_LIST:
        task_queue.put(task)

    #print('TASK LIST Length', len(TASK_LIST))
    #print(len(TASK_LIST[0][1]))
    #print()
    #print('Start of parametters')
    #for i in range(1, len(TASK_LIST[0][1])):
    #    print(TASK_LIST[0][1][i])
    #print('End of parametters')
    #print()
    #print()
    if use_default_number_of_processes:
        NUMBER_OF_PROCESSES = int(len(TASK_LIST)/2) + 1
    else:
        NUMBER_OF_PROCESSES = processes

    print('Number of processes used:', NUMBER_OF_PROCESSES)
    print()
    start_time = time.time()
    #print('starting of processes', start_time)
    for i in range(NUMBER_OF_PROCESSES):
        Process(target=worker, args=(task_queue, done_queue)).start()

    #time.sleep(6)
    end_time = time.time()
    #print('end of processes', end_time)
    total_time = end_time - start_time
    #print('total time', total_time)

    rect_retrieval_start = time.time()
    for i in range(len(TASK_LIST)):
        res = done_queue.get()

```

```

#print(res)
if res:
    rects += res[0]
    task_times.append(res[1])

rect_retrieval_end = time.time()

rect_retrieval_time = round(rect_retrieval_end - rect_retrieval_start, 4)
#print('Rect retrieval time', rect_retrieval_time)

# Tell child processes to stop
for i in range(NUMBER_OF_PROCESSES):
    task_queue.put('STOP')

if print_task_times:
    print('The times for each task are:', task_times, 'with:')
    print()
    print('Minimum:', min(task_times), 'Maximum:', max(task_times), 'Average:', \
          round(sum(task_times)/len(task_times), 4), 'seconds')
    print()

return rects

```

In [97]: img = mpimg.imread('test_images/test1.jpg')

```

plt.figure(figsize=(16,9))
plt.imshow(img)

```

Out[97]: <matplotlib.image.AxesImage at 0x7f910ff514a8>



```
In [98]: #if __name__ == '__main__':
    use_default_number_of_processes = 1
    freeze_support()
    t0 = time.time()
    r = []
    r = parallelize_find_cars(img, 2)
    t1 = time.time()
    #print()
    #print(r)
    #print()
    '''for i in range(5):
        print('Sleeping for:', (5-i), 'seconds...')
        time.sleep(1)'''
    #print()
    print('for Scale', round(t1-t0, 6))
```

```
Length of task list: 5
Number of processes used: 3
```

```
for Scale 2.501906
```

```
In [99]: print(r)
```

```
[((837, 400), (897, 460)), ((837, 419), (897, 479)), ((837, 438), (897, 498)), ((856, 400), (916,
```

```
In [100]: img_with_boxes = np.copy(img)
for box in r:
    cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
plt.figure(figsize=(16,9))
plt.imshow(img_with_boxes)
```

```
Out[100]: <matplotlib.image.AxesImage at 0x7f9149381160>
```



```
In [101]: test_images = glob.glob('test_images/test*.jpg')
        print(test_images)

['test_images/test1.jpg', 'test_images/test6.jpg', 'test_images/test2.jpg', 'test_images/test3.j
```



```
In [102]: test_images.sort()
        print(test_images)

['test_images/test1.jpg', 'test_images/test2.jpg', 'test_images/test3.jpg', 'test_images/test4.j
```



```
In [103]: # Now get the boxes for all test images
        use_default_number_of_processes = 1

        for image in test_images:
            img = mpimg.imread(image)
            freeze_support()
            t0 = time.time()
            r = []
            r = parallelize_find_cars(img, 2)
            t1 = time.time()
            print('Time taken for processing each image', round(t1-t0, 6))

            img_with_boxes = np.copy(img)
            for box in r:
                cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
```

```
plt.figure(figsize=(16,9))
plt.imshow(img_with_boxes)

Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.606093
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.493645
Length of task list: 5
Number of processes used: 3

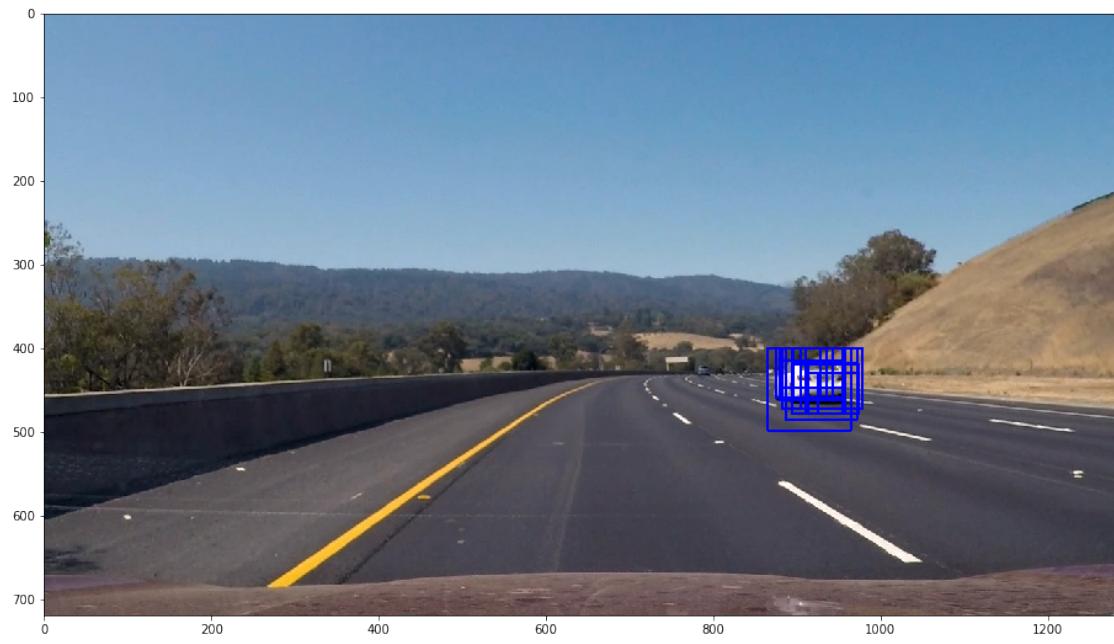
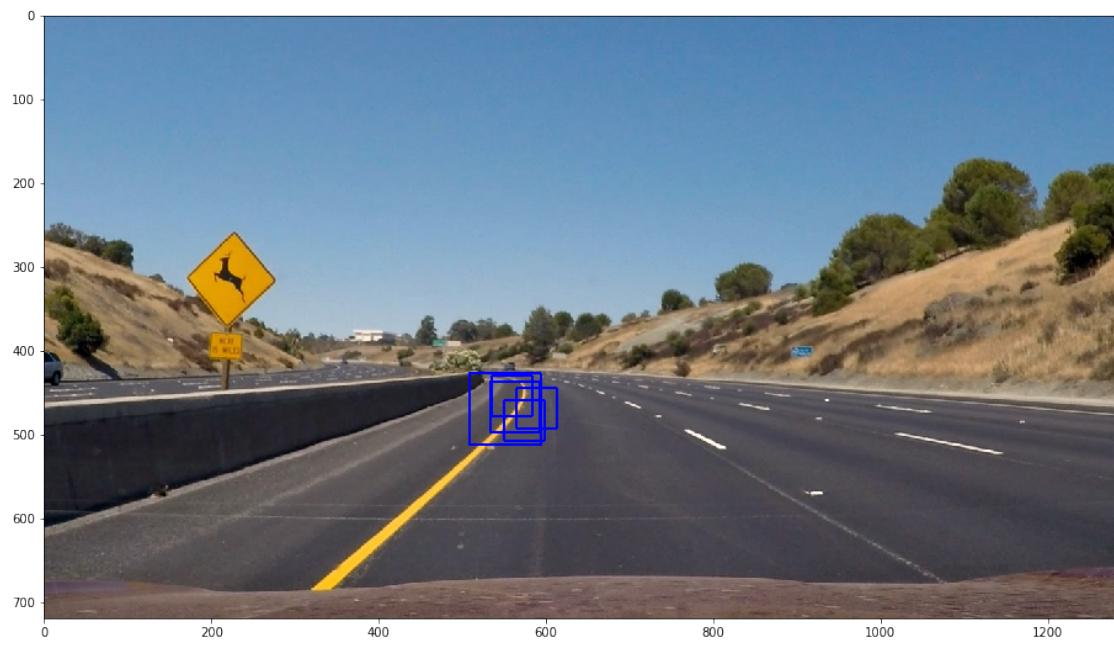
Time taken for processing each image 2.62556
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.445703
Length of task list: 5
Number of processes used: 3

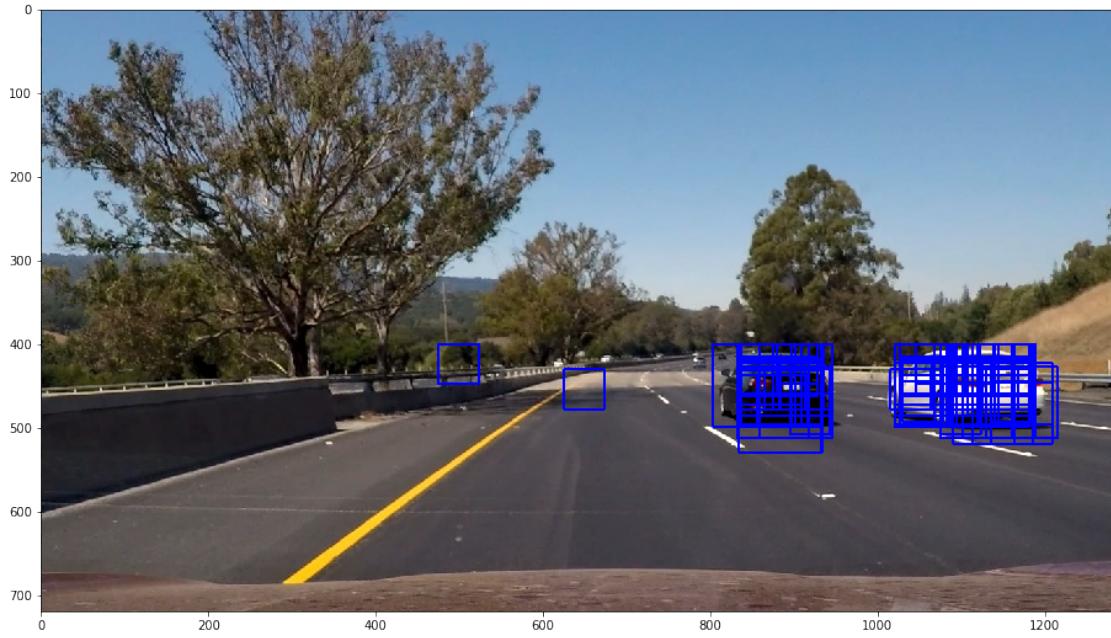
Time taken for processing each image 2.810079
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.647641
```









```
In [104]: ## Reduce the sample size because
# The quiz evaluator times out after 13s of CPU time
sample_size = 500
cars = car_images
notcars = noncar_images

### TODO: Tweak these parameters and see how the results change.
color_space = 'YCrCb' # Can be RGB, HSV, LUV, HLS, YUV, YCrCb
orient = 9 # HOG orientations
pix_per_cell = 8 # HOG pixels per cell
cell_per_block = 2 # HOG cells per block
hog_channel = 'ALL' # Can be 0, 1, 2, or "ALL"
spatial_size = (32, 32) # Spatial binning dimensions
hist_bins = 32 # Number of histogram bins
spatial_feat = False # Spatial features on or off
hist_feat = False
hog_feat = True # HOG features on or off
y_start_stop = [None, None] # Min and max in y to search in slide_window()

print("Extracting features...")

t0 = time.time()
car_features = extract_features(cars, color_space=color_space,
                                 spatial_size=spatial_size, hist_bins=hist_bins,
                                 orient=orient, pix_per_cell=pix_per_cell,
```

```

        cell_per_block=cell_per_block,
        hog_channel=hog_channel, spatial_feat=spatial_feat,
        hist_feat=hist_feat, hog_feat=hog_feat)

t1 = time.time()
print("Extracting car features took ", round(t1-t0, 4), " seconds")
notcar_features = extract_features(notcars, color_space=color_space,
                                    spatial_size=spatial_size, hist_bins=hist_bins,
                                    orient=orient, pix_per_cell=pix_per_cell,
                                    cell_per_block=cell_per_block,
                                    hog_channel=hog_channel, spatial_feat=spatial_feat,
                                    hist_feat=hist_feat, hog_feat=hog_feat)

t2 = time.time()
print("Extracting noncar features took ", round(t2-t1, 4), " seconds")

# Create an array stack of feature vectors
X = np.vstack((car_features, notcar_features)).astype(np.float64)

# Define the labels vector
y = np.hstack((np.ones(len(car_features)), np.zeros(len(notcar_features)))))

# Split up data into randomized training and test sets
rand_state = np.random.randint(0, 100)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=rand_state)

# Fit a per-column scaler
X_scaler = StandardScaler().fit(X_train)
# Apply the scaler to X
X_train = X_scaler.transform(X_train)
X_test = X_scaler.transform(X_test)

print('Using:', orient, 'orientations', pix_per_cell,
      'pixels per cell and', cell_per_block, 'cells per block')
print('Feature vector length:', len(X_train[0]))
# Use a linear SVC
svc = LinearSVC()
# Check the training time for the SVC
t=time.time()
svc.fit(X_train, y_train)
t2 = time.time()
print(round(t2-t, 2), 'Seconds to train SVC...')

# Check the score of the SVC
print('Test Accuracy of SVC = ', round(svc.score(X_test, y_test), 4))
# Check the prediction time for a single sample
t = time.time()

# Saving the model to a file

```

```

    with open('model.p', 'wb') as file:
        pickle.dump(svc, file)

Extracting features...
Extracting car features took 56.7898 seconds
Extracting noncar features took 59.3729 seconds
Using: 9 orientations 8 pixels per cell and 2 cells per block
Feature vector length: 5292
17.88 Seconds to train SVC...
Test Accuracy of SVC = 0.9809

```

```

In [105]: # Now get the boxes for all test images
use_default_number_of_processes = 1

for image in test_images:
    img = mpimg.imread(image)
    freeze_support()
    t0 = time.time()
    r = []
    r = parallelize_find_cars(img, 2)
    t1 = time.time()
    print('Time taken for processing each image', round(t1-t0, 6))

    img_with_boxes = np.copy(img)
    for box in r:
        cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
    plt.figure(figsize=(16,9))
    plt.imshow(img_with_boxes)

Length of task list: 5
Number of processes used: 3

Time taken for processing each image 1.510233
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 1.566912
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 1.563974
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 1.612893
Length of task list: 5
Number of processes used: 3

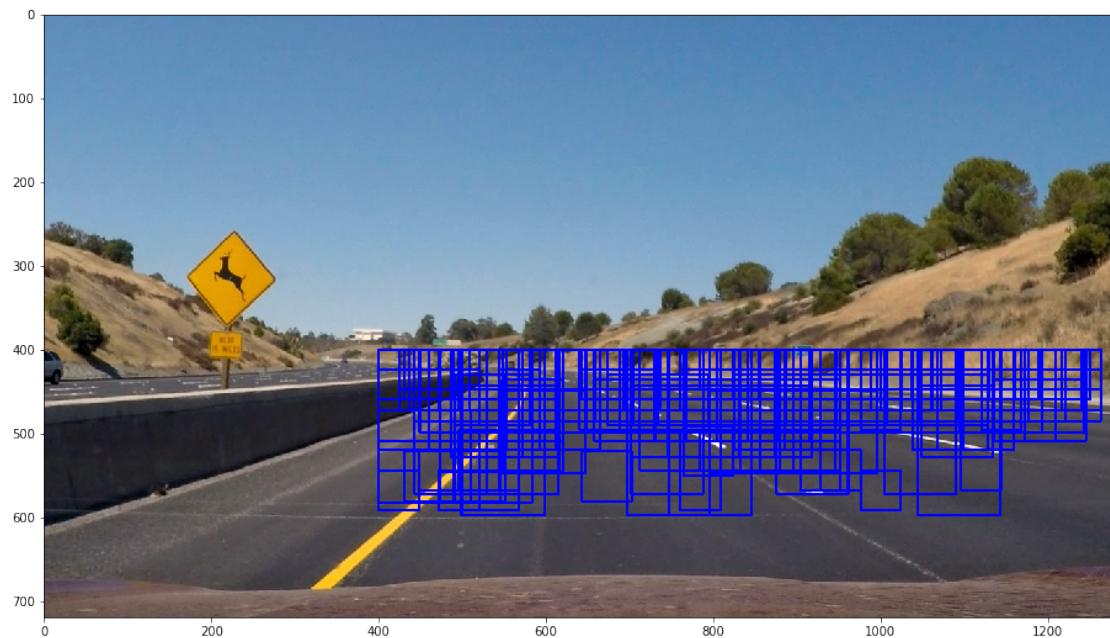
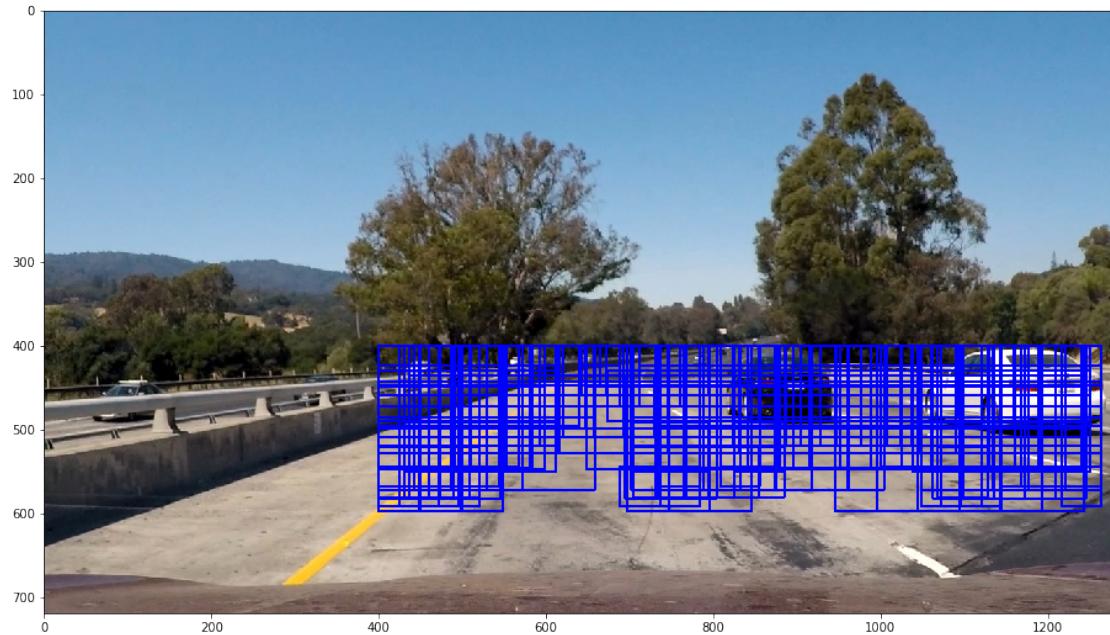
```

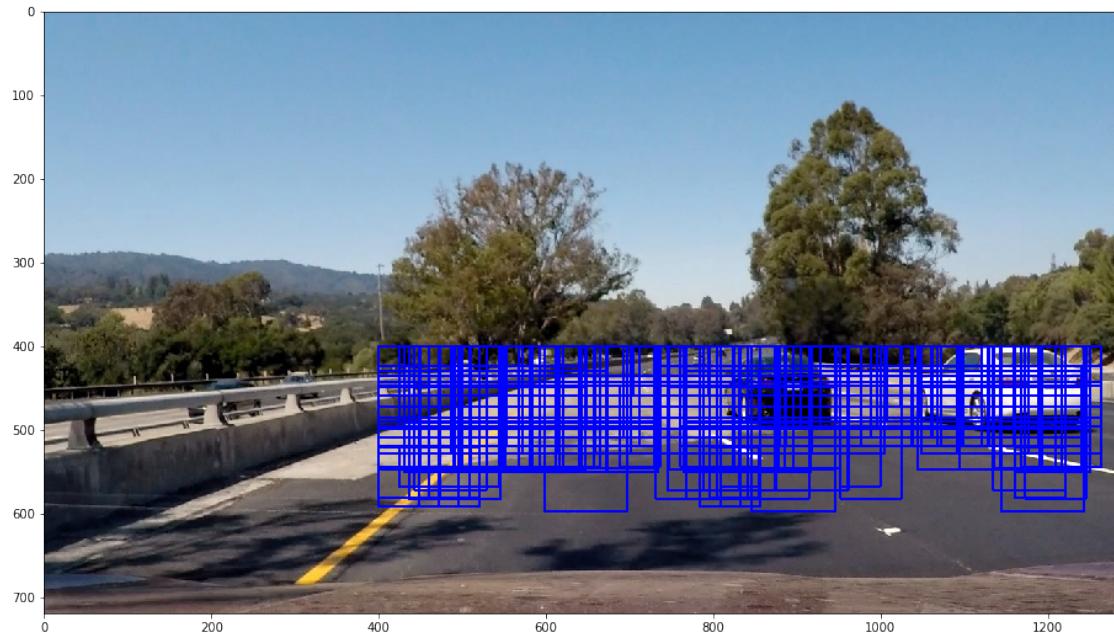
Time taken for processing each image 1.242857

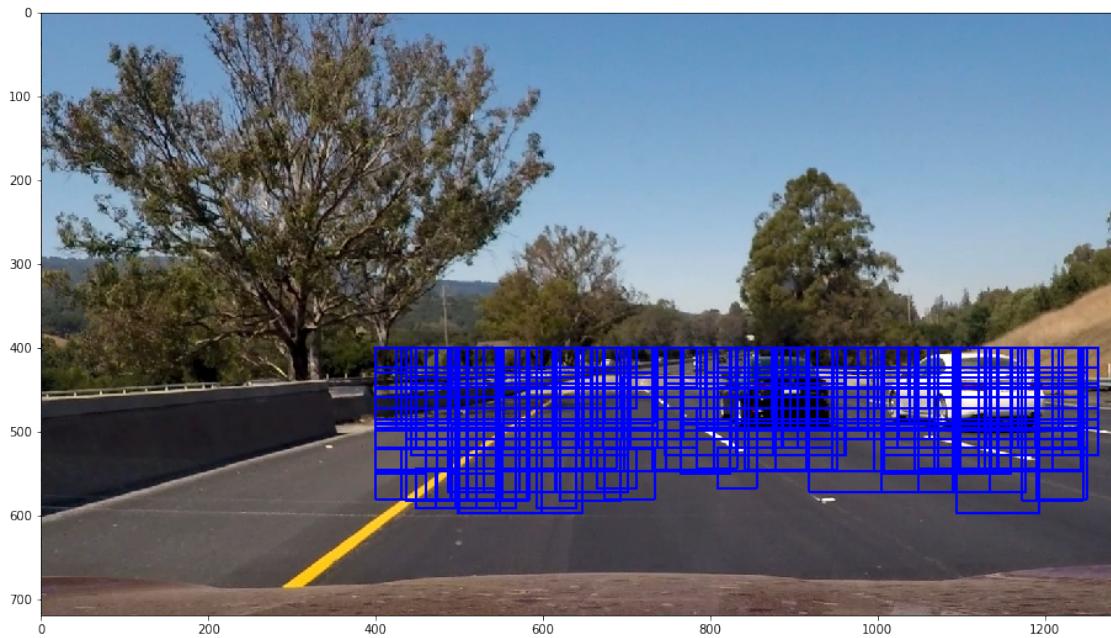
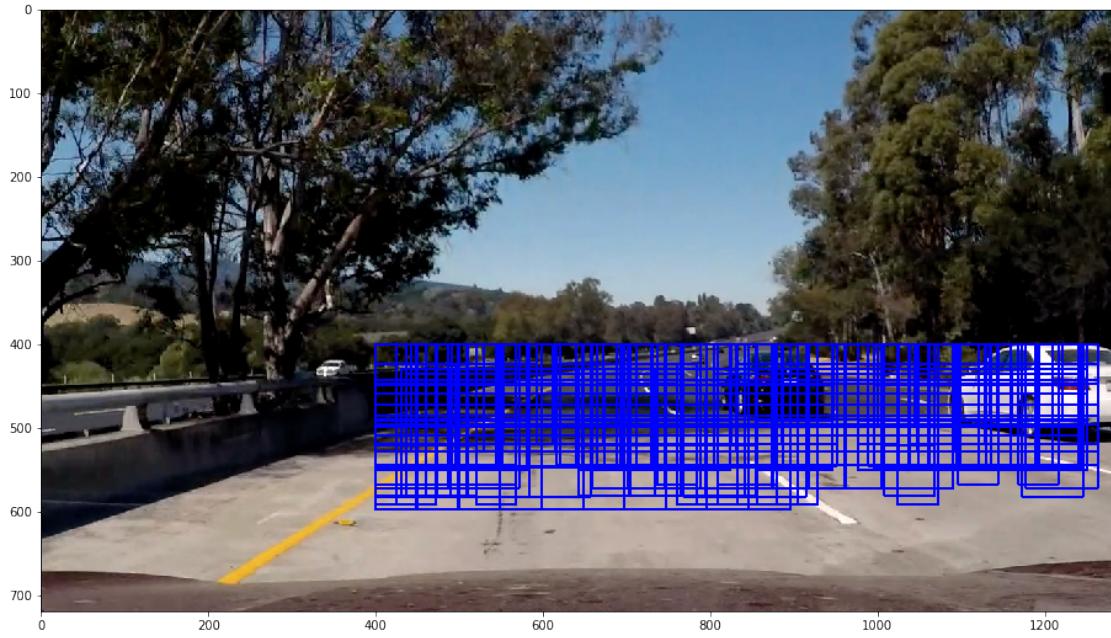
Length of task list: 5

Number of processes used: 3

Time taken for processing each image 1.636863







```
In [106]: ## Reduce the sample size because
# The quiz evaluator times out after 13s of CPU time
sample_size = 500
cars = car_images
notcars = noncar_images
```

```

### TODO: Tweak these parameters and see how the results change.
color_space = 'YCrCb' # Can be RGB, HSV, LUV, HLS, YUV, YCrCb
orient = 13 # HOG orientations
pix_per_cell = 8 # HOG pixels per cell
cell_per_block = 2 # HOG cells per block
hog_channel = 'ALL' # Can be 0, 1, 2, or "ALL"
spatial_size = (32, 32) # Spatial binning dimensions
hist_bins = 32 # Number of histogram bins
spatial_feat = False # Spatial features on or off
hist_feat = False
hog_feat = True # HOG features on or off
y_start_stop = [None, None] # Min and max in y to search in slide_window()

print("Extracting features...")

t0 = time.time()
car_features = extract_features(cars, color_space=color_space,
                                 spatial_size=spatial_size, hist_bins=hist_bins,
                                 orient=orient, pix_per_cell=pix_per_cell,
                                 cell_per_block=cell_per_block,
                                 hog_channel=hog_channel, spatial_feat=spatial_feat,
                                 hist_feat=hist_feat, hog_feat=hog_feat)
t1 = time.time()
print("Extracting car features took ", round(t1-t0, 4), " seconds")
notcar_features = extract_features(notcars, color_space=color_space,
                                    spatial_size=spatial_size, hist_bins=hist_bins,
                                    orient=orient, pix_per_cell=pix_per_cell,
                                    cell_per_block=cell_per_block,
                                    hog_channel=hog_channel, spatial_feat=spatial_feat,
                                    hist_feat=hist_feat, hog_feat=hog_feat)
t2 = time.time()
print("Extracting noncar features took ", round(t2-t1, 4), " seconds")

# Create an array stack of feature vectors
X = np.vstack((car_features, notcar_features)).astype(np.float64)

# Define the labels vector
y = np.hstack((np.ones(len(car_features)), np.zeros(len(notcar_features)))))

# Split up data into randomized training and test sets
rand_state = np.random.randint(0, 100)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=rand_state)

# Fit a per-column scaler
X_scaler = StandardScaler().fit(X_train)

```

```

# Apply the scaler to X
X_train = X_scaler.transform(X_train)
X_test = X_scaler.transform(X_test)

print('Using:',orient,'orientations',pix_per_cell,
      'pixels per cell and', cell_per_block,'cells per block')
print('Feature vector length:', len(X_train[0]))
# Use a linear SVC
svc = LinearSVC()
# Check the training time for the SVC
t=time.time()
svc.fit(X_train, y_train)
t2 = time.time()
print(round(t2-t, 2), 'Seconds to train SVC...')
# Check the score of the SVC
print('Test Accuracy of SVC = ', round(svc.score(X_test, y_test), 4))
# Check the prediction time for a single sample
t = time.time()

# Saving the model to a file

with open('model.p', 'wb') as file:
    pickle.dump(svc, file)

Extracting features...
Extracting car features took 58.5784 seconds
Extracting noncar features took 59.8088 seconds
Using: 13 orientations 8 pixels per cell and 2 cells per block
Feature vector length: 7644
7.65 Seconds to train SVC...
Test Accuracy of SVC = 0.984

```

```

In [107]: # Now get the boxes for all test images
use_default_number_of_processes = 1

for image in test_images:
    img = mpimg.imread(image)
    freeze_support()
    t0 = time.time()
    r = []
    r = parallelize_find_cars(img, 2)
    t1 = time.time()
    print('Time taken for processing each image', round(t1-t0, 6))

    img_with_boxes = np.copy(img)
    for box in r:
        cv2.rectangle(img_with_boxes, box[0],box[1],(0,0,255),2)

```

```
plt.figure(figsize=(16,9))
plt.imshow(img_with_boxes)

Length of task list: 5
Number of processes used: 3

Time taken for processing each image 1.52094
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 1.407931
Length of task list: 5
Number of processes used: 3

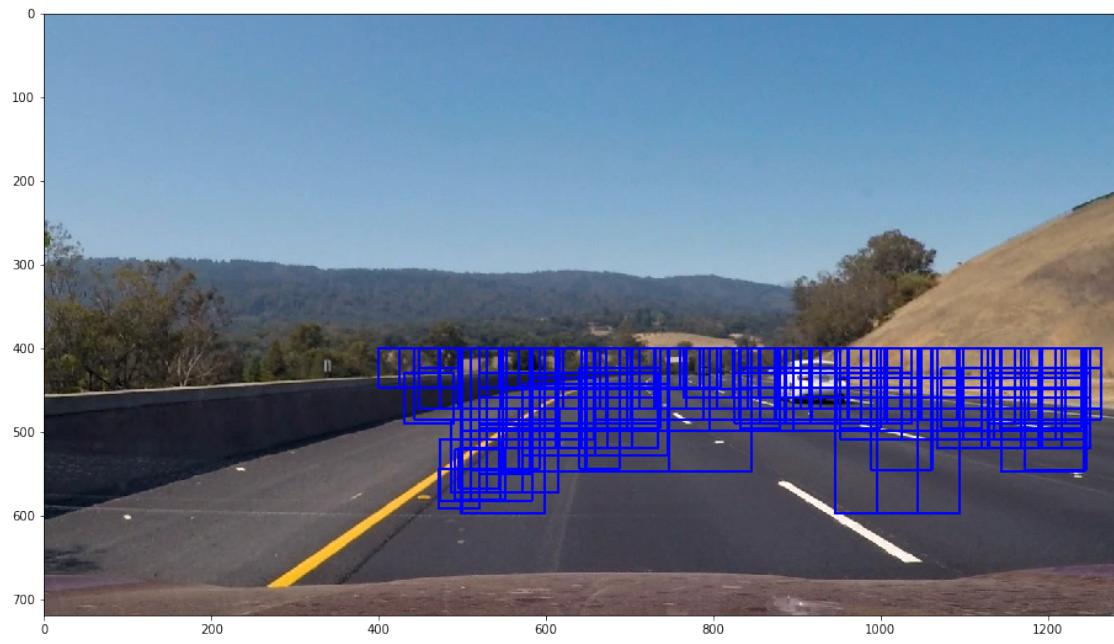
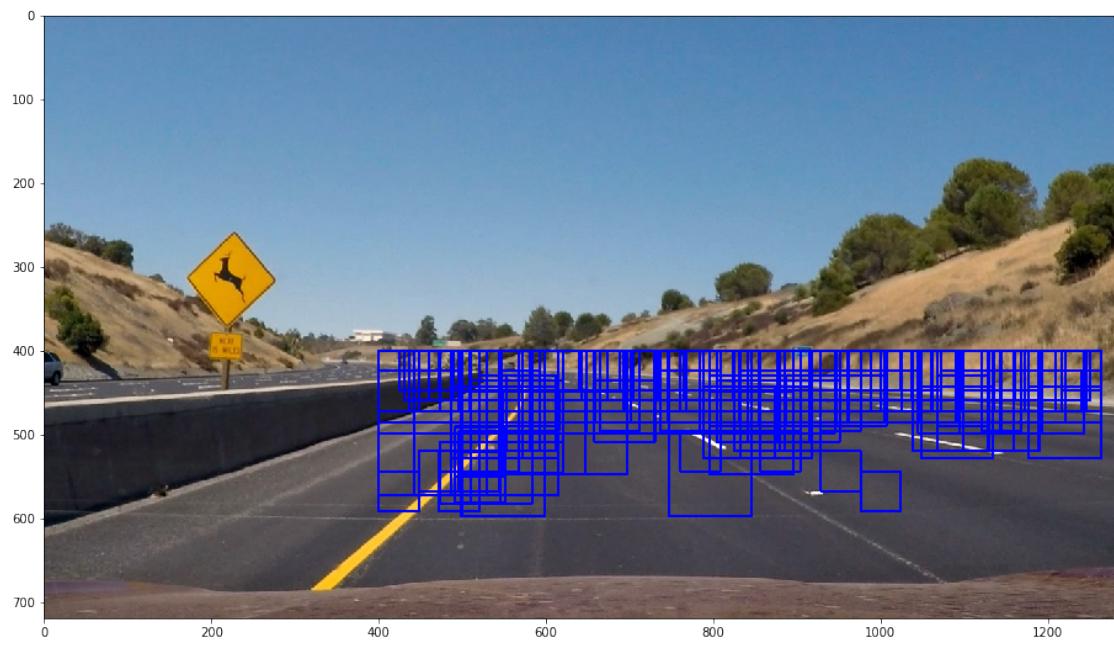
Time taken for processing each image 1.519255
Length of task list: 5
Number of processes used: 3

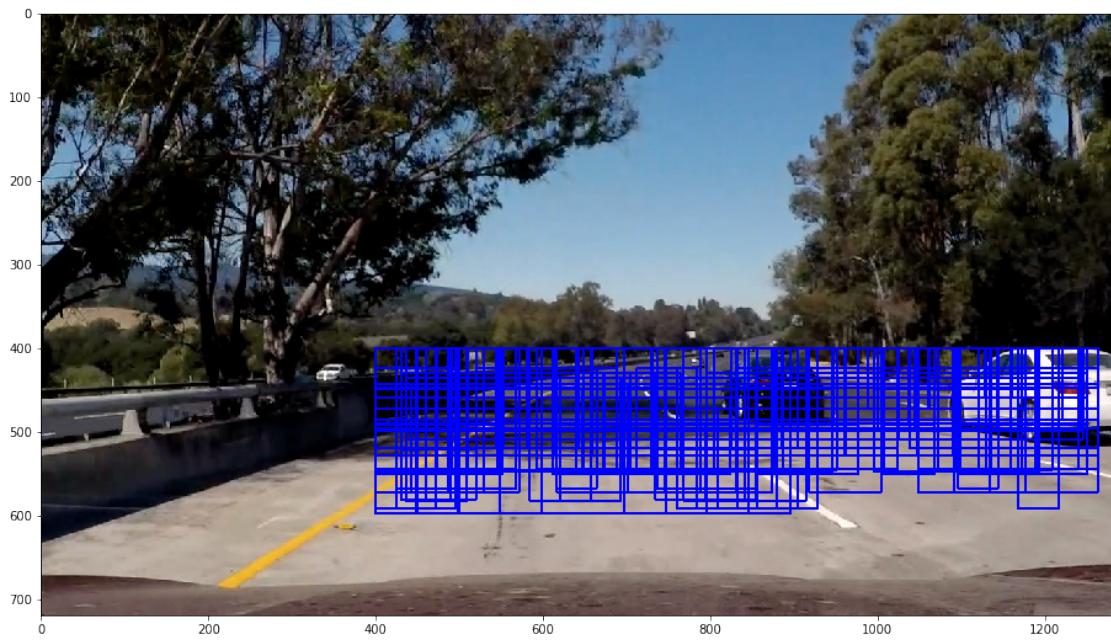
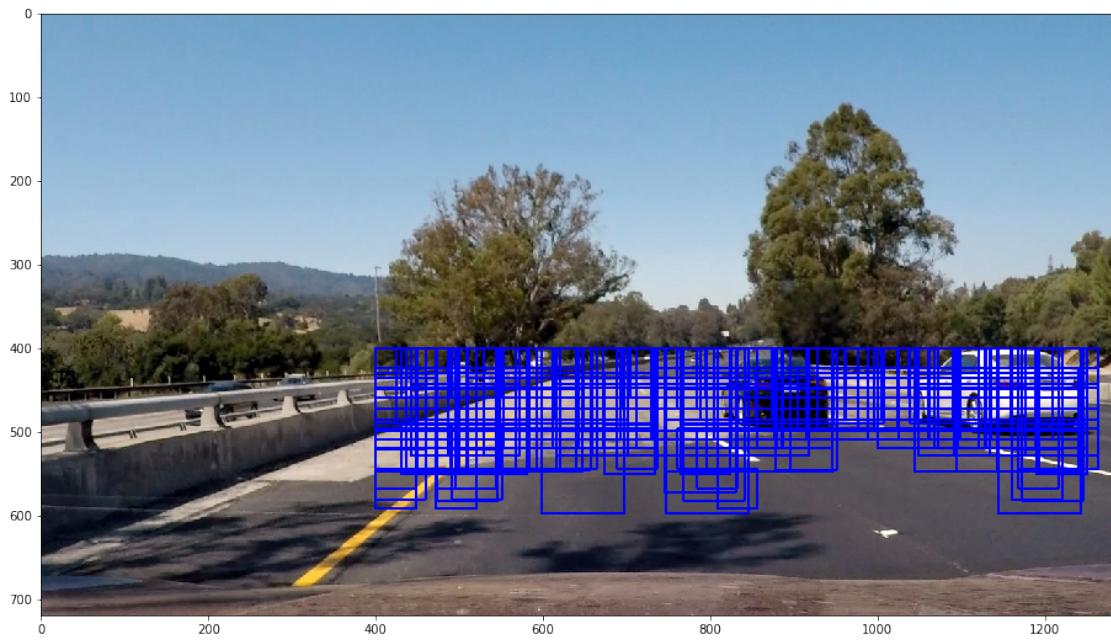
Time taken for processing each image 1.191918
Length of task list: 5
Number of processes used: 3

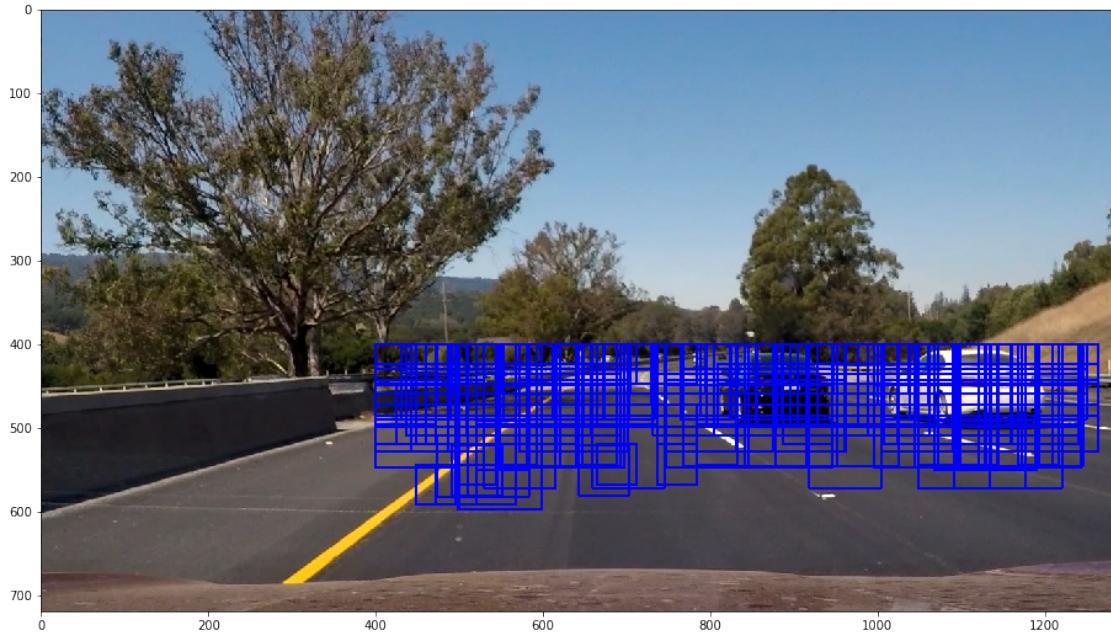
Time taken for processing each image 1.606431
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 1.434461
```









```
In [108]: ## Reduce the sample size because
# The quiz evaluator times out after 13s of CPU time
sample_size = 500
cars = car_images
notcars = noncar_images

### TODO: Tweak these parameters and see how the results change.
color_space = 'YCrCb' # Can be RGB, HSV, LUV, HLS, YUV, YCrCb
orient = 19 # HOG orientations
pix_per_cell = 6 # HOG pixels per cell
cell_per_block = 2 # HOG cells per block
hog_channel = 'ALL' # Can be 0, 1, 2, or "ALL"
spatial_size = (32, 32) # Spatial binning dimensions
hist_bins = 32 # Number of histogram bins
spatial_feat = False # Spatial features on or off
hist_feat = False
hog_feat = True # HOG features on or off
y_start_stop = [None, None] # Min and max in y to search in slide_window()

print("Extracting features...")

t0 = time.time()
car_features = extract_features(cars, color_space=color_space,
                                 spatial_size=spatial_size, hist_bins=hist_bins,
                                 orient=orient, pix_per_cell=pix_per_cell,
```

```

        cell_per_block=cell_per_block,
        hog_channel=hog_channel, spatial_feat=spatial_feat,
        hist_feat=hist_feat, hog_feat=hog_feat)

t1 = time.time()
print("Extracting car features took ", round(t1-t0, 4), " seconds")
notcar_features = extract_features(notcars, color_space=color_space,
                                    spatial_size=spatial_size, hist_bins=hist_bins,
                                    orient=orient, pix_per_cell=pix_per_cell,
                                    cell_per_block=cell_per_block,
                                    hog_channel=hog_channel, spatial_feat=spatial_feat,
                                    hist_feat=hist_feat, hog_feat=hog_feat)

t2 = time.time()
print("Extracting noncar features took ", round(t2-t1, 4), " seconds")

# Create an array stack of feature vectors
X = np.vstack((car_features, notcar_features)).astype(np.float64)

# Define the labels vector
y = np.hstack((np.ones(len(car_features)), np.zeros(len(notcar_features)))))

# Split up data into randomized training and test sets
rand_state = np.random.randint(0, 100)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=rand_state)

# Fit a per-column scaler
X_scaler = StandardScaler().fit(X_train)
# Apply the scaler to X
X_train = X_scaler.transform(X_train)
X_test = X_scaler.transform(X_test)

print('Using:', orient, 'orientations', pix_per_cell,
      'pixels per cell and', cell_per_block, 'cells per block')
print('Feature vector length:', len(X_train[0]))
# Use a linear SVC
svc = LinearSVC()
# Check the training time for the SVC
t=time.time()
svc.fit(X_train, y_train)
t2 = time.time()
print(round(t2-t, 2), 'Seconds to train SVC...')

# Check the score of the SVC
print('Test Accuracy of SVC = ', round(svc.score(X_test, y_test), 4))
# Check the prediction time for a single sample
t = time.time()

# Saving the model to a file

```

```
    with open('model.p', 'wb') as file:
        pickle.dump(svc, file)

Extracting features...
Extracting car features took 84.1716 seconds
Extracting noncar features took 84.4038 seconds
Using: 19 orientations 6 pixels per cell and 2 cells per block
Feature vector length: 18468
10.44 Seconds to train SVC...
Test Accuracy of SVC = 0.9851
```

```
In [109]: # Now get the boxes for all test images
use_default_number_of_processes = 1

for image in test_images:
    img = mpimg.imread(image)
    freeze_support()
    t0 = time.time()
    r = []
    r = parallelize_find_cars(img, 2)
    t1 = time.time()
    print('Time taken for processing each image', round(t1-t0, 6))

    img_with_boxes = np.copy(img)
    for box in r:
        cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
    plt.figure(figsize=(16,9))
    plt.imshow(img_with_boxes)

Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.008376
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 1.792213
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 1.749802
Length of task list: 5
Number of processes used: 3

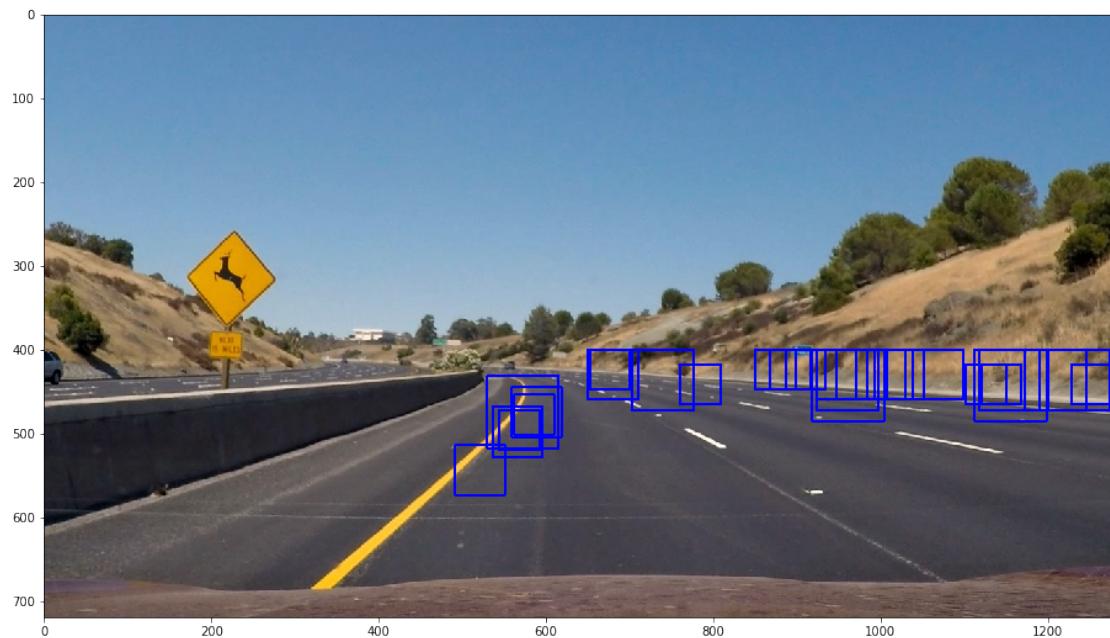
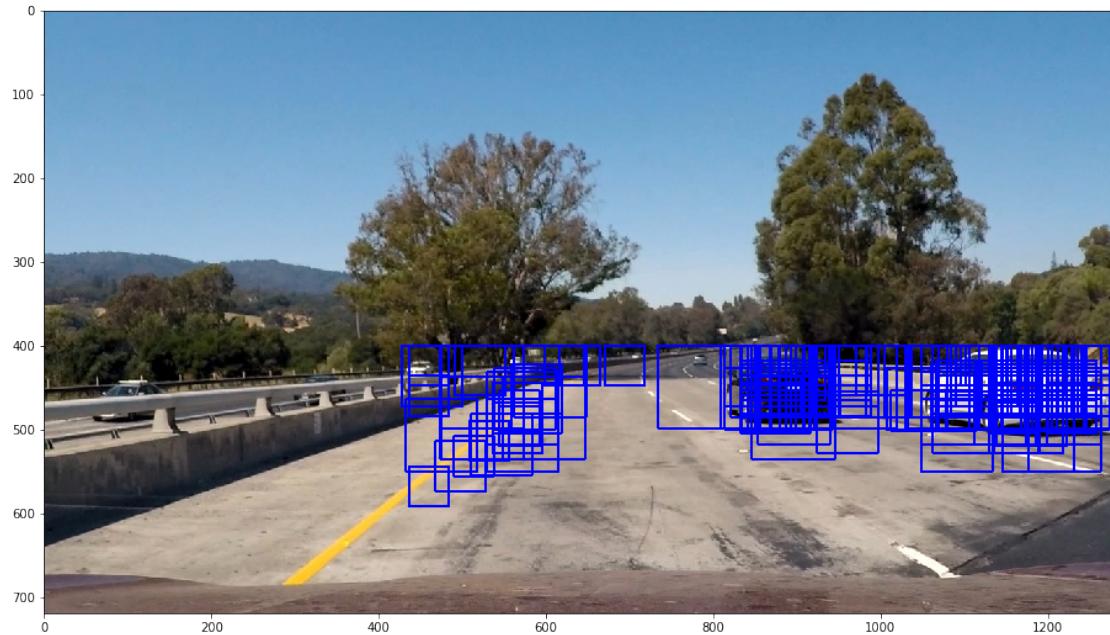
Time taken for processing each image 1.948521
Length of task list: 5
Number of processes used: 3
```

Time taken for processing each image 1.939917

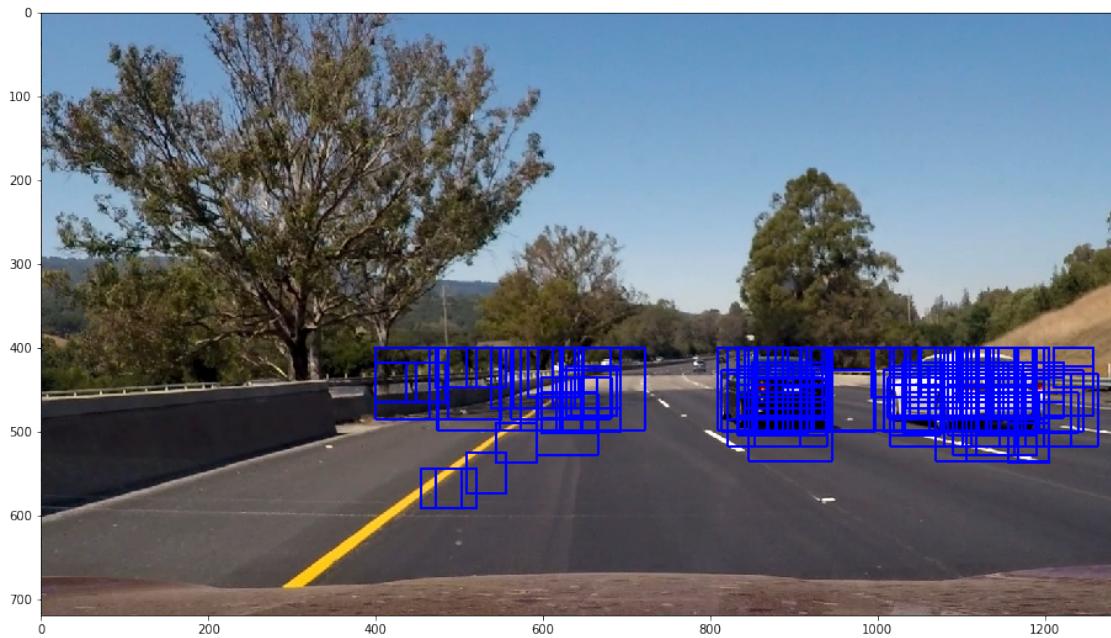
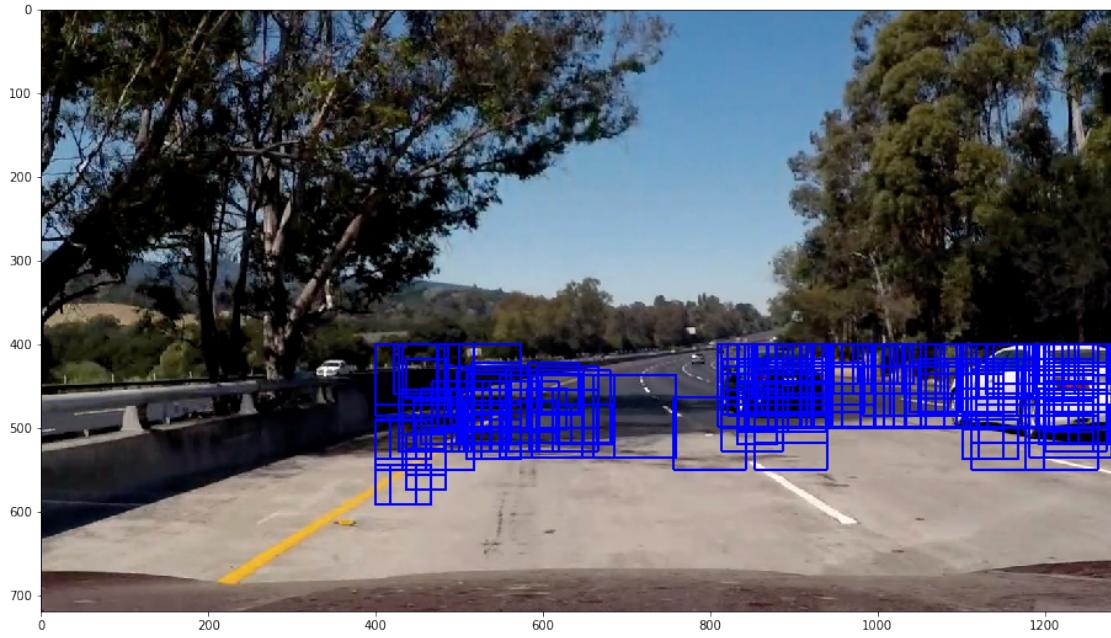
Length of task list: 5

Number of processes used: 3

Time taken for processing each image 2.178585







```
In [110]: ## Reduce the sample size because
# The quiz evaluator times out after 13s of CPU time
sample_size = 500
cars = car_images
notcars = noncar_images
```

```

### TODO: Tweak these parameters and see how the results change.
color_space = 'YCrCb' # Can be RGB, HSV, LUV, HLS, YUV, YCrCb
orient = 23 # HOG orientations
pix_per_cell = 6 # HOG pixels per cell
cell_per_block = 2 # HOG cells per block
hog_channel = 'ALL' # Can be 0, 1, 2, or "ALL"
spatial_size = (32, 32) # Spatial binning dimensions
hist_bins = 32 # Number of histogram bins
spatial_feat = False # Spatial features on or off
hist_feat = False
hog_feat = True # HOG features on or off
y_start_stop = [None, None] # Min and max in y to search in slide_window()

print("Extracting features...")

t0 = time.time()
car_features = extract_features(cars, color_space=color_space,
                                 spatial_size=spatial_size, hist_bins=hist_bins,
                                 orient=orient, pix_per_cell=pix_per_cell,
                                 cell_per_block=cell_per_block,
                                 hog_channel=hog_channel, spatial_feat=spatial_feat,
                                 hist_feat=hist_feat, hog_feat=hog_feat)
t1 = time.time()
print("Extracting car features took ", round(t1-t0, 4), " seconds")
notcar_features = extract_features(notcars, color_space=color_space,
                                    spatial_size=spatial_size, hist_bins=hist_bins,
                                    orient=orient, pix_per_cell=pix_per_cell,
                                    cell_per_block=cell_per_block,
                                    hog_channel=hog_channel, spatial_feat=spatial_feat,
                                    hist_feat=hist_feat, hog_feat=hog_feat)
t2 = time.time()
print("Extracting noncar features took ", round(t2-t1, 4), " seconds")

# Create an array stack of feature vectors
X = np.vstack((car_features, notcar_features)).astype(np.float64)

# Define the labels vector
y = np.hstack((np.ones(len(car_features)), np.zeros(len(notcar_features)))))

# Split up data into randomized training and test sets
rand_state = np.random.randint(0, 100)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=rand_state)

# Fit a per-column scaler
X_scaler = StandardScaler().fit(X_train)

```

```

# Apply the scaler to X
X_train = X_scaler.transform(X_train)
X_test = X_scaler.transform(X_test)

print('Using:',orient,'orientations',pix_per_cell,
      'pixels per cell and', cell_per_block, 'cells per block')
print('Feature vector length:', len(X_train[0]))
# Use a linear SVC
svc = LinearSVC()
# Check the training time for the SVC
t=time.time()
svc.fit(X_train, y_train)
t2 = time.time()
print(round(t2-t, 2), 'Seconds to train SVC...')
# Check the score of the SVC
print('Test Accuracy of SVC = ', round(svc.score(X_test, y_test), 4))
# Check the prediction time for a single sample
t = time.time()

# Saving the model to a file

with open('model.p', 'wb') as file:
    pickle.dump(svc, file)

```

Extracting features...
Extracting car features took 84.9188 seconds
Extracting noncar features took 85.2282 seconds
Using: 23 orientations 6 pixels per cell and 2 cells per block
Feature vector length: 22356
12.55 Seconds to train SVC...
Test Accuracy of SVC = 0.9823

```

In [111]: # Now get the boxes for all test images
use_default_number_of_processes = 1

for image in test_images:
    img = mpimg.imread(image)
    freeze_support()
    t0 = time.time()
    r = []
    r = parallelize_find_cars(img, 2)
    t1 = time.time()
    print('Time taken for processing each image', round(t1-t0, 6))

    img_with_boxes = np.copy(img)
    for box in r:
        cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)

```

```
plt.figure(figsize=(16,9))
plt.imshow(img_with_boxes)

Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.144241
Length of task list: 5
Number of processes used: 3

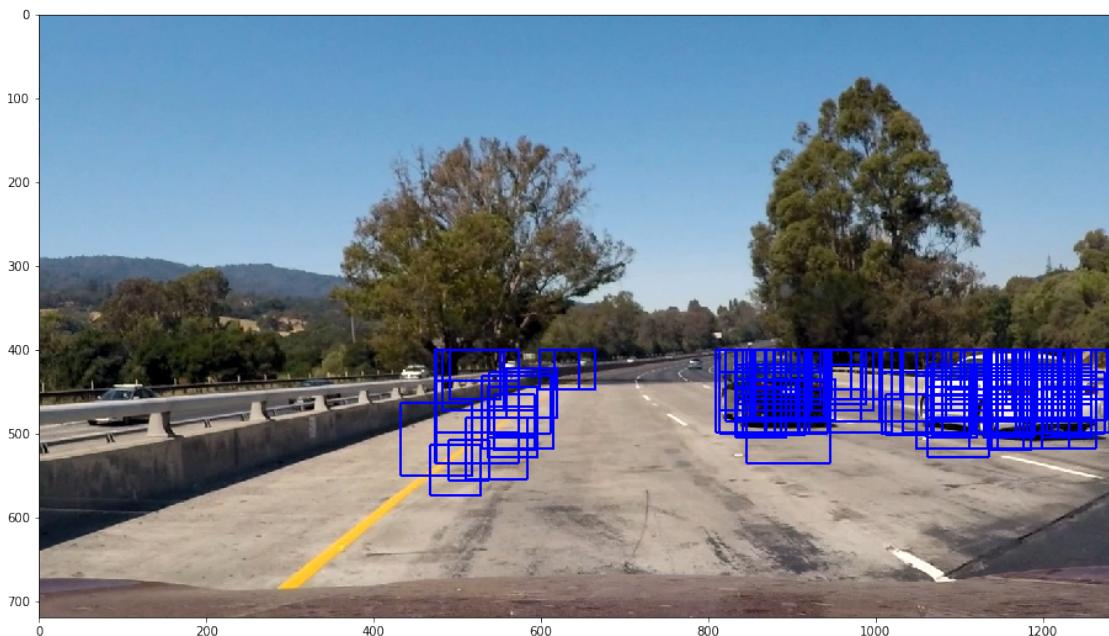
Time taken for processing each image 1.769417
Length of task list: 5
Number of processes used: 3

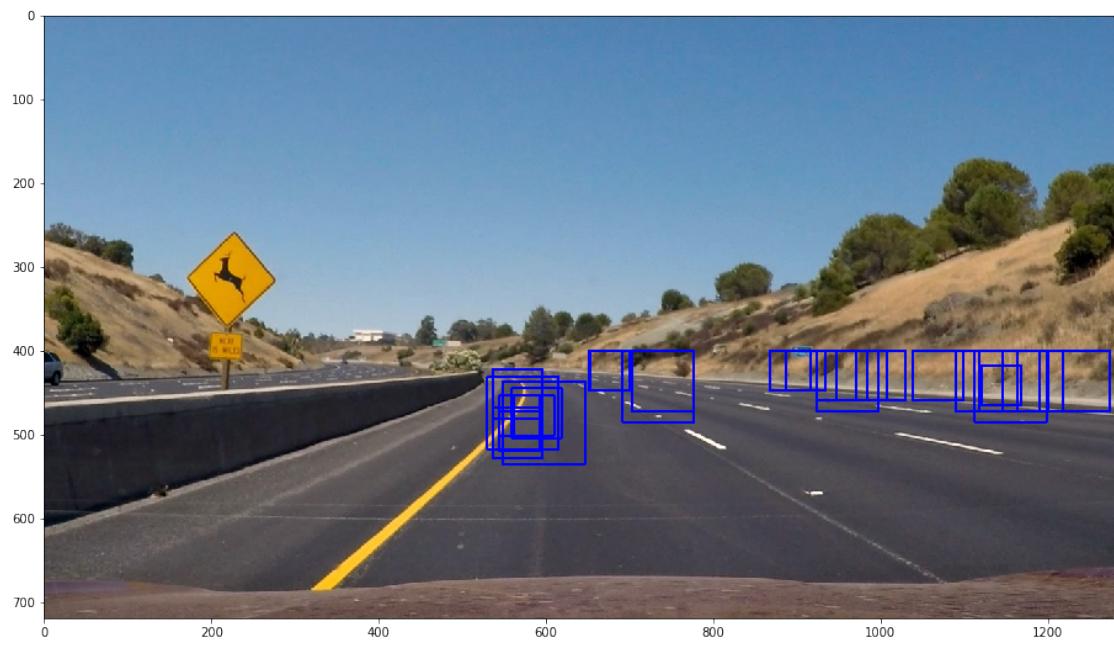
Time taken for processing each image 1.99668
Length of task list: 5
Number of processes used: 3

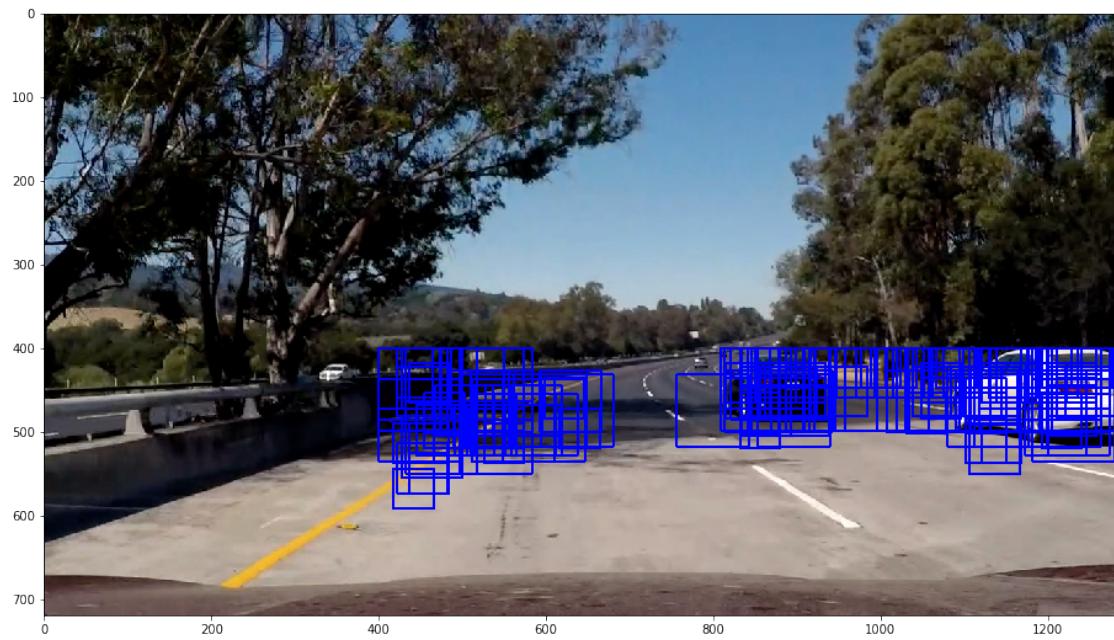
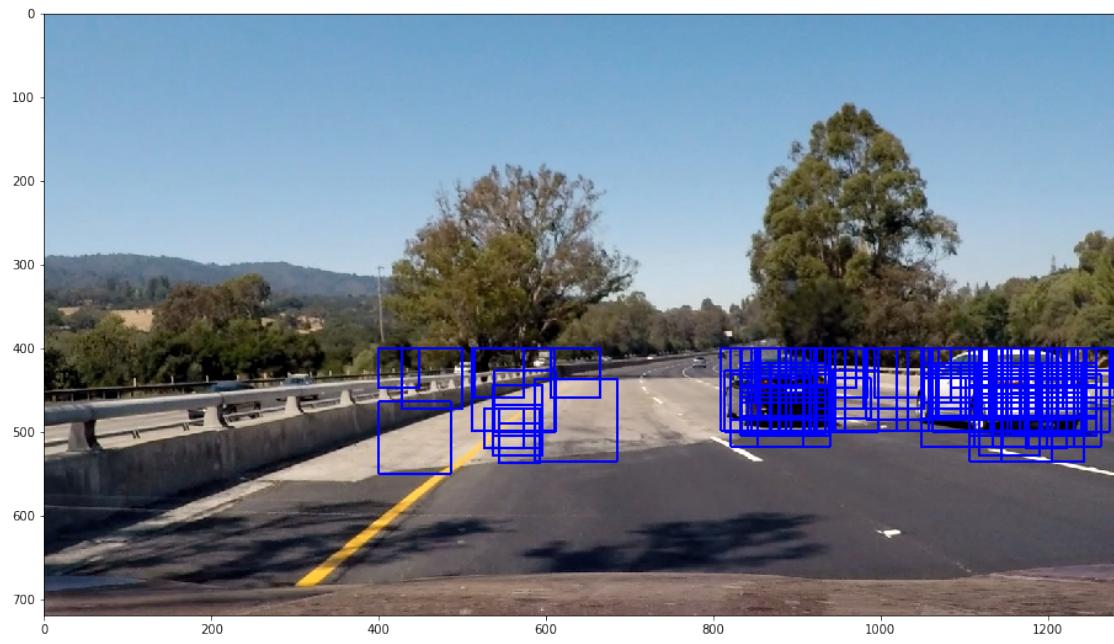
Time taken for processing each image 2.179799
Length of task list: 5
Number of processes used: 3

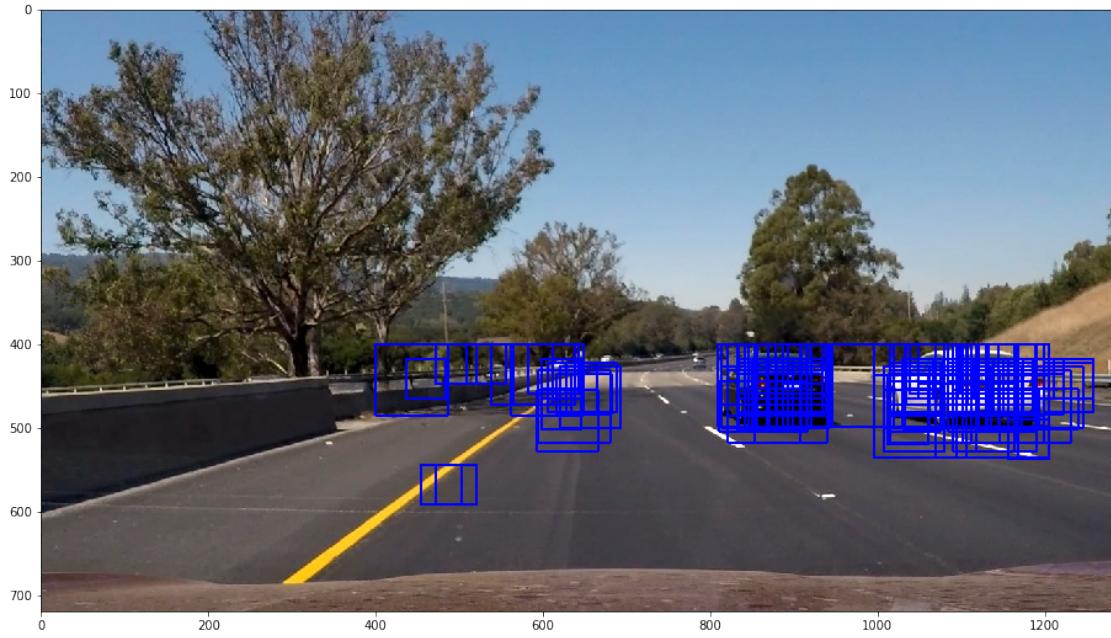
Time taken for processing each image 1.949977
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.181973
```









```
In [112]: ## Reduce the sample size because
# The quiz evaluator times out after 13s of CPU time
sample_size = 500
cars = car_images
notcars = noncar_images

### TODO: Tweak these parameters and see how the results change.
color_space = 'YCrCb' # Can be RGB, HSV, LUV, HLS, YUV, YCrCb
orient = 31 # HOG orientations
pix_per_cell = 5 # HOG pixels per cell
cell_per_block = 2 # HOG cells per block
hog_channel = 'ALL' # Can be 0, 1, 2, or "ALL"
spatial_size = (32, 32) # Spatial binning dimensions
hist_bins = 32 # Number of histogram bins
spatial_feat = False # Spatial features on or off
hist_feat = False
hog_feat = True # HOG features on or off
y_start_stop = [None, None] # Min and max in y to search in slide_window()

print("Extracting features...")

t0 = time.time()
car_features = extract_features(cars, color_space=color_space,
                                 spatial_size=spatial_size, hist_bins=hist_bins,
                                 orient=orient, pix_per_cell=pix_per_cell,
```

```

        cell_per_block=cell_per_block,
        hog_channel=hog_channel, spatial_feat=spatial_feat,
        hist_feat=hist_feat, hog_feat=hog_feat)

t1 = time.time()
print("Extracting car features took ", round(t1-t0, 4), " seconds")
notcar_features = extract_features(notcars, color_space=color_space,
                                    spatial_size=spatial_size, hist_bins=hist_bins,
                                    orient=orient, pix_per_cell=pix_per_cell,
                                    cell_per_block=cell_per_block,
                                    hog_channel=hog_channel, spatial_feat=spatial_feat,
                                    hist_feat=hist_feat, hog_feat=hog_feat)

t2 = time.time()
print("Extracting noncar features took ", round(t2-t1, 4), " seconds")

# Create an array stack of feature vectors
X = np.vstack((car_features, notcar_features)).astype(np.float64)

# Define the labels vector
y = np.hstack((np.ones(len(car_features)), np.zeros(len(notcar_features)))))

# Split up data into randomized training and test sets
rand_state = np.random.randint(0, 100)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=rand_state)

# Fit a per-column scaler
X_scaler = StandardScaler().fit(X_train)
# Apply the scaler to X
X_train = X_scaler.transform(X_train)
X_test = X_scaler.transform(X_test)

print('Using:', orient, 'orientations', pix_per_cell,
      'pixels per cell and', cell_per_block, 'cells per block')
print('Feature vector length:', len(X_train[0]))
# Use a linear SVC
svc = LinearSVC()
# Check the training time for the SVC
t=time.time()
svc.fit(X_train, y_train)
t2 = time.time()
print(round(t2-t, 2), 'Seconds to train SVC...')

# Check the score of the SVC
print('Test Accuracy of SVC = ', round(svc.score(X_test, y_test), 4))
# Check the prediction time for a single sample
t = time.time()

# Saving the model to a file

```

```
    with open('model.p', 'wb') as file:
        pickle.dump(svc, file)

Extracting features...
Extracting car features took 113.4719 seconds
Extracting noncar features took 114.573 seconds
Using: 31 orientations 5 pixels per cell and 2 cells per block
Feature vector length: 45012
26.52 Seconds to train SVC...
Test Accuracy of SVC = 0.9837
```

```
In [113]: # Now get the boxes for all test images
use_default_number_of_processes = 1

for image in test_images:
    img = mpimg.imread(image)
    freeze_support()
    t0 = time.time()
    r = []
    r = parallelize_find_cars(img, 2)
    t1 = time.time()
    print('Time taken for processing each image', round(t1-t0, 6))

    img_with_boxes = np.copy(img)
    for box in r:
        cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
    plt.figure(figsize=(16,9))
    plt.imshow(img_with_boxes)

Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.576974
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.487775
Length of task list: 5
Number of processes used: 3

Time taken for processing each image 2.689806
Length of task list: 5
Number of processes used: 3

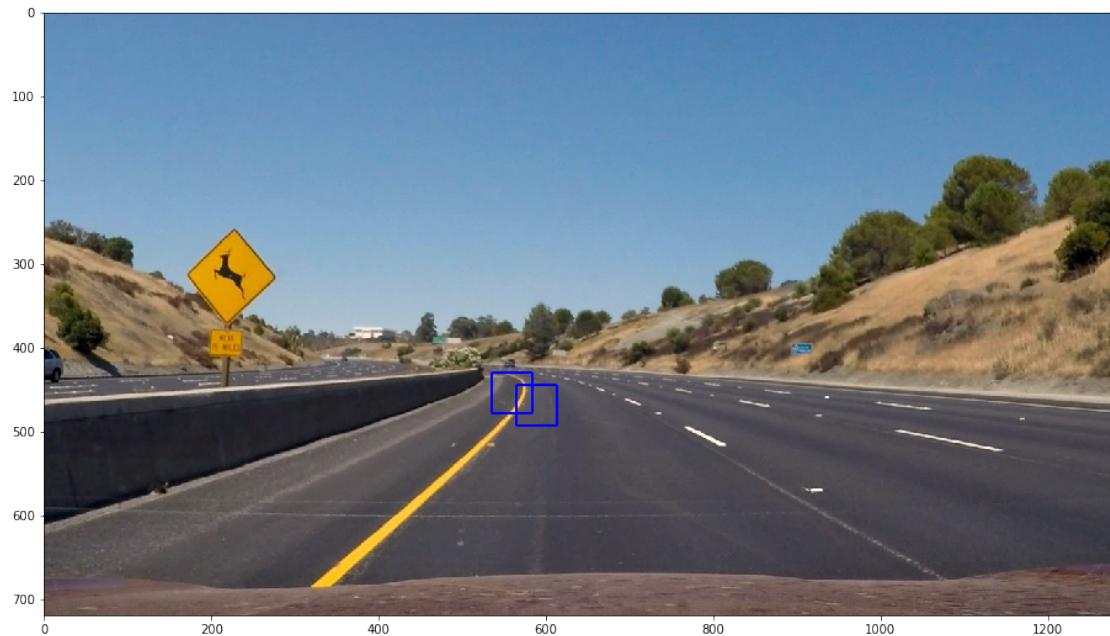
Time taken for processing each image 2.271713
Length of task list: 5
Number of processes used: 3
```

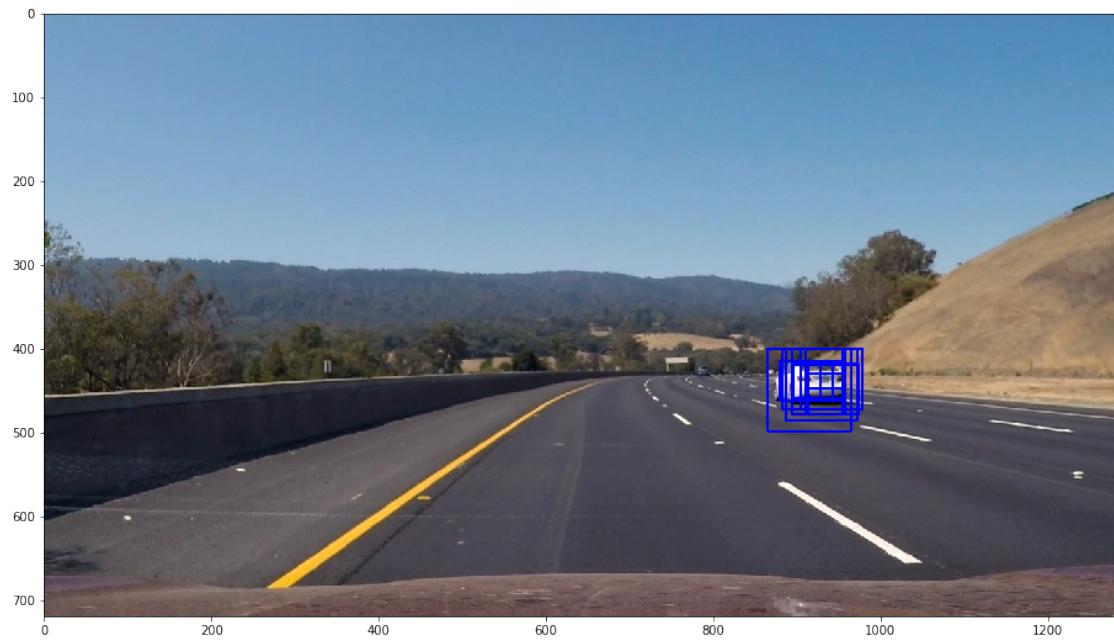
Time taken for processing each image 2.59244

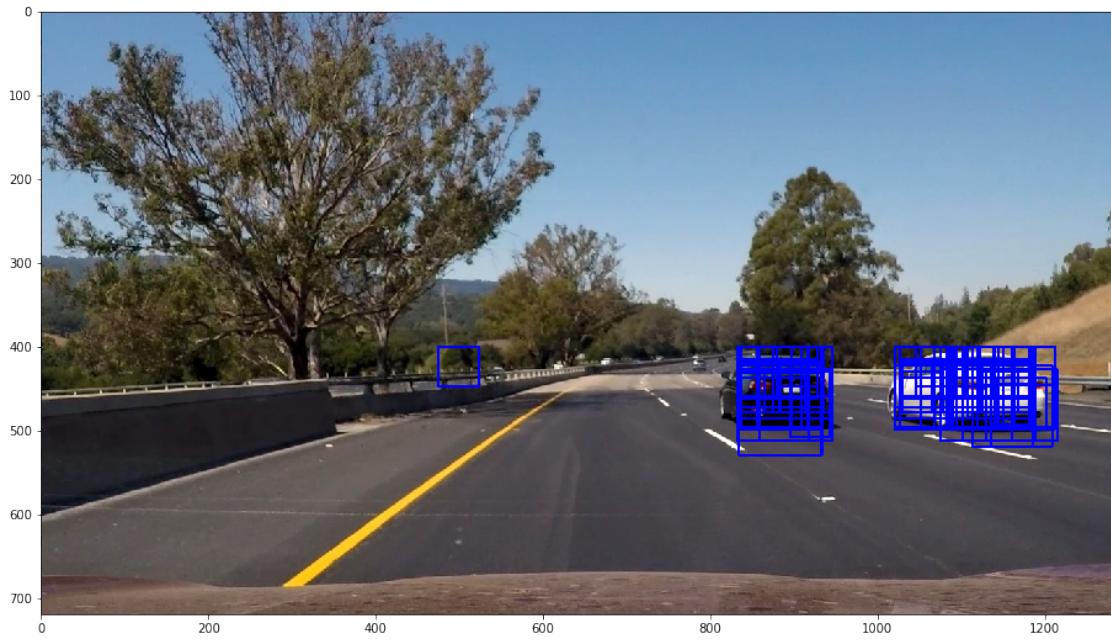
Length of task list: 5

Number of processes used: 3

Time taken for processing each image 2.639971







1.12 Checking the performance based on different window sizes and multiprocessing on all test images

```
In [114]: # Testing the performance with multi processing with various number of processes per s  
# (range 1 to number of windowsizes)
```

```

use_default_number_of_processes = 0

time_elapsed = 0
for i in range(7):
    for j in range(1, len(scaleslist[i])+1):
        time_for_all_images = []
        test_images_with_boxes = []
        number_of_processes_used = j
        for image in test_images:
            img = mpimg.imread(image)
            freeze_support()
            t0 = time.time()
            r = []
            r = parallelize_find_cars(img, number_of_processes_used, i)
            t1 = time.time()
            processing_time = round(t1-t0, 4)
            #print('Time taken for processing each image', processing_time)
            time_for_all_images.append(processing_time)
            img_with_boxes = np.copy(img)
            for box in r:
                cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
            #plt.figure(figsize=(16,9))
            #plt.imshow(img_with_boxes)
            test_images_with_boxes.append(img_with_boxes)

        print('Window sizes used:', scaleslist[i])
        print(number_of_processes_used, 'processes used for testing', len(scaleslist[i]))
        print('Processing times for each image', time_for_all_images, \
              'with an average of ', round(sum(time_for_all_images)/len(time_for_all_i
        print()
        time_elapsed += sum(time_for_all_images)
        print('Time elapsed so far...', time_elapsed)
        print()
        print()

```

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

Window sizes used: [0.95]
1 processes used for testing 1 window sizes
Processing times for each image [1.5304, 1.3848, 1.3727, 1.8957, 1.6475, 1.6049] with an average

Time elapsed so far... 9.436

Length of task list: 3
Number of processes used: 1

Length of task list: 3
Number of processes used: 1

Length of task list: 3
Number of processes used: 1

Length of task list: 3
Number of processes used: 1

Length of task list: 3
Number of processes used: 1

Length of task list: 3
Number of processes used: 1

Window sizes used: [1.15, 1.35, 1.55]
1 processes used for testing 3 window sizes
Processing times for each image [2.5064, 2.2022, 2.6602, 2.9924, 2.5625, 2.2393] with an average

Time elapsed so far... 24.599

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

Window sizes used: [1.15, 1.35, 1.55]
2 processes used for testing 3 window sizes
Processing times for each image [1.9187, 1.5929, 2.1444, 2.2488, 1.6508, 2.2322] with an average

Time elapsed so far... 36.3868

Length of task list: 3
Number of processes used: 3

Length of task list: 3
Number of processes used: 3

Length of task list: 3
Number of processes used: 3

Length of task list: 3
Number of processes used: 3

Length of task list: 3
Number of processes used: 3

Length of task list: 3
Number of processes used: 3

Window sizes used: [1.15, 1.35, 1.55]
3 processes used for testing 3 window sizes
Processing times for each image [1.5997, 1.7594, 1.4861, 2.1041, 1.6892, 1.7554] with an average

Time elapsed so far... 46.7807

Length of task list: 5
Number of processes used: 1

Length of task list: 5
Number of processes used: 1

Length of task list: 5
Number of processes used: 1

Length of task list: 5

Number of processes used: 1

Length of task list: 5

Number of processes used: 1

Length of task list: 5

Number of processes used: 1

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

1 processes used for testing 5 window sizes

Processing times for each image [4.8465, 5.9765, 5.0433, 4.7491, 5.1001, 5.2437] with an average

Time elapsed so far... 77.7399

Length of task list: 5

Number of processes used: 2

Length of task list: 5

Number of processes used: 2

Length of task list: 5

Number of processes used: 2

Length of task list: 5

Number of processes used: 2

Length of task list: 5

Number of processes used: 2

Length of task list: 5

Number of processes used: 2

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

2 processes used for testing 5 window sizes

Processing times for each image [3.4958, 3.2555, 3.3954, 3.1358, 3.4752, 3.1667] with an average

Time elapsed so far... 97.6643

Length of task list: 5

Number of processes used: 3

Length of task list: 5

Number of processes used: 3

Length of task list: 5

Number of processes used: 3

Length of task list: 5

Number of processes used: 3

Length of task list: 5

Number of processes used: 3

Length of task list: 5

Number of processes used: 3

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

3 processes used for testing 5 window sizes

Processing times for each image [2.4718, 2.5422, 2.9204, 2.5312, 2.9938, 2.6311] with an average

Time elapsed so far... 113.75479999999999

Length of task list: 5

Number of processes used: 4

Length of task list: 5

Number of processes used: 4

Length of task list: 5

Number of processes used: 4

Length of task list: 5

Number of processes used: 4

Length of task list: 5

Number of processes used: 4

Length of task list: 5

Number of processes used: 4

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

4 processes used for testing 5 window sizes

Processing times for each image [2.523, 2.4833, 2.5155, 2.2847, 2.3944, 2.7811] with an average

Time elapsed so far... 128.7368

Length of task list: 5

Number of processes used: 5

Length of task list: 5

Number of processes used: 5

Length of task list: 5

Number of processes used: 5

Length of task list: 5

Number of processes used: 5

Length of task list: 5

Number of processes used: 5

Length of task list: 5

Number of processes used: 5

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

5 processes used for testing 5 window sizes

Processing times for each image [2.3711, 2.3689, 2.4911, 2.4108, 2.4218, 2.4803] with an average

Time elapsed so far... 143.2808

Length of task list: 5

Number of processes used: 1

Length of task list: 5

Number of processes used: 1

Length of task list: 5

Number of processes used: 1

Length of task list: 5

Number of processes used: 1

Length of task list: 5

Number of processes used: 1

Length of task list: 5

Number of processes used: 1

Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]

1 processes used for testing 5 window sizes

Processing times for each image [3.1269, 4.4733, 3.9132, 3.3639, 3.7161, 3.3964] with an average

Time elapsed so far... 165.2706

Length of task list: 5

Number of processes used: 2

Length of task list: 5

Number of processes used: 2

Length of task list: 5

Number of processes used: 2

Length of task list: 5
Number of processes used: 2

Length of task list: 5
Number of processes used: 2

Length of task list: 5
Number of processes used: 2

Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]

2 processes used for testing 5 window sizes

Processing times for each image [2.367, 2.2101, 2.1381, 2.6034, 2.4787, 2.6863] with an average

Time elapsed so far... 179.7542

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]

3 processes used for testing 5 window sizes

Processing times for each image [2.3005, 2.0406, 1.9134, 1.9504, 1.8841, 2.2183] with an average

Time elapsed so far... 192.0615

Length of task list: 5
Number of processes used: 4

Length of task list: 5
Number of processes used: 4

Length of task list: 5
Number of processes used: 4

Length of task list: 5
Number of processes used: 4

Length of task list: 5
Number of processes used: 4

Length of task list: 5
Number of processes used: 4

Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]

4 processes used for testing 5 window sizes

Processing times for each image [2.0284, 2.041, 2.1592, 2.2314, 2.1378, 2.01] with an average of

Time elapsed so far... 204.6693

Length of task list: 5
Number of processes used: 5

Length of task list: 5
Number of processes used: 5

Length of task list: 5
Number of processes used: 5

Length of task list: 5
Number of processes used: 5

Length of task list: 5
Number of processes used: 5

Length of task list: 5
Number of processes used: 5

Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]

5 processes used for testing 5 window sizes

Processing times for each image [2.2968, 2.3415, 2.4255, 2.3822, 2.2617, 2.1236] with an average of

Time elapsed so far... 218.5006

Length of task list: 6
Number of processes used: 1

Length of task list: 6
Number of processes used: 1

Length of task list: 6
Number of processes used: 1

Length of task list: 6
Number of processes used: 1

Length of task list: 6
Number of processes used: 1

Length of task list: 6
Number of processes used: 1

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05]
1 processes used for testing 6 window sizes
Processing times for each image [7.7084, 8.8874, 8.536, 8.3057, 7.9452, 8.1885] with an average

Time elapsed so far... 268.0718

Length of task list: 6
Number of processes used: 2

Length of task list: 6
Number of processes used: 2

Length of task list: 6
Number of processes used: 2

Length of task list: 6
Number of processes used: 2

Length of task list: 6
Number of processes used: 2

Length of task list: 6
Number of processes used: 2

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05]
2 processes used for testing 6 window sizes
Processing times for each image [4.524, 4.3918, 4.4491, 4.7214, 4.434, 4.8758] with an average

Time elapsed so far... 295.4679

Length of task list: 6
Number of processes used: 3

Length of task list: 6
Number of processes used: 3

Length of task list: 6
Number of processes used: 3

Length of task list: 6
Number of processes used: 3

Length of task list: 6
Number of processes used: 3

Length of task list: 6
Number of processes used: 3

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05]

3 processes used for testing 6 window sizes

Processing times for each image [3.6298, 3.6894, 3.5192, 3.6158, 3.681, 3.5072] with an average

Time elapsed so far... 317.1103

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05]

4 processes used for testing 6 window sizes

Processing times for each image [3.1613, 3.5603, 3.4305, 3.2023, 3.5025, 3.2226] with an average

Time elapsed so far... 337.1898

Length of task list: 6
Number of processes used: 5

Length of task list: 6
Number of processes used: 5

Length of task list: 6
Number of processes used: 5

Length of task list: 6
Number of processes used: 5

Length of task list: 6
Number of processes used: 5

Length of task list: 6
Number of processes used: 5

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05]

5 processes used for testing 6 window sizes

Processing times for each image [3.1717, 3.073, 2.9494, 3.2601, 3.1015, 3.5127] with an average

Time elapsed so far... 356.2582

Length of task list: 6
Number of processes used: 6

Length of task list: 6
Number of processes used: 6

Length of task list: 6
Number of processes used: 6

Length of task list: 6
Number of processes used: 6

Length of task list: 6
Number of processes used: 6

Length of task list: 6
Number of processes used: 6

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05]

6 processes used for testing 6 window sizes

Processing times for each image [3.2406, 3.3053, 3.3092, 3.3206, 3.4013, 3.0965] with an average

Time elapsed so far... 375.9317

Length of task list: 9
Number of processes used: 1

Length of task list: 9
Number of processes used: 1

Length of task list: 9

Number of processes used: 1

Length of task list: 9

Number of processes used: 1

Length of task list: 9

Number of processes used: 1

Length of task list: 9

Number of processes used: 1

Window sizes used: [0.75, 0.85, 0.95, 1.05, 1.15, 1.25, 1.35, 1.45, 1.55]

1 processes used for testing 9 window sizes

Processing times for each image [8.4853, 7.7933, 8.9508, 7.7084, 8.8171, 8.3201] with an average

Time elapsed so far... 426.00669999999997

Length of task list: 9

Number of processes used: 2

Length of task list: 9

Number of processes used: 2

Length of task list: 9

Number of processes used: 2

Length of task list: 9

Number of processes used: 2

Length of task list: 9

Number of processes used: 2

Length of task list: 9

Number of processes used: 2

Window sizes used: [0.75, 0.85, 0.95, 1.05, 1.15, 1.25, 1.35, 1.45, 1.55]

2 processes used for testing 9 window sizes

Processing times for each image [4.8447, 4.8429, 4.8516, 4.7801, 5.2987, 5.0871] with an average

Time elapsed so far... 455.7118

Length of task list: 9

Number of processes used: 3

Length of task list: 9

Number of processes used: 3

Length of task list: 9

Number of processes used: 3

Length of task list: 9

Number of processes used: 3

Length of task list: 9

Number of processes used: 3

Length of task list: 9

Number of processes used: 3

Window sizes used: [0.75, 0.85, 0.95, 1.05, 1.15, 1.25, 1.35, 1.45, 1.55]

3 processes used for testing 9 window sizes

Processing times for each image [3.5698, 3.9802, 3.7921, 3.8548, 3.8044, 4.0901] with an average

Time elapsed so far... 478.8032

Length of task list: 9

Number of processes used: 4

Length of task list: 9

Number of processes used: 4

Length of task list: 9

Number of processes used: 4

Length of task list: 9

Number of processes used: 4

Length of task list: 9

Number of processes used: 4

Length of task list: 9

Number of processes used: 4

Time elapsed so far... 500.0816

Length of task list: 9

Number of processes used: 5

Length of task list: 9

Number of processes used: 5

Length of task list: 9

Number of processes used: 7

Length of task list: 9

Number of processes used: 7

Length of task list: 9

Number of processes used: 7

Length of task list: 9

Number of processes used: 7

Window sizes used: [0.75, 0.85, 0.95, 1.05, 1.15, 1.25, 1.35, 1.45, 1.55]

7 processes used for testing 9 window sizes

Processing times for each image [3.1403, 3.4763, 3.619, 3.0892, 3.0807, 3.4167] with an average

Time elapsed so far... 557.8490999999999

Length of task list: 9

Number of processes used: 8

Length of task list: 9

Number of processes used: 8

Length of task list: 9

Number of processes used: 8

Length of task list: 9

Number of processes used: 8

Length of task list: 9

Number of processes used: 8

Length of task list: 9

Number of processes used: 8

Length of task list: 9

Number of processes used: 8

Window sizes used: [0.75, 0.85, 0.95, 1.05, 1.15, 1.25, 1.35, 1.45, 1.55]

8 processes used for testing 9 window sizes

Processing times for each image [3.3722, 3.7044, 3.6492, 3.724, 3.6313, 3.4893] with an average

Time elapsed so far... 579.4194999999999

Length of task list: 9

Number of processes used: 9

Length of task list: 9

Number of processes used: 9

Length of task list: 9

Number of processes used: 9

Length of task list: 9

Number of processes used: 9

Length of task list: 9

Number of processes used: 9

Length of task list: 9

Number of processes used: 9

Window sizes used: [0.75, 0.85, 0.95, 1.05, 1.15, 1.25, 1.35, 1.45, 1.55]

9 processes used for testing 9 window sizes

Processing times for each image [3.8438, 3.8628, 4.0178, 3.7769, 3.8172, 3.9555] with an average

Time elapsed so far... 602.6934999999999

Length of task list: 17

Number of processes used: 1

Length of task list: 17

Number of processes used: 1

Length of task list: 17

Number of processes used: 1

Length of task list: 17

Number of processes used: 1

Length of task list: 17

Number of processes used: 1

Length of task list: 17

Number of processes used: 1

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45]

1 processes used for testing 17 window sizes

Processing times for each image [15.629, 14.7625, 16.6981, 15.5009, 16.5083, 15.4949] with an average

Time elapsed so far... 697.2871999999999

Length of task list: 17

Number of processes used: 2

Length of task list: 17

Number of processes used: 2

Length of task list: 17
Number of processes used: 2

Length of task list: 17
Number of processes used: 2

Length of task list: 17
Number of processes used: 2

Length of task list: 17
Number of processes used: 2

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
2 processes used for testing 17 window sizes
Processing times for each image [8.8325, 8.5592, 8.1243, 9.178, 9.5195, 9.0564] with an average

Time elapsed so far... 750.5570999999999

Length of task list: 17
Number of processes used: 3

Length of task list: 17
Number of processes used: 3

Length of task list: 17
Number of processes used: 3

Length of task list: 17
Number of processes used: 3

Length of task list: 17
Number of processes used: 3

Length of task list: 17
Number of processes used: 3

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
3 processes used for testing 17 window sizes
Processing times for each image [6.0118, 6.2373, 6.2985, 6.326, 5.82, 6.3485] with an average of

Time elapsed so far... 787.5991999999999

Length of task list: 17
Number of processes used: 4

Length of task list: 17
Number of processes used: 4

Length of task list: 17
Number of processes used: 4

Length of task list: 17
Number of processes used: 4

Length of task list: 17
Number of processes used: 4

Length of task list: 17
Number of processes used: 4

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
4 processes used for testing 17 window sizes
Processing times for each image [5.4352, 5.243, 5.3752, 4.9716, 5.1732, 5.1795] with an average

Time elapsed so far... 818.9768999999999

Length of task list: 17
Number of processes used: 5

Length of task list: 17
Number of processes used: 5

Length of task list: 17
Number of processes used: 5

Length of task list: 17
Number of processes used: 5

Length of task list: 17
Number of processes used: 5

Length of task list: 17
Number of processes used: 5

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
5 processes used for testing 17 window sizes
Processing times for each image [4.5038, 4.8954, 4.815, 4.8928, 4.4363, 4.5666] with an average

Time elapsed so far... 847.0867999999999

Length of task list: 17
Number of processes used: 6

Length of task list: 17
Number of processes used: 6

Length of task list: 17
Number of processes used: 6

Length of task list: 17
Number of processes used: 6

Length of task list: 17
Number of processes used: 6

Length of task list: 17
Number of processes used: 6

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
6 processes used for testing 17 window sizes
Processing times for each image [4.2274, 4.3475, 4.641, 4.6059, 4.3203, 4.4228] with an average

Time elapsed so far... 873.6516999999999

Length of task list: 17
Number of processes used: 7

Length of task list: 17
Number of processes used: 7

Length of task list: 17
Number of processes used: 7

Length of task list: 17
Number of processes used: 7

Length of task list: 17
Number of processes used: 7

Length of task list: 17
Number of processes used: 7

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
7 processes used for testing 17 window sizes
Processing times for each image [4.306, 4.098, 4.0203, 4.3229, 4.1451, 4.1871] with an average

Time elapsed so far... 898.7310999999999

Length of task list: 17
Number of processes used: 8

Length of task list: 17
Number of processes used: 8

Length of task list: 17
Number of processes used: 8

Length of task list: 17
Number of processes used: 8

Length of task list: 17
Number of processes used: 8

Length of task list: 17
Number of processes used: 8

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
8 processes used for testing 17 window sizes
Processing times for each image [4.3945, 4.2005, 4.4209, 4.2903, 4.2424, 4.1479] with an average c

Time elapsed so far... 924.4275999999999

Length of task list: 17
Number of processes used: 9

Length of task list: 17
Number of processes used: 9

Length of task list: 17
Number of processes used: 9

Length of task list: 17
Number of processes used: 9

Length of task list: 17
Number of processes used: 9

Length of task list: 17
Number of processes used: 9

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
9 processes used for testing 17 window sizes
Processing times for each image [4.8929, 4.209, 4.5504, 4.2781, 4.638, 4.1787] with an average c

Time elapsed so far... 951.1746999999999

Length of task list: 17
Number of processes used: 10

Length of task list: 17
Number of processes used: 10

Length of task list: 17
Number of processes used: 10

Length of task list: 17
Number of processes used: 10

Length of task list: 17
Number of processes used: 10

Length of task list: 17
Number of processes used: 10

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
10 processes used for testing 17 window sizes
Processing times for each image [4.508, 4.5688, 4.5939, 4.68, 4.4683, 4.3958] with an average of

Time elapsed so far... 978.3894999999999

Length of task list: 17
Number of processes used: 11

Length of task list: 17
Number of processes used: 11

Length of task list: 17
Number of processes used: 11

Length of task list: 17
Number of processes used: 11

Length of task list: 17
Number of processes used: 11

Length of task list: 17
Number of processes used: 11

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
11 processes used for testing 17 window sizes
Processing times for each image [4.7681, 4.8006, 4.6331, 4.7264, 4.672, 4.7412] with an average

Time elapsed so far... 1006.7308999999999

Length of task list: 17
Number of processes used: 12

Length of task list: 17
Number of processes used: 12

Length of task list: 17
Number of processes used: 12

Length of task list: 17
Number of processes used: 12

Length of task list: 17
Number of processes used: 12

Length of task list: 17
Number of processes used: 12

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
12 processes used for testing 17 window sizes
Processing times for each image [5.1437, 4.924, 5.1061, 5.0348, 5.1289, 4.8367] with an average

Time elapsed so far... 1036.9051

Length of task list: 17
Number of processes used: 13

Length of task list: 17
Number of processes used: 13

Length of task list: 17
Number of processes used: 13

Length of task list: 17
Number of processes used: 13

Length of task list: 17
Number of processes used: 13

Length of task list: 17
Number of processes used: 13

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
13 processes used for testing 17 window sizes
Processing times for each image [5.3974, 5.4444, 5.1307, 5.3268, 5.3184, 5.3195] with an average

Time elapsed so far... 1068.8423

Length of task list: 17
Number of processes used: 14

Length of task list: 17
Number of processes used: 14

Length of task list: 17
Number of processes used: 14

Length of task list: 17
Number of processes used: 14

Length of task list: 17
Number of processes used: 14

Length of task list: 17
Number of processes used: 14

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
14 processes used for testing 17 window sizes
Processing times for each image [5.7238, 5.539, 5.6192, 5.57, 5.6041, 5.6192] with an average of

Time elapsed so far... 1102.5176000000001

Length of task list: 17
Number of processes used: 15

Length of task list: 17
Number of processes used: 15

Length of task list: 17
Number of processes used: 15

Length of task list: 17
Number of processes used: 15

Length of task list: 17
Number of processes used: 15

Length of task list: 17
Number of processes used: 15

Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
15 processes used for testing 17 window sizes
Processing times for each image [6.2245, 6.124, 6.054, 6.0136, 5.879, 5.9541] with an average of

Time elapsed so far... 1138.7668

Length of task list: 17
Number of processes used: 16

Length of task list: 17
Number of processes used: 16

```
Length of task list: 17
Number of processes used: 16
```

```
Length of task list: 17
Number of processes used: 16
```

```
Length of task list: 17
Number of processes used: 16
```

```
Length of task list: 17
Number of processes used: 16
```

```
Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
16 processes used for testing 17 window sizes
Processing times for each image [6.3875, 5.9231, 6.3259, 6.2455, 6.2223, 6.2862] with an average
```

```
Time elapsed so far... 1176.1573
```

```
Length of task list: 17
Number of processes used: 17
```

```
Length of task list: 17
Number of processes used: 17
```

```
Length of task list: 17
Number of processes used: 17
```

```
Length of task list: 17
Number of processes used: 17
```

```
Length of task list: 17
Number of processes used: 17
```

```
Length of task list: 17
Number of processes used: 17
```

```
Window sizes used: [0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45
17 processes used for testing 17 window sizes
Processing times for each image [6.3961, 6.9878, 6.2808, 6.2977, 6.8764, 6.6336] with an average
```

```
Time elapsed so far... 1215.6297000000002
```

```
In [115]: # Now get the boxes for all test images
```

```
use_default_number_of_processes = 1
```

```

for i in range(7):
    time_for_all_images = []
    test_images_with_boxes = []

    number_of_processes_used = int(len(scaleslist[i])/2) + 1
    for image in test_images:
        img = mpimg.imread(image)
        freeze_support()
        t0 = time.time()
        r = []
        r = parallelize_find_cars(img, number_of_processes_used, i)
        t1 = time.time()
        processing_time = round(t1-t0, 4)
        #print('Time taken for processing each image', processing_time)
        time_for_all_images.append(processing_time)
        img_with_boxes = np.copy(img)
        for box in r:
            cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
        #plt.figure(figsize=(16,9))
        #plt.imshow(img_with_boxes)
        test_images_with_boxes.append(img_with_boxes)
    fig = plt.figure(figsize=(16,9))
    plt.subplot(131)
    plt.imshow(test_images_with_boxes[0])
    plt.title('test_image_1')
    plt.subplot(132)
    plt.imshow(test_images_with_boxes[1])
    plt.title('test_image_2')
    plt.subplot(133)
    plt.imshow(test_images_with_boxes[2])
    plt.title('test_image_3')
    fig = plt.figure(figsize=(16,9))
    plt.subplot(231)
    plt.imshow(test_images_with_boxes[3])
    plt.title('test_image_4')
    plt.subplot(232)
    plt.imshow(test_images_with_boxes[4])
    plt.title('test_image_5')
    plt.subplot(233)
    plt.imshow(test_images_with_boxes[5])
    plt.title('test_image_6')
    fig.tight_layout()
    print(number_of_processes_used, 'processes used for testing', len(scaleslist[i]),
    print('Processing times for each image', time_for_all_images,
          'with an average of ', round(sum(time_for_all_images)/len(time_for_all_image
    print())

```

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

Length of task list: 1
Number of processes used: 1

1 processes used for testing 1 window sizes

Processing times for each image [1.6084, 1.4641, 1.2137, 1.4375, 1.5991, 1.9102] with an average

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

Length of task list: 3
Number of processes used: 2

2 processes used for testing 3 window sizes

Processing times for each image [1.8489, 1.5482, 1.6957, 1.5006, 1.9035, 2.1408] with an average

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

3 processes used for testing 5 window sizes

Processing times for each image [2.7476, 2.9411, 2.5272, 2.8785, 2.5518, 2.6258] with an average

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

Length of task list: 5
Number of processes used: 3

3 processes used for testing 5 window sizes

Processing times for each image [2.1728, 2.108, 2.0594, 2.1374, 2.3032, 2.2225] with an average

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

Length of task list: 6
Number of processes used: 4

4 processes used for testing 6 window sizes

Processing times for each image [3.7647, 3.2331, 3.7702, 3.5407, 3.622, 3.5731] with an average

Length of task list: 9
Number of processes used: 5

Length of task list: 9
Number of processes used: 5

Length of task list: 9
Number of processes used: 5

Length of task list: 9
Number of processes used: 5

Length of task list: 9
Number of processes used: 5

Length of task list: 9
Number of processes used: 5

5 processes used for testing 9 window sizes

Processing times for each image [3.1645, 3.1705, 3.0049, 2.9336, 3.3418, 2.9678] with an average

Length of task list: 17
Number of processes used: 9

Length of task list: 17
Number of processes used: 9

Length of task list: 17
Number of processes used: 9

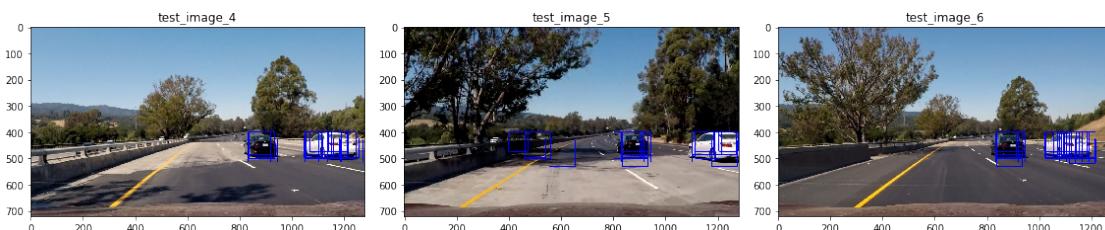
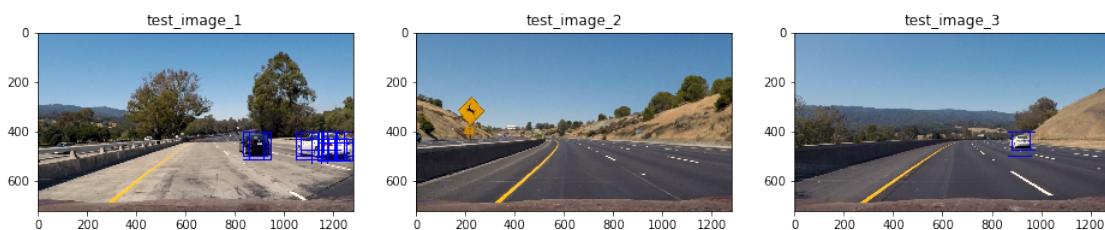
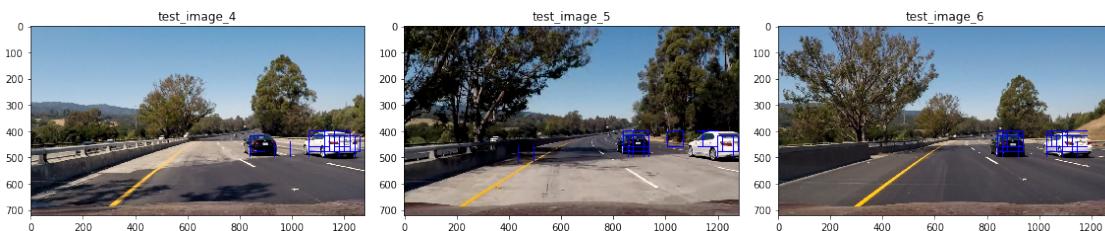
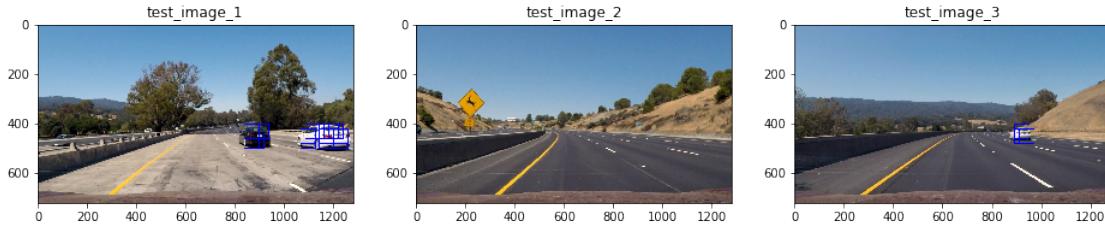
Length of task list: 17
Number of processes used: 9

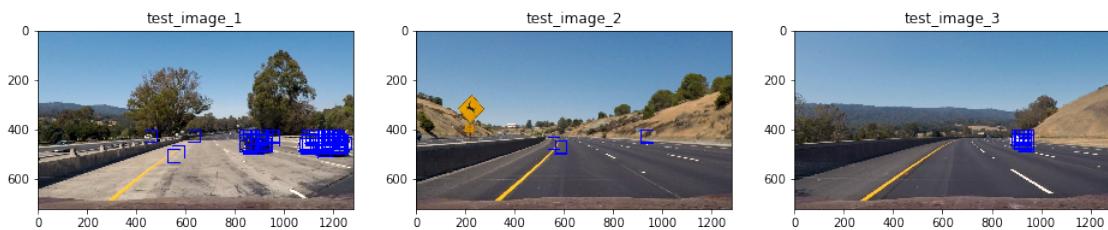
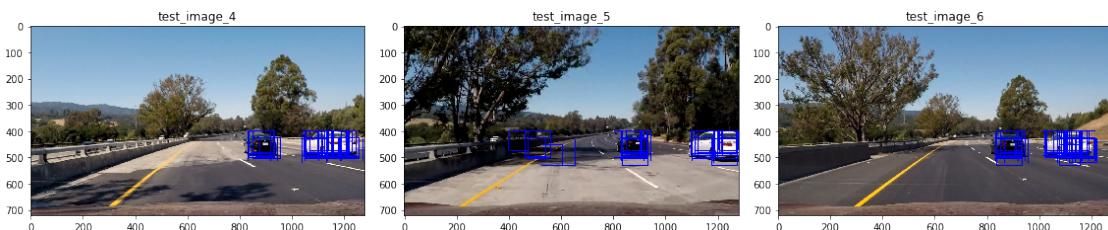
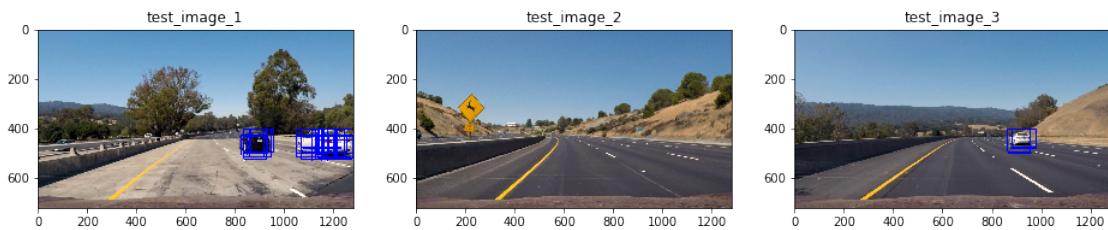
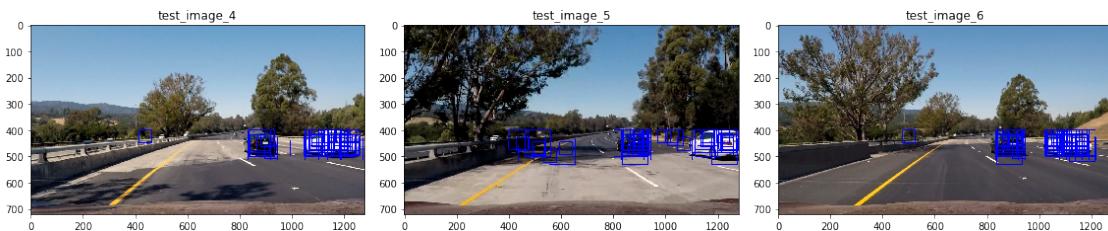
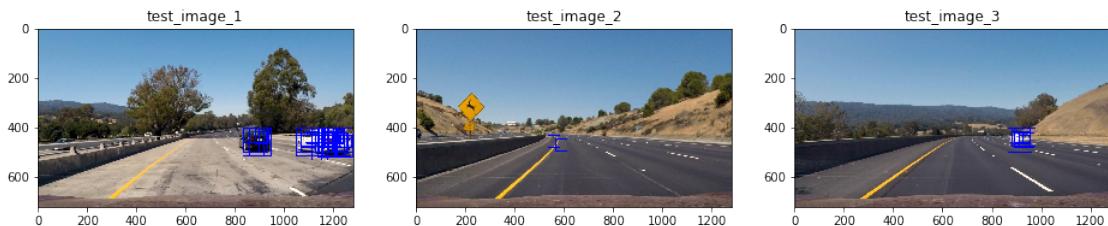
Length of task list: 17
Number of processes used: 9

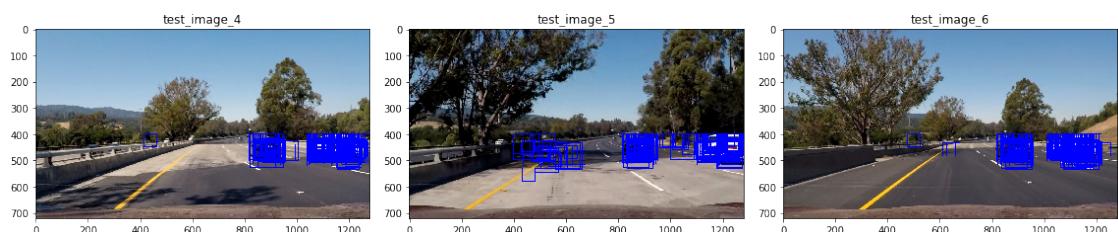
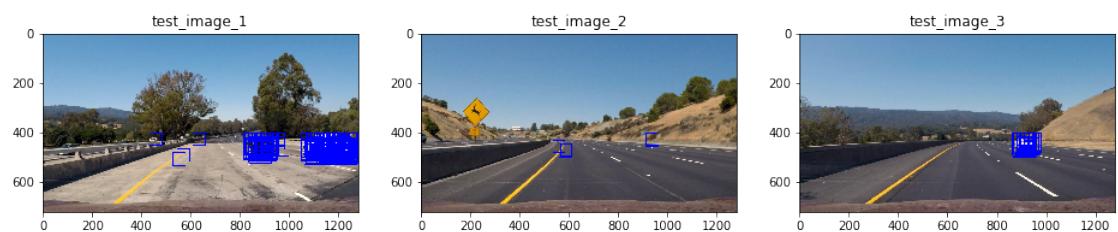
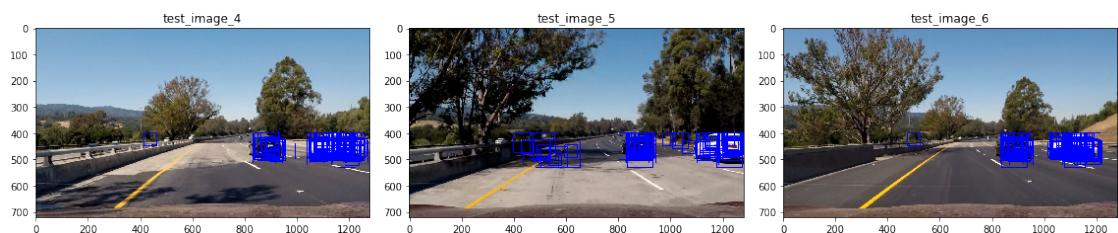
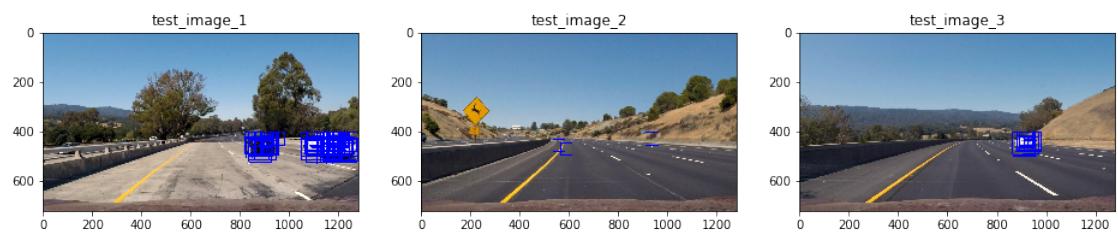
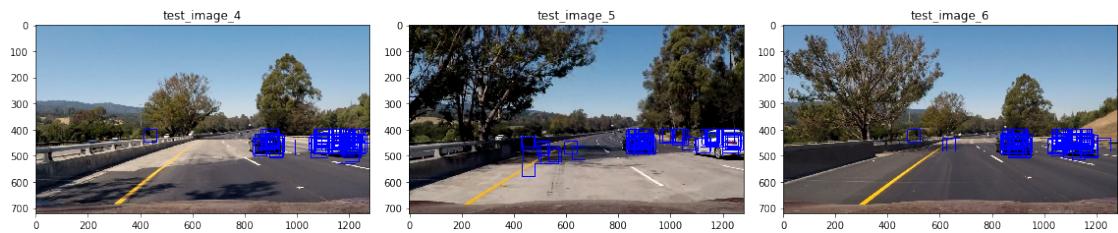
Length of task list: 17
Number of processes used: 9

9 processes used for testing 17 window sizes

Processing times for each image [4.1188, 4.1797, 4.3913, 4.2801, 4.1239, 4.1373] with an average







```

In [116]: print_find_car_step_times = 0
find_car_hog_parallelism_enabled = 0
def find_cars_new(img, ystart, ystop, xstart, xstop, scale, window, cells_per_step,\n                 svc, X_scaler, orient, pix_per_cell, cell_per_block, spatial_size, h
    global print_find_car_step_times
    global find_car_hog_parallelism_enabled
    time.sleep(0.5*random.random())
    #print(process_name + ', in the called function')
    rectangles = []
    #print(process_name + 'Xstart', xstart, 'Xstop:', xstop, 'Y start:', ystart, 'Y st
    find_cars_start_time = time.time()
    #import pdb; pdb.set_trace()

        if xstart > 1280 - window*scale or xstop > 1280 or ystart > 660 - window*scale or
            print(process_name + ': returning for wrong sizes')
            return rectangles

        #print(process_name + ', entering step 1...')
        #draw_img = np.copy(img)
        # Step 1: Divide with 255
        step1_start_time = time.time()
        try:
            img = img.astype(np.float32)/255
        except:
            print("Step 1: failed... could not initialize image as float!")
            raise

        step1_end_time = time.time()
        #print(process_name + ', exiting step 1...')
        step1_time = round(step1_end_time - step1_start_time, 6)

        # Step 2: Resize if scale is not 1
        #print(process_name + ', entering step 2...')
        step2_start_time = time.time()

        if xstop - xstart < scale*window or ystop - ystart < scale*window:
            print(process_name + ': step2: returning for bad sizes')
            print(process_name + ':Step2: Xstart', xstart, 'Xstop:', xstop, 'Y start:', y
            print("Step 2: failed... inappropriate start and stop sizes!")
            raise

```

```

try:
    img_tosearch = img[ystart:ystop,xstart:xstop]
except:
    raise

#print(process_name + ', patch shape...', img_tosearch.shape)
try:
    ctrans_tosearch = convert_color(img_tosearch, conv='RGB2YCrCb')
except:
    raise
#print(process_name + ', patch shape post color conversion...', ctrans_tosearch.shape)

try:
    if scale != 1:
        imshape = ctrans_tosearch.shape
        ctrans_tosearch = cv2.resize(ctrans_tosearch, (np.int(imshape[1]/scale), np.int(imshape[0]/scale)))
except:
    raise

#print(process_name + ', patch shape post resize...', ctrans_tosearch.shape)
step2_end_time = time.time()
#print(process_name + ', exiting step 2...')

step2_time = round(step2_end_time - step2_start_time, 6)

# Step 3: Get HOG channels
#print(process_name + ', entering step 3...')

step3_start_time = time.time()

try:
    if hog_channel == 'ALL':
        ch1 = ctrans_tosearch[:, :, 0]
        ch2 = ctrans_tosearch[:, :, 1]
        ch3 = ctrans_tosearch[:, :, 2]
    else:
        ch1 = ctrans_tosearch[:, :, hog_channel]
except:
    raise

step3_end_time = time.time()
#print(process_name + ', exiting step 3...')

step3_time = round(step3_end_time - step3_start_time, 6)

# Step 4: Define blocks and steps as above
#print(process_name + ', entering step 4...')

#print(process_name + ', channel shape st begin of step4...', ch1.shape)

```

```

step4_start_time = time.time()

nxblocks = (ch1.shape[1] // pix_per_cell) - cell_per_block + 1
nyblocks = (ch1.shape[0] // pix_per_cell) - cell_per_block + 1

nfeat_per_block = orient*cell_per_block**2

nblocks_per_window = (window // pix_per_cell) - cell_per_block + 1

nxsteps = (nxblocks - nblocks_per_window) // cells_per_step + 1
nysteps = (nyblocks - nblocks_per_window) // cells_per_step + 1

#print(process_name + ', step4, nx blocks', nxblocks)
#print(process_name + ', step4, ny blocks', nyblocks)
#print(process_name + ', step4, nx steps', nxsteps)
#print(process_name + ', step4, ny steps', nysteps)
#print(process_name + ', step4, nblocks_per_window', nblocks_per_window)

step4_end_time = time.time()
#print(process_name + ', exiting step 4...')

step4_time = round(step4_end_time - step4_start_time, 6)

# Step 5: Compute individual channel HOG features for the entire image
#print(process_name + ', entering step 5...')

step5_start_time = time.time()

if hog_channel == 'ALL':
    if find_car_hog_parallelism_enabled:
        pool = mp.Pool(processes=3)
        channels = [ch1, ch2, ch3]
        feature_vec = False
        hog = [pool.apply(get_hog_features, args=(ch, orient, pix_per_cell, cell_per_block,
                                                    for ch in channels)]
        #print(hog)
        hog1 = hog[0]
        hog2 = hog[1]
        hog3 = hog[2]
        #hog1_1 = get_hog_features(ch1, orient, pix_per_cell, cell_per_block, feature_vec)
        #hog2_1 = get_hog_features(ch2, orient, pix_per_cell, cell_per_block, feature_vec)
        #hog3_1 = get_hog_features(ch3, orient, pix_per_cell, cell_per_block, feature_vec)

        #print(hog1_1.shape, hog1_1.shape)

        #print('channel shape', channels[0].shape)
        #print(nyblocks, nxblocks, cell_per_block, cell_per_block, orient)
        hog1 = hog[0].reshape(nyblocks, nxblocks, cell_per_block, cell_per_block, cell_per_block,
        hog2 = hog[1].reshape(nyblocks, nxblocks, cell_per_block, cell_per_block, cell_per_block,

```

```

        hog3 = hog[2].reshape(nyblocks, nxblocks, cell_per_block, cell_per_block,
    else:
        hog1 = get_hog_features(ch1, orient, pix_per_cell, cell_per_block, feature
        hog2 = get_hog_features(ch2, orient, pix_per_cell, cell_per_block, feature
        hog3 = get_hog_features(ch3, orient, pix_per_cell, cell_per_block, feature
    else:
        hog1 = get_hog_features(ch1, orient, pix_per_cell, cell_per_block, feature_vec

step5_end_time = time.time()
#print(process_name + ', exiting step 5...')

step5_time = round(step5_end_time - step5_start_time, 6)

# Step 6: Misc initializations
#print(process_name + ', entering step 6...')

step6_start_time = time.time()
iterations = 0
x_positions = []
y_positions = []
x_real_positions = []
y_real_positions = []
window_sizes = []

step6_end_time = time.time()
#print(process_name + ', exiting step 6...')

step6_time = round(step6_end_time - step6_start_time, 6)

# Step 7: for loop
#print(process_name + ', entering step 7...')

step7_start_time = time.time()
for xb in range(nxsteps):
    for yb in range(nysteps):
        ypos = yb*cells_per_step
        xpos = xb*cells_per_step
        x_positions.append(xpos)
        y_positions.append(ypos)

        # Extract HOG for this patch
        hog_feat1 = hog1[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window]
        if hog_channel == 'ALL':
            hog_feat2 = hog2[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window]
            hog_feat3 = hog3[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window]
            hog_features = np.hstack((hog_feat1, hog_feat2, hog_feat3))
        else:
            hog_features = np.hstack((hog_feat1))

```

```

xleft = xpos*pix_per_cell
ytop = ypos*pix_per_cell

hog_features = hog_features.reshape(1, -1)

test_prediction = svc.predict(hog_features) # added

if test_prediction == 1:
    xbox_left = np.int(xleft*scale)
    ytop_draw = np.int(ytop*scale)
    win_draw = np.int(window*scale)

    x_real_positions.append(xbox_left)
    y_real_positions.append(ytop_draw)
    window_sizes.append(win_draw)

#cu2.rectangle(draw_img, (xbox_left+xstart, ytop_draw+ystart), (xbox_left+win_draw+xstart, ytop_draw+window_sizes[-1]+ystart), (0, 0, 255), 2)
rectangles.append(((xbox_left+xstart, ytop_draw+ystart),(xbox_left+win_draw+xstart, ytop_draw+window_sizes[-1]+ystart)))

step7_end_time = time.time()
#print(process_name + ', exiting step 7...')

step7_time = round(step7_end_time - step7_start_time, 6)

find_cars_end_time = time.time()

total_find_cars_time = round(find_cars_end_time - find_cars_start_time, 6)
all_steps_time = step1_time + step2_time + step3_time + step4_time + step5_time + step6_time + step7_time

if print_find_car_step_times:
    print(process_name + ', Step 1: Divide with 255, processing time:', step1_time, 'seconds')
    print(process_name + ', Step 2: Resize if scale is not 1:', step2_time, 'seconds')
    print(process_name + ', Step 3: Get HOG channels:', step3_time, 'seconds')
    print(process_name + ', Step 4: Define blocks and steps as above:', step4_time, 'seconds')
    print(process_name + ', Step 5: Compute individual channel HOG features for the blocks:', step5_time, 'seconds')
    print(process_name + ', Step 6: Misc initializations:', step6_time, 'seconds')
    print(process_name + ', Step 7: for loop:', step7_time, 'seconds')
    print()

    print(process_name + 'All steps time:', all_steps_time, 'seconds')
    print()

    print(process_name + 'Find cars processing time:', total_find_cars_time, 'seconds')
    print()

return (rectangles, total_find_cars_time)

```

```
In [117]: use_default_number_of_processes = 0
print_find_car_step_times = 1
find_car_hog_parallelism_enabled = 1

time_elapsed = 0
for i in range(3):
    for j in range(1, len(scaleslist[i])+1):
        time_for_all_images = []
        test_images_with_boxes = []
        number_of_processes_used = j
        for image in test_images:
            img = mpimg.imread(image)
            freeze_support()
            t0 = time.time()
            r = []
            r = parallelize_find_cars(img, number_of_processes_used, i)
            t1 = time.time()
            processing_time = round(t1-t0, 4)
            #print('Time taken for processing each image', processing_time)
            time_for_all_images.append(processing_time)
            img_with_boxes = np.copy(img)
            for box in r:
                cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
            #plt.figure(figsize=(16,9))
            #plt.imshow(img_with_boxes)
            test_images_with_boxes.append(img_with_boxes)

        print('Window sizes used:', scaleslist[i])
        print(number_of_processes_used, 'processes used for testing', len(scaleslist[i]))
        print('Processing times for each image', time_for_all_images, \
              'with an average of ', round(sum(time_for_all_images)/len(time_for_all_images), 4))
        print()
        time_elapsed += sum(time_for_all_images)
        print('Time elapsed so far...', time_elapsed)
        print()
    print()

Length of task list: 1
Number of processes used: 1
```

```
Process-1813, Step 1: Divide with 255, processing time: 0.037669 seconds
Process-1813, Step 2: Resize if scale is not 1: 0.018789 seconds
Process-1813, Step 3: Get HOG channels: 0.000141 seconds
Process-1813, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1813, Step 5: Compute individual channel HOG features for the entire image: 1.921702 seconds
Process-1813, Step 6: Misc initializations: 8e-06 seconds
Process-1813, Step 7: for loop: 0.063749 seconds
```

Process-1813All steps time: 2.042068 seconds

Process-1813, Find cars processing time: 2.042397 seconds

Length of task list: 1

Number of processes used: 1

Process-1814, Step 1: Divide with 255, processing time: 0.034919 seconds

Process-1814, Step 2: Resize if scale is not 1: 0.020033 seconds

Process-1814, Step 3: Get HOG channels: 0.000102 seconds

Process-1814, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1814, Step 5: Compute individual channel HOG features for the entire image: 2.044033 seconds

Process-1814, Step 6: Misc initializations: 1.9e-05 seconds

Process-1814, Step 7: for loop: 0.062511 seconds

Process-1814All steps time: 2.1616270000000006 seconds

Process-1814, Find cars processing time: 2.161966 seconds

Length of task list: 1

Number of processes used: 1

Process-1815, Step 1: Divide with 255, processing time: 0.03049 seconds

Process-1815, Step 2: Resize if scale is not 1: 0.022934 seconds

Process-1815, Step 3: Get HOG channels: 8.5e-05 seconds

Process-1815, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1815, Step 5: Compute individual channel HOG features for the entire image: 2.129241 seconds

Process-1815, Step 6: Misc initializations: 1.9e-05 seconds

Process-1815, Step 7: for loop: 0.071761 seconds

Process-1815All steps time: 2.254542 seconds

Process-1815, Find cars processing time: 2.254864 seconds

Length of task list: 1

Number of processes used: 1

Process-1816, Step 1: Divide with 255, processing time: 0.034991 seconds

Process-1816, Step 2: Resize if scale is not 1: 0.018177 seconds

Process-1816, Step 3: Get HOG channels: 8.2e-05 seconds

Process-1816, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1816, Step 5: Compute individual channel HOG features for the entire image: 2.106769 seconds

Process-1816, Step 6: Misc initializations: 1.4e-05 seconds

Process-1816, Step 7: for loop: 0.06396 seconds

Process-1816All steps time: 2.2240029999999997 seconds

Process-1816, Find cars processing time: 2.224294 seconds

Length of task list: 1
Number of processes used: 1

Process-1817, Step 1: Divide with 255, processing time: 0.029655 seconds
Process-1817, Step 2: Resize if scale is not 1: 0.020013 seconds
Process-1817, Step 3: Get HOG channels: 8.6e-05 seconds
Process-1817, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1817, Step 5: Compute individual channel HOG features for the entire image: 2.23342 seconds
Process-1817, Step 6: Misc initializations: 9e-06 seconds
Process-1817, Step 7: for loop: 0.062854 seconds

Process-1817All steps time: 2.3460470000000004 seconds

Process-1817, Find cars processing time: 2.346287 seconds

Length of task list: 1
Number of processes used: 1

Process-1818, Step 1: Divide with 255, processing time: 0.043637 seconds
Process-1818, Step 2: Resize if scale is not 1: 0.020318 seconds
Process-1818, Step 3: Get HOG channels: 9.9e-05 seconds
Process-1818, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1818, Step 5: Compute individual channel HOG features for the entire image: 1.999057 seconds
Process-1818, Step 6: Misc initializations: 1.3e-05 seconds
Process-1818, Step 7: for loop: 0.100862 seconds

Process-1818All steps time: 2.163998 seconds

Process-1818, Find cars processing time: 2.164259 seconds

Window sizes used: [0.95]
1 processes used for testing 1 window sizes
Processing times for each image [2.9243, 2.7486, 2.6903, 2.7133, 2.7679, 2.909] with an average

Time elapsed so far... 16.7534

Length of task list: 3
Number of processes used: 1

Process-1819, Step 1: Divide with 255, processing time: 0.028409 seconds
Process-1819, Step 2: Resize if scale is not 1: 0.020471 seconds
Process-1819, Step 3: Get HOG channels: 0.000101 seconds
Process-1819, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1819, Step 5: Compute individual channel HOG features for the entire image: 1.65298 seconds
Process-1819, Step 6: Misc initializations: 6e-06 seconds
Process-1819, Step 7: for loop: 0.041484 seconds

Process-1819All steps time: 1.743461999999998 seconds

Process-1819, Find cars processing time: 1.743772 seconds

Process-1819, Step 1: Divide with 255, processing time: 0.025925 seconds

Process-1819, Step 2: Resize if scale is not 1: 0.015649 seconds

Process-1819, Step 3: Get HOG channels: 1.6e-05 seconds

Process-1819, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1819, Step 5: Compute individual channel HOG features for the entire image: 1.489906 sec

Process-1819, Step 6: Misc initializations: 7e-06 seconds

Process-1819, Step 7: for loop: 0.030748 seconds

Process-1819All steps time: 1.5622580000000001 seconds

Process-1819, Find cars processing time: 1.562382 seconds

Process-1819, Step 1: Divide with 255, processing time: 0.027797 seconds

Process-1819, Step 2: Resize if scale is not 1: 0.012288 seconds

Process-1819, Step 3: Get HOG channels: 1.5e-05 seconds

Process-1819, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1819, Step 5: Compute individual channel HOG features for the entire image: 1.358734 sec

Process-1819, Step 6: Misc initializations: 1.4e-05 seconds

Process-1819, Step 7: for loop: 0.026585 seconds

Process-1819All steps time: 1.4254390000000001 seconds

Process-1819, Find cars processing time: 1.425586 seconds

Length of task list: 3

Number of processes used: 1

Process-1820, Step 1: Divide with 255, processing time: 0.030853 seconds

Process-1820, Step 2: Resize if scale is not 1: 0.02045 seconds

Process-1820, Step 3: Get HOG channels: 0.000128 seconds

Process-1820, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1820, Step 5: Compute individual channel HOG features for the entire image: 2.359686 sec

Process-1820, Step 6: Misc initializations: 1.6e-05 seconds

Process-1820, Step 7: for loop: 0.048286 seconds

Process-1820All steps time: 2.459428 seconds

Process-1820, Find cars processing time: 2.459739 seconds

Process-1820, Step 1: Divide with 255, processing time: 0.028605 seconds

Process-1820, Step 2: Resize if scale is not 1: 0.013297 seconds

Process-1820, Step 3: Get HOG channels: 3.7e-05 seconds

Process-1820, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1820, Step 5: Compute individual channel HOG features for the entire image: 1.53917 seconds
Process-1820, Step 6: Misc initializations: 1.2e-05 seconds
Process-1820, Step 7: for loop: 0.059177 seconds

Process-1820All steps time: 1.6403089999999998 seconds

Process-1820, Find cars processing time: 1.640494 seconds

Process-1820, Step 1: Divide with 255, processing time: 0.030628 seconds
Process-1820, Step 2: Resize if scale is not 1: 0.012034 seconds
Process-1820, Step 3: Get HOG channels: 2.3e-05 seconds
Process-1820, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1820, Step 5: Compute individual channel HOG features for the entire image: 1.45407 seconds
Process-1820, Step 6: Misc initializations: 7e-06 seconds
Process-1820, Step 7: for loop: 0.027333 seconds

Process-1820All steps time: 1.524102 seconds

Process-1820, Find cars processing time: 1.524235 seconds

Length of task list: 3
Number of processes used: 1

Process-1821, Step 1: Divide with 255, processing time: 0.025337 seconds
Process-1821, Step 2: Resize if scale is not 1: 0.02652 seconds
Process-1821, Step 3: Get HOG channels: 6.8e-05 seconds
Process-1821, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1821, Step 5: Compute individual channel HOG features for the entire image: 2.163803 seconds
Process-1821, Step 6: Misc initializations: 3e-06 seconds
Process-1821, Step 7: for loop: 0.057585 seconds

Process-1821All steps time: 2.273326 seconds

Process-1821, Find cars processing time: 2.273576 seconds

Process-1821, Step 1: Divide with 255, processing time: 0.029369 seconds
Process-1821, Step 2: Resize if scale is not 1: 0.012792 seconds
Process-1821, Step 3: Get HOG channels: 2.6e-05 seconds
Process-1821, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1821, Step 5: Compute individual channel HOG features for the entire image: 1.518975 seconds
Process-1821, Step 6: Misc initializations: 1e-05 seconds
Process-1821, Step 7: for loop: 0.037015 seconds

Process-1821All steps time: 1.598197 seconds

Process-1821, Find cars processing time: 1.598367 seconds

Process-1821, Step 1: Divide with 255, processing time: 0.026959 seconds

Process-1821, Step 2: Resize if scale is not 1: 0.011769 seconds
Process-1821, Step 3: Get HOG channels: 2.1e-05 seconds
Process-1821, Step 4: Define blocks and steps as above: 1.4e-05 seconds
Process-1821, Step 5: Compute individual channel HOG features for the entire image: 1.450516 seconds
Process-1821, Step 6: Misc initializations: 1.2e-05 seconds
Process-1821, Step 7: for loop: 0.03075 seconds

Process-1821All steps time: 1.520041 seconds

Process-1821, Find cars processing time: 1.520228 seconds

Length of task list: 3
Number of processes used: 1

Process-1822, Step 1: Divide with 255, processing time: 0.030334 seconds
Process-1822, Step 2: Resize if scale is not 1: 0.019107 seconds
Process-1822, Step 3: Get HOG channels: 0.000131 seconds
Process-1822, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1822, Step 5: Compute individual channel HOG features for the entire image: 1.777159 seconds
Process-1822, Step 6: Misc initializations: 3.1e-05 seconds
Process-1822, Step 7: for loop: 0.063529 seconds

Process-1822All steps time: 1.8903019999999997 seconds

Process-1822, Find cars processing time: 1.890591 seconds

Process-1822, Step 1: Divide with 255, processing time: 0.036885 seconds
Process-1822, Step 2: Resize if scale is not 1: 0.012655 seconds
Process-1822, Step 3: Get HOG channels: 1.8e-05 seconds
Process-1822, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1822, Step 5: Compute individual channel HOG features for the entire image: 1.500327 seconds
Process-1822, Step 6: Misc initializations: 8e-06 seconds
Process-1822, Step 7: for loop: 0.037125 seconds

Process-1822All steps time: 1.5870250000000001 seconds

Process-1822, Find cars processing time: 1.587208 seconds

Process-1822, Step 1: Divide with 255, processing time: 0.029611 seconds
Process-1822, Step 2: Resize if scale is not 1: 0.013101 seconds
Process-1822, Step 3: Get HOG channels: 2.5e-05 seconds
Process-1822, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1822, Step 5: Compute individual channel HOG features for the entire image: 1.415732 seconds
Process-1822, Step 6: Misc initializations: 6e-06 seconds
Process-1822, Step 7: for loop: 0.028803 seconds

Process-1822All steps time: 1.4872869999999998 seconds

Process-1822, Find cars processing time: 1.487431 seconds

Length of task list: 3

Number of processes used: 1

Process-1823, Step 1: Divide with 255, processing time: 0.028995 seconds

Process-1823, Step 2: Resize if scale is not 1: 0.018931 seconds

Process-1823, Step 3: Get HOG channels: 7e-05 seconds

Process-1823, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1823, Step 5: Compute individual channel HOG features for the entire image: 1.903538 seconds

Process-1823, Step 6: Misc initializations: 8e-06 seconds

Process-1823, Step 7: for loop: 0.04742 seconds

Process-1823All steps time: 1.998972 seconds

Process-1823, Find cars processing time: 1.999183 seconds

Process-1823, Step 1: Divide with 255, processing time: 0.029865 seconds

Process-1823, Step 2: Resize if scale is not 1: 0.012439 seconds

Process-1823, Step 3: Get HOG channels: 1.6e-05 seconds

Process-1823, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1823, Step 5: Compute individual channel HOG features for the entire image: 1.521036 seconds

Process-1823, Step 6: Misc initializations: 8e-06 seconds

Process-1823, Step 7: for loop: 0.032419 seconds

Process-1823All steps time: 1.595791 seconds

Process-1823, Find cars processing time: 1.595934 seconds

Process-1823, Step 1: Divide with 255, processing time: 0.023187 seconds

Process-1823, Step 2: Resize if scale is not 1: 0.011272 seconds

Process-1823, Step 3: Get HOG channels: 1.9e-05 seconds

Process-1823, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1823, Step 5: Compute individual channel HOG features for the entire image: 1.413973 seconds

Process-1823, Step 6: Misc initializations: 7e-06 seconds

Process-1823, Step 7: for loop: 0.02462 seconds

Process-1823All steps time: 1.4730850000000002 seconds

Process-1823, Find cars processing time: 1.473212 seconds

Length of task list: 3

Number of processes used: 1

Process-1824, Step 1: Divide with 255, processing time: 0.02705 seconds

Process-1824, Step 2: Resize if scale is not 1: 0.021065 seconds

Process-1824, Step 3: Get HOG channels: 0.000111 seconds

Process-1824, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1824, Step 5: Compute individual channel HOG features for the entire image: 1.887205 seconds
Process-1824, Step 6: Misc initializations: 2e-06 seconds
Process-1824, Step 7: for loop: 0.0422 seconds

Process-1824All steps time: 1.9776420000000001 seconds

Process-1824, Find cars processing time: 1.977845 seconds

Process-1824, Step 1: Divide with 255, processing time: 0.02292 seconds
Process-1824, Step 2: Resize if scale is not 1: 0.011683 seconds
Process-1824, Step 3: Get HOG channels: 1.4e-05 seconds
Process-1824, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1824, Step 5: Compute individual channel HOG features for the entire image: 1.478757 seconds
Process-1824, Step 6: Misc initializations: 8e-06 seconds
Process-1824, Step 7: for loop: 0.032579 seconds

Process-1824All steps time: 1.545969 seconds

Process-1824, Find cars processing time: 1.546083 seconds

Process-1824, Step 1: Divide with 255, processing time: 0.026619 seconds
Process-1824, Step 2: Resize if scale is not 1: 0.011326 seconds
Process-1824, Step 3: Get HOG channels: 1.7e-05 seconds
Process-1824, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1824, Step 5: Compute individual channel HOG features for the entire image: 1.336 seconds
Process-1824, Step 6: Misc initializations: 7e-06 seconds
Process-1824, Step 7: for loop: 0.024279 seconds

Process-1824All steps time: 1.398254 seconds

Process-1824, Find cars processing time: 1.398366 seconds

Window sizes used: [1.15, 1.35, 1.55]

1 processes used for testing 3 window sizes

Processing times for each image [5.9678, 7.3006, 6.3953, 5.8105, 6.0507, 5.7606] with an average

Time elapsed so far... 54.0389

Length of task list: 3

Number of processes used: 2

Process-1826, Step 1: Divide with 255, processing time: 0.028149 seconds
Process-1826, Step 2: Resize if scale is not 1: 0.016585 seconds
Process-1826, Step 3: Get HOG channels: 4.6e-05 seconds
Process-1826, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1826, Step 5: Compute individual channel HOG features for the entire image: 1.95886 seconds
Process-1826, Step 6: Misc initializations: 3e-06 seconds
Process-1826, Step 7: for loop: 0.034337 seconds

Process-1826All steps time: 2.037988 seconds

Process-1826, Find cars processing time: 2.038201 seconds

Process-1825, Step 1: Divide with 255, processing time: 0.02894 seconds

Process-1825, Step 2: Resize if scale is not 1: 0.018005 seconds

Process-1825, Step 3: Get HOG channels: 8.3e-05 seconds

Process-1825, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1825, Step 5: Compute individual channel HOG features for the entire image: 2.444316 seconds

Process-1825, Step 6: Misc initializations: 3e-06 seconds

Process-1825, Step 7: for loop: 0.04904 seconds

Process-1825All steps time: 2.5403970000000005 seconds

Process-1825, Find cars processing time: 2.540678 seconds

Process-1826, Step 1: Divide with 255, processing time: 0.025952 seconds

Process-1826, Step 2: Resize if scale is not 1: 0.010128 seconds

Process-1826, Step 3: Get HOG channels: 3.3e-05 seconds

Process-1826, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1826, Step 5: Compute individual channel HOG features for the entire image: 1.412792 seconds

Process-1826, Step 6: Misc initializations: 2e-06 seconds

Process-1826, Step 7: for loop: 0.025485 seconds

Process-1826All steps time: 1.4743970000000002 seconds

Process-1826, Find cars processing time: 1.474515 seconds

Length of task list: 3

Number of processes used: 2

Process-1827, Step 1: Divide with 255, processing time: 0.02728 seconds

Process-1827, Step 2: Resize if scale is not 1: 0.017982 seconds

Process-1827, Step 3: Get HOG channels: 8.3e-05 seconds

Process-1827, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1827, Step 5: Compute individual channel HOG features for the entire image: 2.203349 seconds

Process-1827, Step 6: Misc initializations: 2e-06 seconds

Process-1827, Step 7: for loop: 0.047458 seconds

Process-1827All steps time: 2.296163 seconds

Process-1827, Find cars processing time: 2.296435 seconds

Process-1828, Step 1: Divide with 255, processing time: 0.027961 seconds

Process-1828, Step 2: Resize if scale is not 1: 0.017853 seconds

Process-1828, Step 3: Get HOG channels: 5.5e-05 seconds

Process-1828, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1828, Step 5: Compute individual channel HOG features for the entire image: 1.867765 seconds
Process-1828, Step 6: Misc initializations: 1.3e-05 seconds
Process-1828, Step 7: for loop: 0.032573 seconds

Process-1828All steps time: 1.946227 seconds

Process-1828, Find cars processing time: 1.946458 seconds

Process-1827, Step 1: Divide with 255, processing time: 0.028513 seconds
Process-1827, Step 2: Resize if scale is not 1: 0.010671 seconds
Process-1827, Step 3: Get HOG channels: 3.3e-05 seconds
Process-1827, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1827, Step 5: Compute individual channel HOG features for the entire image: 1.350296 seconds
Process-1827, Step 6: Misc initializations: 2e-06 seconds
Process-1827, Step 7: for loop: 0.028196 seconds

Process-1827All steps time: 1.4177170000000001 seconds

Process-1827, Find cars processing time: 1.417868 seconds

Length of task list: 3
Number of processes used: 2

Process-1829, Step 1: Divide with 255, processing time: 0.027771 seconds
Process-1829, Step 2: Resize if scale is not 1: 0.01822 seconds
Process-1829, Step 3: Get HOG channels: 9.5e-05 seconds
Process-1829, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1829, Step 5: Compute individual channel HOG features for the entire image: 1.802597 seconds
Process-1829, Step 6: Misc initializations: 1.9e-05 seconds
Process-1829, Step 7: for loop: 0.047251 seconds

Process-1829All steps time: 1.8959629999999998 seconds

Process-1829, Find cars processing time: 1.896278 seconds

Process-1830, Step 1: Divide with 255, processing time: 0.02857 seconds
Process-1830, Step 2: Resize if scale is not 1: 0.015755 seconds
Process-1830, Step 3: Get HOG channels: 2.6e-05 seconds
Process-1830, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1830, Step 5: Compute individual channel HOG features for the entire image: 2.312573 seconds
Process-1830, Step 6: Misc initializations: 3e-06 seconds
Process-1830, Step 7: for loop: 0.038626 seconds

Process-1830All steps time: 2.3955599999999997 seconds

Process-1830, Find cars processing time: 2.395791 seconds

Process-1829, Step 1: Divide with 255, processing time: 0.026673 seconds

Process-1829, Step 2: Resize if scale is not 1: 0.010725 seconds
Process-1829, Step 3: Get HOG channels: 3.7e-05 seconds
Process-1829, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1829, Step 5: Compute individual channel HOG features for the entire image: 1.442153 seconds
Process-1829, Step 6: Misc initializations: 1.1e-05 seconds
Process-1829, Step 7: for loop: 0.022463 seconds

Process-1829All steps time: 1.502068 seconds

Process-1829, Find cars processing time: 1.502202 seconds

Length of task list: 3
Number of processes used: 2

Process-1832, Step 1: Divide with 255, processing time: 0.02396 seconds
Process-1832, Step 2: Resize if scale is not 1: 0.015733 seconds
Process-1832, Step 3: Get HOG channels: 3.2e-05 seconds
Process-1832, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1832, Step 5: Compute individual channel HOG features for the entire image: 1.790674 seconds
Process-1832, Step 6: Misc initializations: 7e-06 seconds
Process-1832, Step 7: for loop: 0.035623 seconds

Process-1832All steps time: 1.8660370000000002 seconds

Process-1832, Find cars processing time: 1.866249 seconds

Process-1831, Step 1: Divide with 255, processing time: 0.028021 seconds
Process-1831, Step 2: Resize if scale is not 1: 0.017999 seconds
Process-1831, Step 3: Get HOG channels: 8.4e-05 seconds
Process-1831, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1831, Step 5: Compute individual channel HOG features for the entire image: 1.968834 seconds
Process-1831, Step 6: Misc initializations: 2e-06 seconds
Process-1831, Step 7: for loop: 0.051664 seconds

Process-1831All steps time: 2.066614 seconds

Process-1831, Find cars processing time: 2.066915 seconds

Process-1832, Step 1: Divide with 255, processing time: 0.047332 seconds
Process-1832, Step 2: Resize if scale is not 1: 0.01179 seconds
Process-1832, Step 3: Get HOG channels: 2.1e-05 seconds
Process-1832, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1832, Step 5: Compute individual channel HOG features for the entire image: 1.399302 seconds
Process-1832, Step 6: Misc initializations: 1.3e-05 seconds
Process-1832, Step 7: for loop: 0.032919 seconds

Process-1832All steps time: 1.491384 seconds

Process-1832, Find cars processing time: 1.491558 seconds

Length of task list: 3

Number of processes used: 2

Process-1834, Step 1: Divide with 255, processing time: 0.042822 seconds

Process-1834, Step 2: Resize if scale is not 1: 0.016065 seconds

Process-1834, Step 3: Get HOG channels: 5.7e-05 seconds

Process-1834, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1834, Step 5: Compute individual channel HOG features for the entire image: 1.767364 seconds

Process-1834, Step 6: Misc initializations: 2e-06 seconds

Process-1834, Step 7: for loop: 0.036632 seconds

Process-1834All steps time: 1.8629499999999999 seconds

Process-1834, Find cars processing time: 1.863172 seconds

Process-1833, Step 1: Divide with 255, processing time: 0.034368 seconds

Process-1833, Step 2: Resize if scale is not 1: 0.026006 seconds

Process-1833, Step 3: Get HOG channels: 5.9e-05 seconds

Process-1833, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1833, Step 5: Compute individual channel HOG features for the entire image: 2.394703 seconds

Process-1833, Step 6: Misc initializations: 2e-06 seconds

Process-1833, Step 7: for loop: 0.048445 seconds

Process-1833All steps time: 2.5035939999999997 seconds

Process-1833, Find cars processing time: 2.503839 seconds

Process-1834, Step 1: Divide with 255, processing time: 0.027837 seconds

Process-1834, Step 2: Resize if scale is not 1: 0.011311 seconds

Process-1834, Step 3: Get HOG channels: 3.2e-05 seconds

Process-1834, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1834, Step 5: Compute individual channel HOG features for the entire image: 1.312103 seconds

Process-1834, Step 6: Misc initializations: 7e-06 seconds

Process-1834, Step 7: for loop: 0.024735 seconds

Process-1834All steps time: 1.37603 seconds

Process-1834, Find cars processing time: 1.376159 seconds

Length of task list: 3

Number of processes used: 2

Process-1836, Step 1: Divide with 255, processing time: 0.032778 seconds

Process-1836, Step 2: Resize if scale is not 1: 0.013379 seconds

Process-1836, Step 3: Get HOG channels: 1.9e-05 seconds

Process-1836, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1836, Step 5: Compute individual channel HOG features for the entire image: 1.481869 seconds
Process-1836, Step 6: Misc initializations: 2e-06 seconds
Process-1836, Step 7: for loop: 0.038433 seconds

Process-1836All steps time: 1.566486 seconds

Process-1836, Find cars processing time: 1.566639 seconds

Process-1835, Step 1: Divide with 255, processing time: 0.028127 seconds
Process-1835, Step 2: Resize if scale is not 1: 0.017355 seconds
Process-1835, Step 3: Get HOG channels: 8.4e-05 seconds
Process-1835, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1835, Step 5: Compute individual channel HOG features for the entire image: 1.695372 seconds
Process-1835, Step 6: Misc initializations: 7e-06 seconds
Process-1835, Step 7: for loop: 0.046038 seconds

Process-1835All steps time: 1.7869930000000003 seconds

Process-1835, Find cars processing time: 1.787232 seconds

Process-1836, Step 1: Divide with 255, processing time: 0.026427 seconds
Process-1836, Step 2: Resize if scale is not 1: 0.011091 seconds
Process-1836, Step 3: Get HOG channels: 3.8e-05 seconds
Process-1836, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1836, Step 5: Compute individual channel HOG features for the entire image: 1.313401 seconds
Process-1836, Step 6: Misc initializations: 1.6e-05 seconds
Process-1836, Step 7: for loop: 0.028243 seconds

Process-1836All steps time: 1.3792250000000001 seconds

Process-1836, Find cars processing time: 1.379356 seconds

Window sizes used: [1.15, 1.35, 1.55]

2 processes used for testing 3 window sizes

Processing times for each image [4.82, 4.6368, 4.4428, 4.5384, 4.3821, 4.0452] with an average of 4.3821

Time elapsed so far... 80.9042

Length of task list: 3

Number of processes used: 3

Process-1837, Step 1: Divide with 255, processing time: 0.026461 seconds
Process-1837, Step 2: Resize if scale is not 1: 0.020765 seconds
Process-1837, Step 3: Get HOG channels: 6.8e-05 seconds
Process-1837, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1837, Step 5: Compute individual channel HOG features for the entire image: 2.242209 seconds
Process-1837, Step 6: Misc initializations: 2e-06 seconds
Process-1839, Step 1: Divide with 255, processing time: 0.0259 seconds

Process-1837, Step 7: for loop: 0.050536 seconds
Process-1839, Step 2: Resize if scale is not 1: 0.014982 seconds

Process-1839, Step 3: Get HOG channels: 6.1e-05 seconds
Process-1839, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1837All steps time: 2.3400529999999997 seconds
Process-1839, Step 5: Compute individual channel HOG features for the entire image: 1.320557 seconds

Process-1839, Step 6: Misc initializations: 7e-06 seconds
Process-1839, Step 7: for loop: 0.027106 seconds
Process-1837, Find cars processing time: 2.340328 seconds

Process-1839All steps time: 1.388619 seconds

Process-1839, Find cars processing time: 1.388845 seconds

Process-1838, Step 1: Divide with 255, processing time: 0.026284 seconds
Process-1838, Step 2: Resize if scale is not 1: 0.01485 seconds
Process-1838, Step 3: Get HOG channels: 2.1e-05 seconds
Process-1838, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1838, Step 5: Compute individual channel HOG features for the entire image: 2.4636 seconds
Process-1838, Step 6: Misc initializations: 2e-06 seconds
Process-1838, Step 7: for loop: 0.038312 seconds

Process-1838All steps time: 2.543074999999995 seconds

Process-1838, Find cars processing time: 2.543272 seconds

Length of task list: 3
Number of processes used: 3

Process-1841, Step 1: Divide with 255, processing time: 0.03053 seconds
Process-1841, Step 2: Resize if scale is not 1: 0.015214 seconds
Process-1841, Step 3: Get HOG channels: 3.1e-05 seconds
Process-1841, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1841, Step 5: Compute individual channel HOG features for the entire image: 1.375829 seconds
Process-1841, Step 6: Misc initializations: 6e-06 seconds
Process-1841, Step 7: for loop: 0.030658 seconds

Process-1841All steps time: 1.452275 seconds

Process-1841, Find cars processing time: 1.452493 seconds

Process-1840, Step 1: Divide with 255, processing time: 0.02971 seconds
Process-1840, Step 2: Resize if scale is not 1: 0.017194 seconds
Process-1840, Step 3: Get HOG channels: 8.3e-05 seconds
Process-1840, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1840, Step 5: Compute individual channel HOG features for the entire image: 2.057058 seconds
Process-1840, Step 6: Misc initializations: 2e-06 seconds
Process-1840, Step 7: for loop: 0.055404 seconds

Process-1840All steps time: 2.1594599999999997 seconds

Process-1840, Find cars processing time: 2.159764 seconds

Process-1842, Step 1: Divide with 255, processing time: 0.02781 seconds
Process-1842, Step 2: Resize if scale is not 1: 0.016132 seconds
Process-1842, Step 3: Get HOG channels: 3.2e-05 seconds
Process-1842, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1842, Step 5: Compute individual channel HOG features for the entire image: 1.31179 seconds
Process-1842, Step 6: Misc initializations: 2e-06 seconds
Process-1842, Step 7: for loop: 0.025507 seconds

Process-1842All steps time: 1.3812820000000001 seconds

Process-1842, Find cars processing time: 1.381524 seconds

Length of task list: 3
Number of processes used: 3

Process-1843, Step 1: Divide with 255, processing time: 0.027556 seconds
Process-1843, Step 2: Resize if scale is not 1: 0.017597 seconds
Process-1843, Step 3: Get HOG channels: 8.9e-05 seconds
Process-1843, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1843, Step 5: Compute individual channel HOG features for the entire image: 1.659873 seconds
Process-1843, Step 6: Misc initializations: 1.5e-05 seconds
Process-1843, Step 7: for loop: 0.042649 seconds

Process-1843All steps time: 1.747788 seconds

Process-1843, Find cars processing time: 1.748039 seconds

Process-1844, Step 1: Divide with 255, processing time: 0.026478 seconds
Process-1844, Step 2: Resize if scale is not 1: 0.016264 seconds
Process-1844, Step 3: Get HOG channels: 3e-05 seconds
Process-1844, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1844, Step 5: Compute individual channel HOG features for the entire image: 1.467757 seconds
Process-1844, Step 6: Misc initializations: 2e-06 seconds
Process-1844, Step 7: for loop: 0.038053 seconds

Process-1844All steps time: 1.5485909999999998 seconds

Process-1844, Find cars processing time: 1.548794 seconds

Process-1845, Step 1: Divide with 255, processing time: 0.026129 seconds

Process-1845, Step 2: Resize if scale is not 1: 0.017149 seconds
Process-1845, Step 3: Get HOG channels: 4.9e-05 seconds
Process-1845, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1845, Step 5: Compute individual channel HOG features for the entire image: 1.317085 seconds
Process-1845, Step 6: Misc initializations: 6e-06 seconds
Process-1845, Step 7: for loop: 0.027312 seconds

Process-1845All steps time: 1.387738 seconds

Process-1845, Find cars processing time: 1.387984 seconds

Length of task list: 3
Number of processes used: 3

Process-1847, Step 1: Divide with 255, processing time: 0.029464 seconds
Process-1847, Step 2: Resize if scale is not 1: 0.014391 seconds
Process-1847, Step 3: Get HOG channels: 4.4e-05 seconds
Process-1847, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1847, Step 5: Compute individual channel HOG features for the entire image: 1.405991 seconds
Process-1847, Step 6: Misc initializations: 2e-06 seconds
Process-1847, Step 7: for loop: 0.039713 seconds

Process-1847All steps time: 1.4896129999999999 seconds

Process-1847, Find cars processing time: 1.489849 seconds

Process-1848, Step 1: Divide with 255, processing time: 0.027325 seconds
Process-1848, Step 2: Resize if scale is not 1: 0.016059 seconds
Process-1848, Step 3: Get HOG channels: 3.7e-05 seconds
Process-1848, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1848, Step 5: Compute individual channel HOG features for the entire image: 1.313341 seconds
Process-1848, Step 6: Misc initializations: 3e-06 seconds
Process-1848, Step 7: for loop: 0.027928 seconds

Process-1848All steps time: 1.3846990000000001 seconds

Process-1848, Find cars processing time: 1.384916 seconds

Process-1846, Step 1: Divide with 255, processing time: 0.032076 seconds
Process-1846, Step 2: Resize if scale is not 1: 0.019269 seconds
Process-1846, Step 3: Get HOG channels: 8.7e-05 seconds
Process-1846, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1846, Step 5: Compute individual channel HOG features for the entire image: 3.151665 seconds
Process-1846, Step 6: Misc initializations: 9e-06 seconds
Process-1846, Step 7: for loop: 0.047998 seconds

Process-1846All steps time: 3.251115 seconds

Process-1846, Find cars processing time: 3.251365 seconds

Length of task list: 3

Number of processes used: 3

Process-1850, Step 1: Divide with 255, processing time: 0.029109 seconds

Process-1850, Step 2: Resize if scale is not 1: 0.014772 seconds

Process-1850, Step 3: Get HOG channels: 3.3e-05 seconds

Process-1850, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1850, Step 5: Compute individual channel HOG features for the entire image: 1.457311 seconds

Process-1850, Step 6: Misc initializations: 7e-06 seconds

Process-1850, Step 7: for loop: 0.04495 seconds

Process-1850All steps time: 1.5461880000000001 seconds

Process-1850, Find cars processing time: 1.546397 seconds

Process-1849, Step 1: Divide with 255, processing time: 0.028114 seconds

Process-1849, Step 2: Resize if scale is not 1: 0.017694 seconds

Process-1849, Step 3: Get HOG channels: 8.8e-05 seconds

Process-1849, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1849, Step 5: Compute individual channel HOG features for the entire image: 1.703634 seconds

Process-1849, Step 6: Misc initializations: 5e-06 seconds

Process-1849, Step 7: for loop: 0.061515 seconds

Process-1849All steps time: 1.8110590000000002 seconds

Process-1849, Find cars processing time: 1.811323 seconds

Process-1851, Step 1: Divide with 255, processing time: 0.029683 seconds

Process-1851, Step 2: Resize if scale is not 1: 0.015977 seconds

Process-1851, Step 3: Get HOG channels: 2.4e-05 seconds

Process-1851, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1851, Step 5: Compute individual channel HOG features for the entire image: 1.377443 seconds

Process-1851, Step 6: Misc initializations: 2e-06 seconds

Process-1851, Step 7: for loop: 0.046785 seconds

Process-1851All steps time: 1.4699220000000002 seconds

Process-1851, Find cars processing time: 1.470148 seconds

Length of task list: 3

Number of processes used: 3

Process-1853, Step 1: Divide with 255, processing time: 0.029329 seconds

Process-1853, Step 2: Resize if scale is not 1: 0.016048 seconds

Process-1853, Step 3: Get HOG channels: 2.3e-05 seconds

Process-1853, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1853, Step 5: Compute individual channel HOG features for the entire image: 1.407724 seconds
Process-1853, Step 6: Misc initializations: 8e-06 seconds
Process-1853, Step 7: for loop: 0.051533 seconds

Process-1853All steps time: 1.504672 seconds

Process-1853, Find cars processing time: 1.504843 seconds

Process-1854, Step 1: Divide with 255, processing time: 0.03365 seconds
Process-1854, Step 2: Resize if scale is not 1: 0.014276 seconds
Process-1854, Step 3: Get HOG channels: 4.6e-05 seconds
Process-1854, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1854, Step 5: Compute individual channel HOG features for the entire image: 1.303065 seconds
Process-1854, Step 6: Misc initializations: 2e-06 seconds
Process-1854, Step 7: for loop: 0.025877 seconds

Process-1854All steps time: 1.376923 seconds

Process-1854, Find cars processing time: 1.377122 seconds

Process-1852, Step 1: Divide with 255, processing time: 0.02771 seconds
Process-1852, Step 2: Resize if scale is not 1: 0.017367 seconds
Process-1852, Step 3: Get HOG channels: 7.8e-05 seconds
Process-1852, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1852, Step 5: Compute individual channel HOG features for the entire image: 2.069461 seconds
Process-1852, Step 6: Misc initializations: 4e-06 seconds
Process-1852, Step 7: for loop: 0.042026 seconds

Process-1852All steps time: 2.156656 seconds

Process-1852, Find cars processing time: 2.156873 seconds

Window sizes used: [1.15, 1.35, 1.55]
3 processes used for testing 3 window sizes
Processing times for each image [3.708, 2.9161, 2.6806, 3.8772, 2.9535, 2.8736] with an average

Time elapsed so far... 99.9132

Length of task list: 5
Number of processes used: 1

Process-1855, Step 1: Divide with 255, processing time: 0.027217 seconds
Process-1855, Step 2: Resize if scale is not 1: 0.017078 seconds
Process-1855, Step 3: Get HOG channels: 7.1e-05 seconds
Process-1855, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1855, Step 5: Compute individual channel HOG features for the entire image: 2.484993 seconds
Process-1855, Step 6: Misc initializations: 9e-06 seconds

Process-1855, Step 7: for loop: 0.140479 seconds

Process-1855All steps time: 2.6698579999999996 seconds

Process-1855, Find cars processing time: 2.67015 seconds

Process-1855, Step 1: Divide with 255, processing time: 0.025477 seconds

Process-1855, Step 2: Resize if scale is not 1: 0.016625 seconds

Process-1855, Step 3: Get HOG channels: 6.7e-05 seconds

Process-1855, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1855, Step 5: Compute individual channel HOG features for the entire image: 1.982957 seconds

Process-1855, Step 6: Misc initializations: 1.4e-05 seconds

Process-1855, Step 7: for loop: 0.066824 seconds

Process-1855All steps time: 2.0919740000000004 seconds

Process-1855, Find cars processing time: 2.09218 seconds

Process-1855, Step 1: Divide with 255, processing time: 0.017438 seconds

Process-1855, Step 2: Resize if scale is not 1: 0.015168 seconds

Process-1855, Step 3: Get HOG channels: 3.4e-05 seconds

Process-1855, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1855, Step 5: Compute individual channel HOG features for the entire image: 1.706502 seconds

Process-1855, Step 6: Misc initializations: 6e-06 seconds

Process-1855, Step 7: for loop: 0.044638 seconds

Process-1855All steps time: 1.783794999999998 seconds

Process-1855, Find cars processing time: 1.783934 seconds

Process-1855, Step 1: Divide with 255, processing time: 0.030535 seconds

Process-1855, Step 2: Resize if scale is not 1: 0.011986 seconds

Process-1855, Step 3: Get HOG channels: 9e-06 seconds

Process-1855, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1855, Step 5: Compute individual channel HOG features for the entire image: 1.49712 seconds

Process-1855, Step 6: Misc initializations: 3e-06 seconds

Process-1855, Step 7: for loop: 0.034739 seconds

Process-1855All steps time: 1.574398 seconds

Process-1855, Find cars processing time: 1.574592 seconds

Process-1855, Step 1: Divide with 255, processing time: 0.027196 seconds

Process-1855, Step 2: Resize if scale is not 1: 0.010177 seconds

Process-1855, Step 3: Get HOG channels: 1.1e-05 seconds

Process-1855, Step 4: Define blocks and steps as above: 1.6e-05 seconds

Process-1855, Step 5: Compute individual channel HOG features for the entire image: 1.392033 seconds

Process-1855, Step 6: Misc initializations: 5e-06 seconds

Process-1855, Step 7: for loop: 0.025153 seconds

Process-1855All steps time: 1.4545910000000002 seconds

Process-1855, Find cars processing time: 1.454744 seconds

Length of task list: 5

Number of processes used: 1

Process-1856, Step 1: Divide with 255, processing time: 0.028466 seconds

Process-1856, Step 2: Resize if scale is not 1: 0.01517 seconds

Process-1856, Step 3: Get HOG channels: 0.00011 seconds

Process-1856, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1856, Step 5: Compute individual channel HOG features for the entire image: 3.222715 sec

Process-1856, Step 6: Misc initializations: 1.4e-05 seconds

Process-1856, Step 7: for loop: 0.142354 seconds

Process-1856All steps time: 3.408841000000002 seconds

Process-1856, Find cars processing time: 3.409191 seconds

Process-1856, Step 1: Divide with 255, processing time: 0.022819 seconds

Process-1856, Step 2: Resize if scale is not 1: 0.014755 seconds

Process-1856, Step 3: Get HOG channels: 5.9e-05 seconds

Process-1856, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1856, Step 5: Compute individual channel HOG features for the entire image: 2.157663 sec

Process-1856, Step 6: Misc initializations: 8.7e-05 seconds

Process-1856, Step 7: for loop: 0.081394 seconds

Process-1856All steps time: 2.276788 seconds

Process-1856, Find cars processing time: 2.277014 seconds

Process-1856, Step 1: Divide with 255, processing time: 0.036528 seconds

Process-1856, Step 2: Resize if scale is not 1: 0.015198 seconds

Process-1856, Step 3: Get HOG channels: 4.3e-05 seconds

Process-1856, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1856, Step 5: Compute individual channel HOG features for the entire image: 1.72833 sec

Process-1856, Step 6: Misc initializations: 1.3e-05 seconds

Process-1856, Step 7: for loop: 0.058835 seconds

Process-1856All steps time: 1.838957999999999 seconds

Process-1856, Find cars processing time: 1.83918 seconds

Process-1856, Step 1: Divide with 255, processing time: 0.02744 seconds

Process-1856, Step 2: Resize if scale is not 1: 0.014333 seconds

Process-1856, Step 3: Get HOG channels: 1e-05 seconds

Process-1856, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1856, Step 5: Compute individual channel HOG features for the entire image: 1.586478 seconds

Process-1856, Step 6: Misc initializations: 3e-06 seconds

Process-1856, Step 7: for loop: 0.039056 seconds

Process-1856All steps time: 1.667326 seconds

Process-1856, Find cars processing time: 1.66751 seconds

Process-1856, Step 1: Divide with 255, processing time: 0.031012 seconds

Process-1856, Step 2: Resize if scale is not 1: 0.011669 seconds

Process-1856, Step 3: Get HOG channels: 1.5e-05 seconds

Process-1856, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1856, Step 5: Compute individual channel HOG features for the entire image: 1.393412 seconds

Process-1856, Step 6: Misc initializations: 7e-06 seconds

Process-1856, Step 7: for loop: 0.027694 seconds

Process-1856All steps time: 1.4638200000000003 seconds

Process-1856, Find cars processing time: 1.463976 seconds

Length of task list: 5

Number of processes used: 1

Process-1857, Step 1: Divide with 255, processing time: 0.026844 seconds

Process-1857, Step 2: Resize if scale is not 1: 0.015093 seconds

Process-1857, Step 3: Get HOG channels: 0.0001 seconds

Process-1857, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1857, Step 5: Compute individual channel HOG features for the entire image: 2.927676 seconds

Process-1857, Step 6: Misc initializations: 9e-06 seconds

Process-1857, Step 7: for loop: 0.126497 seconds

Process-1857All steps time: 3.096229 seconds

Process-1857, Find cars processing time: 3.096497 seconds

Process-1857, Step 1: Divide with 255, processing time: 0.023606 seconds

Process-1857, Step 2: Resize if scale is not 1: 0.025347 seconds

Process-1857, Step 3: Get HOG channels: 4.5e-05 seconds

Process-1857, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1857, Step 5: Compute individual channel HOG features for the entire image: 2.012548 seconds

Process-1857, Step 6: Misc initializations: 8e-06 seconds

Process-1857, Step 7: for loop: 0.065213 seconds

Process-1857All steps time: 2.1267769999999997 seconds

Process-1857, Find cars processing time: 2.126915 seconds

Process-1857, Step 1: Divide with 255, processing time: 0.024133 seconds
Process-1857, Step 2: Resize if scale is not 1: 0.017735 seconds
Process-1857, Step 3: Get HOG channels: 3.4e-05 seconds
Process-1857, Step 4: Define blocks and steps as above: 2.2e-05 seconds
Process-1857, Step 5: Compute individual channel HOG features for the entire image: 1.843843 seconds
Process-1857, Step 6: Misc initializations: 1e-05 seconds
Process-1857, Step 7: for loop: 0.056446 seconds

Process-1857All steps time: 1.942223 seconds

Process-1857, Find cars processing time: 1.942371 seconds

Process-1857, Step 1: Divide with 255, processing time: 0.02677 seconds
Process-1857, Step 2: Resize if scale is not 1: 0.013384 seconds
Process-1857, Step 3: Get HOG channels: 1.2e-05 seconds
Process-1857, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1857, Step 5: Compute individual channel HOG features for the entire image: 1.512257 seconds
Process-1857, Step 6: Misc initializations: 8e-06 seconds
Process-1857, Step 7: for loop: 0.033267 seconds

Process-1857All steps time: 1.585707 seconds

Process-1857, Find cars processing time: 1.585875 seconds

Process-1857, Step 1: Divide with 255, processing time: 0.030303 seconds
Process-1857, Step 2: Resize if scale is not 1: 0.011955 seconds
Process-1857, Step 3: Get HOG channels: 7e-06 seconds
Process-1857, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1857, Step 5: Compute individual channel HOG features for the entire image: 1.40622 seconds
Process-1857, Step 6: Misc initializations: 2e-06 seconds
Process-1857, Step 7: for loop: 0.02391 seconds

Process-1857All steps time: 1.4724050000000002 seconds

Process-1857, Find cars processing time: 1.472536 seconds

Length of task list: 5
Number of processes used: 1

Process-1858, Step 1: Divide with 255, processing time: 0.024603 seconds
Process-1858, Step 2: Resize if scale is not 1: 0.018602 seconds
Process-1858, Step 3: Get HOG channels: 9.7e-05 seconds
Process-1858, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1858, Step 5: Compute individual channel HOG features for the entire image: 2.93642 seconds
Process-1858, Step 6: Misc initializations: 1.9e-05 seconds
Process-1858, Step 7: for loop: 0.129029 seconds

Process-1858All steps time: 3.10878 seconds

Process-1858, Find cars processing time: 3.109078 seconds

Process-1858, Step 1: Divide with 255, processing time: 0.028958 seconds

Process-1858, Step 2: Resize if scale is not 1: 0.01622 seconds

Process-1858, Step 3: Get HOG channels: 5.1e-05 seconds

Process-1858, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1858, Step 5: Compute individual channel HOG features for the entire image: 1.916012 seconds

Process-1858, Step 6: Misc initializations: 3e-06 seconds

Process-1858, Step 7: for loop: 0.065183 seconds

Process-1858All steps time: 2.0264360000000003 seconds

Process-1858, Find cars processing time: 2.026593 seconds

Process-1858, Step 1: Divide with 255, processing time: 0.020811 seconds

Process-1858, Step 2: Resize if scale is not 1: 0.01488 seconds

Process-1858, Step 3: Get HOG channels: 3.4e-05 seconds

Process-1858, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1858, Step 5: Compute individual channel HOG features for the entire image: 1.695203 seconds

Process-1858, Step 6: Misc initializations: 9e-06 seconds

Process-1858, Step 7: for loop: 0.05724 seconds

Process-1858All steps time: 1.788187 seconds

Process-1858, Find cars processing time: 1.788336 seconds

Process-1858, Step 1: Divide with 255, processing time: 0.028665 seconds

Process-1858, Step 2: Resize if scale is not 1: 0.012379 seconds

Process-1858, Step 3: Get HOG channels: 8e-06 seconds

Process-1858, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1858, Step 5: Compute individual channel HOG features for the entire image: 1.490614 seconds

Process-1858, Step 6: Misc initializations: 2e-06 seconds

Process-1858, Step 7: for loop: 0.040715 seconds

Process-1858All steps time: 1.5723900000000002 seconds

Process-1858, Find cars processing time: 1.572546 seconds

Process-1858, Step 1: Divide with 255, processing time: 0.028329 seconds

Process-1858, Step 2: Resize if scale is not 1: 0.014242 seconds

Process-1858, Step 3: Get HOG channels: 1.4e-05 seconds

Process-1858, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1858, Step 5: Compute individual channel HOG features for the entire image: 1.390377 seconds

Process-1858, Step 6: Misc initializations: 2e-06 seconds

Process-1858, Step 7: for loop: 0.025165 seconds

Process-1858All steps time: 1.4581400000000002 seconds

Process-1858, Find cars processing time: 1.458285 seconds

Length of task list: 5

Number of processes used: 1

Process-1859, Step 1: Divide with 255, processing time: 0.028268 seconds

Process-1859, Step 2: Resize if scale is not 1: 0.018371 seconds

Process-1859, Step 3: Get HOG channels: 6.6e-05 seconds

Process-1859, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1859, Step 5: Compute individual channel HOG features for the entire image: 2.678658 seconds

Process-1859, Step 6: Misc initializations: 3e-06 seconds

Process-1859, Step 7: for loop: 0.124467 seconds

Process-1859All steps time: 2.84984 seconds

Process-1859, Find cars processing time: 2.850039 seconds

Process-1859, Step 1: Divide with 255, processing time: 0.028094 seconds

Process-1859, Step 2: Resize if scale is not 1: 0.016105 seconds

Process-1859, Step 3: Get HOG channels: 7.2e-05 seconds

Process-1859, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1859, Step 5: Compute individual channel HOG features for the entire image: 2.023863 seconds

Process-1859, Step 6: Misc initializations: 1.7e-05 seconds

Process-1859, Step 7: for loop: 0.065339 seconds

Process-1859All steps time: 2.1334999999999997 seconds

Process-1859, Find cars processing time: 2.133677 seconds

Process-1859, Step 1: Divide with 255, processing time: 0.023203 seconds

Process-1859, Step 2: Resize if scale is not 1: 0.017501 seconds

Process-1859, Step 3: Get HOG channels: 6.5e-05 seconds

Process-1859, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1859, Step 5: Compute individual channel HOG features for the entire image: 1.764944 seconds

Process-1859, Step 6: Misc initializations: 6e-06 seconds

Process-1859, Step 7: for loop: 0.051259 seconds

Process-1859All steps time: 1.856987 seconds

Process-1859, Find cars processing time: 1.857127 seconds

Process-1859, Step 1: Divide with 255, processing time: 0.030416 seconds

Process-1859, Step 2: Resize if scale is not 1: 0.014849 seconds

Process-1859, Step 3: Get HOG channels: 1.8e-05 seconds

Process-1859, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-1859, Step 5: Compute individual channel HOG features for the entire image: 1.514693 seconds

Process-1859, Step 6: Misc initializations: 2e-06 seconds

Process-1859, Step 7: for loop: 0.033482 seconds

Process-1859All steps time: 1.593474 seconds

Process-1859, Find cars processing time: 1.593647 seconds

Process-1859, Step 1: Divide with 255, processing time: 0.026349 seconds

Process-1859, Step 2: Resize if scale is not 1: 0.01304 seconds

Process-1859, Step 3: Get HOG channels: 8e-06 seconds

Process-1859, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1859, Step 5: Compute individual channel HOG features for the entire image: 1.399622 seconds

Process-1859, Step 6: Misc initializations: 6e-06 seconds

Process-1859, Step 7: for loop: 0.025389 seconds

Process-1859All steps time: 1.46442 seconds

Process-1859, Find cars processing time: 1.464544 seconds

Length of task list: 5

Number of processes used: 1

Process-1860, Step 1: Divide with 255, processing time: 0.031926 seconds

Process-1860, Step 2: Resize if scale is not 1: 0.018326 seconds

Process-1860, Step 3: Get HOG channels: 8.1e-05 seconds

Process-1860, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1860, Step 5: Compute individual channel HOG features for the entire image: 2.928668 seconds

Process-1860, Step 6: Misc initializations: 1.3e-05 seconds

Process-1860, Step 7: for loop: 0.145014 seconds

Process-1860All steps time: 3.1240370000000004 seconds

Process-1860, Find cars processing time: 3.124294 seconds

Process-1860, Step 1: Divide with 255, processing time: 0.025195 seconds

Process-1860, Step 2: Resize if scale is not 1: 0.016782 seconds

Process-1860, Step 3: Get HOG channels: 7.3e-05 seconds

Process-1860, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1860, Step 5: Compute individual channel HOG features for the entire image: 2.164122 seconds

Process-1860, Step 6: Misc initializations: 1.8e-05 seconds

Process-1860, Step 7: for loop: 0.076789 seconds

Process-1860All steps time: 2.28299 seconds

Process-1860, Find cars processing time: 2.283156 seconds

Process-1860, Step 1: Divide with 255, processing time: 0.021987 seconds

Process-1860, Step 2: Resize if scale is not 1: 0.015031 seconds

Process-1860, Step 3: Get HOG channels: 3.7e-05 seconds

Process-1860, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1860, Step 5: Compute individual channel HOG features for the entire image: 1.786068 seconds

Process-1860, Step 6: Misc initializations: 1.4e-05 seconds

Process-1860, Step 7: for loop: 0.059282 seconds

Process-1860All steps time: 1.882428 seconds

Process-1860, Find cars processing time: 1.882593 seconds

Process-1860, Step 1: Divide with 255, processing time: 0.030342 seconds

Process-1860, Step 2: Resize if scale is not 1: 0.012465 seconds

Process-1860, Step 3: Get HOG channels: 8e-06 seconds

Process-1860, Step 4: Define blocks and steps as above: 1.7e-05 seconds

Process-1860, Step 5: Compute individual channel HOG features for the entire image: 1.577766 seconds

Process-1860, Step 6: Misc initializations: 6e-06 seconds

Process-1860, Step 7: for loop: 0.039299 seconds

Process-1860All steps time: 1.659903 seconds

Process-1860, Find cars processing time: 1.660036 seconds

Process-1860, Step 1: Divide with 255, processing time: 0.028687 seconds

Process-1860, Step 2: Resize if scale is not 1: 0.013258 seconds

Process-1860, Step 3: Get HOG channels: 9e-06 seconds

Process-1860, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1860, Step 5: Compute individual channel HOG features for the entire image: 1.404329 seconds

Process-1860, Step 6: Misc initializations: 4e-06 seconds

Process-1860, Step 7: for loop: 0.032385 seconds

Process-1860All steps time: 1.4786789999999996 seconds

Process-1860, Find cars processing time: 1.478806 seconds

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

1 processes used for testing 5 window sizes

Processing times for each image [11.3261, 12.3316, 12.1952, 11.267, 11.8752, 12.2554] with an average of 12.0000

Time elapsed so far... 171.1637

Length of task list: 5

Number of processes used: 2

Process-1861, Step 1: Divide with 255, processing time: 0.044275 seconds

Process-1861, Step 2: Resize if scale is not 1: 0.017145 seconds

Process-1861, Step 3: Get HOG channels: 0.000154 seconds

Process-1861, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1861, Step 5: Compute individual channel HOG features for the entire image: 2.728516 seconds

Process-1861, Step 6: Misc initializations: 1e-05 seconds

Process-1861, Step 7: for loop: 0.132909 seconds

Process-1861All steps time: 2.92302 seconds

Process-1861, Find cars processing time: 2.92333 seconds

Process-1862, Step 1: Divide with 255, processing time: 0.037729 seconds

Process-1862, Step 2: Resize if scale is not 1: 0.019415 seconds

Process-1862, Step 3: Get HOG channels: 7.8e-05 seconds

Process-1862, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1862, Step 5: Compute individual channel HOG features for the entire image: 2.544291 seconds

Process-1862, Step 6: Misc initializations: 3e-06 seconds

Process-1862, Step 7: for loop: 0.077491 seconds

Process-1862All steps time: 2.679016 seconds

Process-1862, Find cars processing time: 2.679308 seconds

Process-1862, Step 1: Divide with 255, processing time: 0.021286 seconds

Process-1862, Step 2: Resize if scale is not 1: 0.011965 seconds

Process-1862, Step 3: Get HOG channels: 2e-05 seconds

Process-1862, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1862, Step 5: Compute individual channel HOG features for the entire image: 1.453866 seconds

Process-1862, Step 6: Misc initializations: 1.5e-05 seconds

Process-1862, Step 7: for loop: 0.050431 seconds

Process-1862All steps time: 1.5375910000000002 seconds

Process-1862, Find cars processing time: 1.537768 seconds

Process-1861, Step 1: Divide with 255, processing time: 0.023892 seconds

Process-1861, Step 2: Resize if scale is not 1: 0.01478 seconds

Process-1861, Step 3: Get HOG channels: 5.9e-05 seconds

Process-1861, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1861, Step 5: Compute individual channel HOG features for the entire image: 1.673813 seconds

Process-1861, Step 6: Misc initializations: 1.5e-05 seconds

Process-1861, Step 7: for loop: 0.048028 seconds

Process-1861All steps time: 1.760597 seconds

Process-1861, Find cars processing time: 1.760738 seconds

Process-1862, Step 1: Divide with 255, processing time: 0.025791 seconds

Process-1862, Step 2: Resize if scale is not 1: 0.011853 seconds

Process-1862, Step 3: Get HOG channels: 1.6e-05 seconds

Process-1862, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1862, Step 5: Compute individual channel HOG features for the entire image: 1.37003 seconds

Process-1862, Step 6: Misc initializations: 7e-06 seconds

Process-1862, Step 7: for loop: 0.025149 seconds

Process-1862All steps time: 1.4328540000000003 seconds

Process-1862, Find cars processing time: 1.432984 seconds

Length of task list: 5

Number of processes used: 2

Process-1863, Step 1: Divide with 255, processing time: 0.032731 seconds

Process-1863, Step 2: Resize if scale is not 1: 0.019018 seconds

Process-1863, Step 3: Get HOG channels: 0.000181 seconds

Process-1863, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1863, Step 5: Compute individual channel HOG features for the entire image: 2.920598 seconds

Process-1863, Step 6: Misc initializations: 1.3e-05 seconds

Process-1863, Step 7: for loop: 0.127352 seconds

Process-1863All steps time: 3.099905 seconds

Process-1863, Find cars processing time: 3.100298 seconds

Process-1864, Step 1: Divide with 255, processing time: 0.041317 seconds

Process-1864, Step 2: Resize if scale is not 1: 0.020495 seconds

Process-1864, Step 3: Get HOG channels: 6.9e-05 seconds

Process-1864, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1864, Step 5: Compute individual channel HOG features for the entire image: 2.722132 seconds

Process-1864, Step 6: Misc initializations: 2.5e-05 seconds

Process-1864, Step 7: for loop: 0.076326 seconds

Process-1864All steps time: 2.860374 seconds

Process-1864, Find cars processing time: 2.860724 seconds

Process-1864, Step 1: Divide with 255, processing time: 0.020583 seconds

Process-1864, Step 2: Resize if scale is not 1: 0.01149 seconds

Process-1864, Step 3: Get HOG channels: 2.5e-05 seconds

Process-1864, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1864, Step 5: Compute individual channel HOG features for the entire image: 1.489435 seconds

Process-1864, Step 6: Misc initializations: 8e-06 seconds

Process-1864, Step 7: for loop: 0.032287 seconds

Process-1864All steps time: 1.553834 seconds

Process-1864, Find cars processing time: 1.553958 seconds

Process-1863, Step 1: Divide with 255, processing time: 0.023737 seconds

Process-1863, Step 2: Resize if scale is not 1: 0.016684 seconds

Process-1863, Step 3: Get HOG channels: 5.5e-05 seconds

Process-1863, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1863, Step 5: Compute individual channel HOG features for the entire image: 1.674984 seconds
Process-1863, Step 6: Misc initializations: 1.3e-05 seconds
Process-1863, Step 7: for loop: 0.045477 seconds

Process-1863All steps time: 1.760958 seconds

Process-1863, Find cars processing time: 1.761107 seconds

Process-1864, Step 1: Divide with 255, processing time: 0.02542 seconds
Process-1864, Step 2: Resize if scale is not 1: 0.012091 seconds
Process-1864, Step 3: Get HOG channels: 2.4e-05 seconds
Process-1864, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1864, Step 5: Compute individual channel HOG features for the entire image: 1.388628 seconds
Process-1864, Step 6: Misc initializations: 9e-06 seconds
Process-1864, Step 7: for loop: 0.023663 seconds

Process-1864All steps time: 1.449843 seconds

Process-1864, Find cars processing time: 1.449984 seconds

Length of task list: 5
Number of processes used: 2

Process-1866, Step 1: Divide with 255, processing time: 0.044313 seconds
Process-1866, Step 2: Resize if scale is not 1: 0.018021 seconds
Process-1866, Step 3: Get HOG channels: 6.9e-05 seconds
Process-1866, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-1866, Step 5: Compute individual channel HOG features for the entire image: 2.47482 seconds
Process-1866, Step 6: Misc initializations: 2e-06 seconds
Process-1866, Step 7: for loop: 0.075849 seconds

Process-1866All steps time: 2.6130869999999993 seconds

Process-1866, Find cars processing time: 2.613425 seconds

Process-1865, Step 1: Divide with 255, processing time: 0.033087 seconds
Process-1865, Step 2: Resize if scale is not 1: 0.016739 seconds
Process-1865, Step 3: Get HOG channels: 6.6e-05 seconds
Process-1865, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1865, Step 5: Compute individual channel HOG features for the entire image: 3.097002 seconds
Process-1865, Step 6: Misc initializations: 6e-06 seconds
Process-1865, Step 7: for loop: 0.142111 seconds

Process-1865All steps time: 3.289023 seconds

Process-1865, Find cars processing time: 3.28931 seconds

Process-1866, Step 1: Divide with 255, processing time: 0.021364 seconds
Process-1866, Step 2: Resize if scale is not 1: 0.015047 seconds
Process-1866, Step 3: Get HOG channels: 5.3e-05 seconds
Process-1866, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1866, Step 5: Compute individual channel HOG features for the entire image: 1.716717 seconds
Process-1866, Step 6: Misc initializations: 7e-06 seconds
Process-1866, Step 7: for loop: 0.0423 seconds

Process-1866All steps time: 1.7954990000000002 seconds

Process-1866, Find cars processing time: 1.795645 seconds

Process-1865, Step 1: Divide with 255, processing time: 0.022899 seconds
Process-1865, Step 2: Resize if scale is not 1: 0.011989 seconds
Process-1865, Step 3: Get HOG channels: 2.7e-05 seconds
Process-1865, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1865, Step 5: Compute individual channel HOG features for the entire image: 1.454943 seconds
Process-1865, Step 6: Misc initializations: 5e-06 seconds
Process-1865, Step 7: for loop: 0.033959 seconds

Process-1865All steps time: 1.5238300000000002 seconds

Process-1865, Find cars processing time: 1.523976 seconds

Process-1866, Step 1: Divide with 255, processing time: 0.02688 seconds
Process-1866, Step 2: Resize if scale is not 1: 0.011055 seconds
Process-1866, Step 3: Get HOG channels: 1.8e-05 seconds
Process-1866, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1866, Step 5: Compute individual channel HOG features for the entire image: 1.383246 seconds
Process-1866, Step 6: Misc initializations: 8e-06 seconds
Process-1866, Step 7: for loop: 0.026736 seconds

Process-1866All steps time: 1.4479490000000002 seconds

Process-1866, Find cars processing time: 1.44807 seconds

Length of task list: 5
Number of processes used: 2

Process-1867, Step 1: Divide with 255, processing time: 0.026335 seconds
Process-1867, Step 2: Resize if scale is not 1: 0.017742 seconds
Process-1867, Step 3: Get HOG channels: 4.9e-05 seconds
Process-1867, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1867, Step 5: Compute individual channel HOG features for the entire image: 2.878657 seconds
Process-1867, Step 6: Misc initializations: 1.7e-05 seconds
Process-1867, Step 7: for loop: 0.1233 seconds

Process-1867All steps time: 3.046112 seconds

Process-1867, Find cars processing time: 3.046345 seconds

Process-1868, Step 1: Divide with 255, processing time: 0.02516 seconds

Process-1868, Step 2: Resize if scale is not 1: 0.017971 seconds

Process-1868, Step 3: Get HOG channels: 8e-05 seconds

Process-1868, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1868, Step 5: Compute individual channel HOG features for the entire image: 2.772658 seconds

Process-1868, Step 6: Misc initializations: 2e-06 seconds

Process-1868, Step 7: for loop: 0.075627 seconds

Process-1868All steps time: 2.8915099999999994 seconds

Process-1868, Find cars processing time: 2.891844 seconds

Process-1867, Step 1: Divide with 255, processing time: 0.020583 seconds

Process-1867, Step 2: Resize if scale is not 1: 0.015002 seconds

Process-1867, Step 3: Get HOG channels: 4.4e-05 seconds

Process-1867, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1867, Step 5: Compute individual channel HOG features for the entire image: 1.646402 seconds

Process-1867, Step 6: Misc initializations: 8e-06 seconds

Process-1867, Step 7: for loop: 0.042941 seconds

Process-1867All steps time: 1.7249869999999998 seconds

Process-1867, Find cars processing time: 1.725132 seconds

Process-1868, Step 1: Divide with 255, processing time: 0.021831 seconds

Process-1868, Step 2: Resize if scale is not 1: 0.012505 seconds

Process-1868, Step 3: Get HOG channels: 4.9e-05 seconds

Process-1868, Step 4: Define blocks and steps as above: 1.6e-05 seconds

Process-1868, Step 5: Compute individual channel HOG features for the entire image: 1.527554 seconds

Process-1868, Step 6: Misc initializations: 3e-06 seconds

Process-1868, Step 7: for loop: 0.035452 seconds

Process-1868All steps time: 1.59741 seconds

Process-1868, Find cars processing time: 1.597583 seconds

Process-1867, Step 1: Divide with 255, processing time: 0.025599 seconds

Process-1867, Step 2: Resize if scale is not 1: 0.012002 seconds

Process-1867, Step 3: Get HOG channels: 1.6e-05 seconds

Process-1867, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1867, Step 5: Compute individual channel HOG features for the entire image: 1.40042 seconds

Process-1867, Step 6: Misc initializations: 5e-06 seconds

Process-1867, Step 7: for loop: 0.026477 seconds

Process-1867All steps time: 1.464525 seconds

Process-1867, Find cars processing time: 1.464656 seconds

Length of task list: 5

Number of processes used: 2

Process-1870, Step 1: Divide with 255, processing time: 0.028028 seconds

Process-1870, Step 2: Resize if scale is not 1: 0.020579 seconds

Process-1870, Step 3: Get HOG channels: 6.5e-05 seconds

Process-1870, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1870, Step 5: Compute individual channel HOG features for the entire image: 2.755311 seconds

Process-1870, Step 6: Misc initializations: 9e-06 seconds

Process-1870, Step 7: for loop: 0.074524 seconds

Process-1870All steps time: 2.8785269999999996 seconds

Process-1870, Find cars processing time: 2.878838 seconds

Process-1869, Step 1: Divide with 255, processing time: 0.027786 seconds

Process-1869, Step 2: Resize if scale is not 1: 0.017546 seconds

Process-1869, Step 3: Get HOG channels: 6.6e-05 seconds

Process-1869, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1869, Step 5: Compute individual channel HOG features for the entire image: 3.211562 seconds

Process-1869, Step 6: Misc initializations: 7e-06 seconds

Process-1869, Step 7: for loop: 0.128638 seconds

Process-1869All steps time: 3.385617 seconds

Process-1869, Find cars processing time: 3.385839 seconds

Process-1869, Step 1: Divide with 255, processing time: 0.021909 seconds

Process-1869, Step 2: Resize if scale is not 1: 0.012607 seconds

Process-1869, Step 3: Get HOG channels: 2.5e-05 seconds

Process-1869, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1869, Step 5: Compute individual channel HOG features for the entire image: 1.511581 seconds

Process-1869, Step 6: Misc initializations: 7e-06 seconds

Process-1869, Step 7: for loop: 0.03409 seconds

Process-1869All steps time: 1.5802260000000001 seconds

Process-1869, Find cars processing time: 1.580366 seconds

Process-1870, Step 1: Divide with 255, processing time: 0.01857 seconds

Process-1870, Step 2: Resize if scale is not 1: 0.017059 seconds

Process-1870, Step 3: Get HOG channels: 5.1e-05 seconds

Process-1870, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1870, Step 5: Compute individual channel HOG features for the entire image: 1.650416 seconds

Process-1870, Step 6: Misc initializations: 2e-06 seconds

Process-1870, Step 7: for loop: 0.044604 seconds

Process-1870All steps time: 1.7307140000000003 seconds

Process-1870, Find cars processing time: 1.730885 seconds

Process-1869, Step 1: Divide with 255, processing time: 0.026022 seconds

Process-1869, Step 2: Resize if scale is not 1: 0.011844 seconds

Process-1869, Step 3: Get HOG channels: 2.1e-05 seconds

Process-1869, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1869, Step 5: Compute individual channel HOG features for the entire image: 1.382437 seconds

Process-1869, Step 6: Misc initializations: 1.3e-05 seconds

Process-1869, Step 7: for loop: 0.026387 seconds

Process-1869All steps time: 1.4467329999999998 seconds

Process-1869, Find cars processing time: 1.446877 seconds

Length of task list: 5

Number of processes used: 2

Process-1871, Step 1: Divide with 255, processing time: 0.027308 seconds

Process-1871, Step 2: Resize if scale is not 1: 0.017494 seconds

Process-1871, Step 3: Get HOG channels: 7.3e-05 seconds

Process-1871, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1871, Step 5: Compute individual channel HOG features for the entire image: 3.043623 seconds

Process-1871, Step 6: Misc initializations: 9e-06 seconds

Process-1871, Step 7: for loop: 0.1233 seconds

Process-1871All steps time: 3.211819 seconds

Process-1871, Find cars processing time: 3.212058 seconds

Process-1872, Step 1: Divide with 255, processing time: 0.027888 seconds

Process-1872, Step 2: Resize if scale is not 1: 0.022903 seconds

Process-1872, Step 3: Get HOG channels: 6.4e-05 seconds

Process-1872, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1872, Step 5: Compute individual channel HOG features for the entire image: 2.690053 seconds

Process-1872, Step 6: Misc initializations: 1e-05 seconds

Process-1872, Step 7: for loop: 0.081653 seconds

Process-1872All steps time: 2.8225830000000003 seconds

Process-1872, Find cars processing time: 2.822855 seconds

Process-1871, Step 1: Divide with 255, processing time: 0.022 seconds

Process-1871, Step 2: Resize if scale is not 1: 0.014926 seconds

Process-1871, Step 3: Get HOG channels: 6.2e-05 seconds

Process-1871, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1871, Step 5: Compute individual channel HOG features for the entire image: 1.694123 seconds
Process-1871, Step 6: Misc initializations: 9e-06 seconds
Process-1871, Step 7: for loop: 0.046576 seconds

Process-1871All steps time: 1.777707 seconds

Process-1871, Find cars processing time: 1.777871 seconds

Process-1872, Step 1: Divide with 255, processing time: 0.022688 seconds
Process-1872, Step 2: Resize if scale is not 1: 0.01225 seconds
Process-1872, Step 3: Get HOG channels: 2e-05 seconds
Process-1872, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1872, Step 5: Compute individual channel HOG features for the entire image: 1.488171 seconds
Process-1872, Step 6: Misc initializations: 5e-06 seconds
Process-1872, Step 7: for loop: 0.039054 seconds

Process-1872All steps time: 1.5621949999999998 seconds

Process-1872, Find cars processing time: 1.562362 seconds

Process-1871, Step 1: Divide with 255, processing time: 0.023533 seconds
Process-1871, Step 2: Resize if scale is not 1: 0.0101 seconds
Process-1871, Step 3: Get HOG channels: 1.8e-05 seconds
Process-1871, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1871, Step 5: Compute individual channel HOG features for the entire image: 1.37417 seconds
Process-1871, Step 6: Misc initializations: 8e-06 seconds
Process-1871, Step 7: for loop: 0.025831 seconds

Process-1871All steps time: 1.4336659999999999 seconds

Process-1871, Find cars processing time: 1.433782 seconds

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]
2 processes used for testing 5 window sizes
Processing times for each image [7.1353, 7.206, 7.0318, 7.1488, 7.5612, 7.2596] with an average

Time elapsed so far... 214.5064

Length of task list: 5
Number of processes used: 3

Process-1875, Step 1: Divide with 255, processing time: 0.030183 seconds
Process-1875, Step 2: Resize if scale is not 1: 0.020727 seconds
Process-1875, Step 3: Get HOG channels: 9.9e-05 seconds
Process-1875, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-1875, Step 5: Compute individual channel HOG features for the entire image: 1.641138 seconds
Process-1875, Step 6: Misc initializations: 2e-06 seconds

Process-1875, Step 7: for loop: 0.047964 seconds

Process-1875All steps time: 1.7401259999999998 seconds

Process-1875, Find cars processing time: 1.740526 seconds

Process-1874, Step 1: Divide with 255, processing time: 0.025919 seconds

Process-1874, Step 2: Resize if scale is not 1: 0.016199 seconds

Process-1874, Step 3: Get HOG channels: 6.4e-05 seconds

Process-1874, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1874, Step 5: Compute individual channel HOG features for the entire image: 2.366458 seconds

Process-1874, Step 6: Misc initializations: 1e-05 seconds

Process-1874, Step 7: for loop: 0.069726 seconds

Process-1874All steps time: 2.4783870000000006 seconds

Process-1874, Find cars processing time: 2.478677 seconds

Process-1873, Step 1: Divide with 255, processing time: 0.021924 seconds

Process-1873, Step 2: Resize if scale is not 1: 0.018303 seconds

Process-1873, Step 3: Get HOG channels: 6.4e-05 seconds

Process-1873, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1873, Step 5: Compute individual channel HOG features for the entire image: 3.014995 seconds

Process-1873, Step 6: Misc initializations: 1.2e-05 seconds

Process-1873, Step 7: for loop: 0.131291 seconds

Process-1873All steps time: 3.186597 seconds

Process-1873, Find cars processing time: 3.186871 seconds

Process-1874, Step 1: Divide with 255, processing time: 0.01881 seconds

Process-1874, Step 2: Resize if scale is not 1: 0.011621 seconds

Process-1874, Step 3: Get HOG channels: 2.7e-05 seconds

Process-1874, Step 4: Define blocks and steps as above: 1.5e-05 seconds

Process-1874, Step 5: Compute individual channel HOG features for the entire image: 1.322768 seconds

Process-1874, Step 6: Misc initializations: 2e-06 seconds

Process-1874, Step 7: for loop: 0.025903 seconds

Process-1874All steps time: 1.379146 seconds

Process-1874, Find cars processing time: 1.379272 seconds

Process-1875, Step 1: Divide with 255, processing time: 0.028151 seconds

Process-1875, Step 2: Resize if scale is not 1: 0.013566 seconds

Process-1875, Step 3: Get HOG channels: 7.6e-05 seconds

Process-1875, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-1875, Step 5: Compute individual channel HOG features for the entire image: 1.560836 seconds

Process-1875, Step 6: Misc initializations: 2.2e-05 seconds

Process-1875, Step 7: for loop: 0.031043 seconds

Process-1875All steps time: 1.6337069999999998 seconds

Process-1875, Find cars processing time: 1.633898 seconds

Length of task list: 5

Number of processes used: 3

Process-1878, Step 1: Divide with 255, processing time: 0.026874 seconds

Process-1878, Step 2: Resize if scale is not 1: 0.029735 seconds

Process-1878, Step 3: Get HOG channels: 4.6e-05 seconds

Process-1878, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1878, Step 5: Compute individual channel HOG features for the entire image: 1.716001 sec

Process-1878, Step 6: Misc initializations: 2e-06 seconds

Process-1878, Step 7: for loop: 0.048538 seconds

Process-1878All steps time: 1.821206 seconds

Process-1878, Find cars processing time: 1.821482 seconds

Process-1877, Step 1: Divide with 255, processing time: 0.028194 seconds

Process-1877, Step 2: Resize if scale is not 1: 0.031636 seconds

Process-1877, Step 3: Get HOG channels: 5.8e-05 seconds

Process-1877, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1877, Step 5: Compute individual channel HOG features for the entire image: 2.114009 sec

Process-1877, Step 6: Misc initializations: 2e-06 seconds

Process-1877, Step 7: for loop: 0.071225 seconds

Process-1877All steps time: 2.2451329999999996 seconds

Process-1877, Find cars processing time: 2.245401 seconds

Process-1876, Step 1: Divide with 255, processing time: 0.026803 seconds

Process-1876, Step 2: Resize if scale is not 1: 0.019809 seconds

Process-1876, Step 3: Get HOG channels: 6.6e-05 seconds

Process-1876, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1876, Step 5: Compute individual channel HOG features for the entire image: 3.32657 sec

Process-1876, Step 6: Misc initializations: 2e-06 seconds

Process-1876, Step 7: for loop: 0.135265 seconds

Process-1876All steps time: 3.5085269999999995 seconds

Process-1876, Find cars processing time: 3.508795 seconds

Process-1878, Step 1: Divide with 255, processing time: 0.022391 seconds

Process-1878, Step 2: Resize if scale is not 1: 0.012148 seconds

Process-1878, Step 3: Get HOG channels: 4.3e-05 seconds

Process-1878, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1878, Step 5: Compute individual channel HOG features for the entire image: 1.404214 seconds

Process-1878, Step 6: Misc initializations: 5e-06 seconds

Process-1878, Step 7: for loop: 0.033499 seconds

Process-1878All steps time: 1.472307 seconds

Process-1878, Find cars processing time: 1.472446 seconds

Process-1877, Step 1: Divide with 255, processing time: 0.021699 seconds

Process-1877, Step 2: Resize if scale is not 1: 0.012339 seconds

Process-1877, Step 3: Get HOG channels: 3.5e-05 seconds

Process-1877, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1877, Step 5: Compute individual channel HOG features for the entire image: 1.39112 seconds

Process-1877, Step 6: Misc initializations: 7e-06 seconds

Process-1877, Step 7: for loop: 0.028489 seconds

Process-1877All steps time: 1.453698 seconds

Process-1877, Find cars processing time: 1.453948 seconds

Length of task list: 5

Number of processes used: 3

Process-1881, Step 1: Divide with 255, processing time: 0.028402 seconds

Process-1881, Step 2: Resize if scale is not 1: 0.017583 seconds

Process-1881, Step 3: Get HOG channels: 6.5e-05 seconds

Process-1881, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1881, Step 5: Compute individual channel HOG features for the entire image: 1.576564 seconds

Process-1881, Step 6: Misc initializations: 2e-06 seconds

Process-1881, Step 7: for loop: 0.05373 seconds

Process-1881All steps time: 1.676357 seconds

Process-1881, Find cars processing time: 1.67673 seconds

Process-1880, Step 1: Divide with 255, processing time: 0.0235 seconds

Process-1880, Step 2: Resize if scale is not 1: 0.019867 seconds

Process-1880, Step 3: Get HOG channels: 0.000107 seconds

Process-1880, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1880, Step 5: Compute individual channel HOG features for the entire image: 2.014786 seconds

Process-1880, Step 6: Misc initializations: 6e-06 seconds

Process-1880, Step 7: for loop: 0.063169 seconds

Process-1880All steps time: 2.121444 seconds

Process-1880, Find cars processing time: 2.121851 seconds

Process-1879, Step 1: Divide with 255, processing time: 0.02732 seconds
Process-1879, Step 2: Resize if scale is not 1: 0.019006 seconds
Process-1879, Step 3: Get HOG channels: 6.8e-05 seconds
Process-1879, Step 4: Define blocks and steps as above: 2.1e-05 seconds
Process-1879, Step 5: Compute individual channel HOG features for the entire image: 2.643853 seconds
Process-1879, Step 6: Misc initializations: 0.002514 seconds
Process-1879, Step 7: for loop: 0.154312 seconds

Process-1879All steps time: 2.8470940000000002 seconds

Process-1879, Find cars processing time: 2.847393 seconds

Process-1880, Step 1: Divide with 255, processing time: 0.019156 seconds
Process-1880, Step 2: Resize if scale is not 1: 0.009392 seconds
Process-1880, Step 3: Get HOG channels: 3.9e-05 seconds
Process-1880, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1880, Step 5: Compute individual channel HOG features for the entire image: 1.390229 seconds
Process-1880, Step 6: Misc initializations: 7e-06 seconds
Process-1880, Step 7: for loop: 0.02647 seconds

Process-1880All steps time: 1.445301 seconds

Process-1880, Find cars processing time: 1.445439 seconds

Process-1881, Step 1: Divide with 255, processing time: 0.043391 seconds
Process-1881, Step 2: Resize if scale is not 1: 0.01866 seconds
Process-1881, Step 3: Get HOG channels: 3.2e-05 seconds
Process-1881, Step 4: Define blocks and steps as above: 4e-05 seconds
Process-1881, Step 5: Compute individual channel HOG features for the entire image: 1.568521 seconds
Process-1881, Step 6: Misc initializations: 5e-06 seconds
Process-1881, Step 7: for loop: 0.035425 seconds

Process-1881All steps time: 1.666074 seconds

Process-1881, Find cars processing time: 1.666231 seconds

Length of task list: 5
Number of processes used: 3

Process-1884, Step 1: Divide with 255, processing time: 0.026911 seconds
Process-1884, Step 2: Resize if scale is not 1: 0.018706 seconds
Process-1884, Step 3: Get HOG channels: 8.3e-05 seconds
Process-1884, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1884, Step 5: Compute individual channel HOG features for the entire image: 1.706255 seconds
Process-1884, Step 6: Misc initializations: 2e-06 seconds
Process-1884, Step 7: for loop: 0.048872 seconds

Process-1884All steps time: 1.8008400000000002 seconds

Process-1884, Find cars processing time: 1.801137 seconds

Process-1883, Step 1: Divide with 255, processing time: 0.029469 seconds

Process-1883, Step 2: Resize if scale is not 1: 0.024742 seconds

Process-1883, Step 3: Get HOG channels: 5.8e-05 seconds

Process-1883, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1883, Step 5: Compute individual channel HOG features for the entire image: 2.173368 seconds

Process-1883, Step 6: Misc initializations: 2e-06 seconds

Process-1883, Step 7: for loop: 0.098628 seconds

Process-1883All steps time: 2.326279 seconds

Process-1883, Find cars processing time: 2.326536 seconds

Process-1882, Step 1: Divide with 255, processing time: 0.026387 seconds

Process-1882, Step 2: Resize if scale is not 1: 0.015518 seconds

Process-1882, Step 3: Get HOG channels: 6.8e-05 seconds

Process-1882, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1882, Step 5: Compute individual channel HOG features for the entire image: 2.98701 seconds

Process-1882, Step 6: Misc initializations: 1.3e-05 seconds

Process-1882, Step 7: for loop: 0.126782 seconds

Process-1882All steps time: 3.15579 seconds

Process-1882, Find cars processing time: 3.156045 seconds

Process-1884, Step 1: Divide with 255, processing time: 0.025626 seconds

Process-1884, Step 2: Resize if scale is not 1: 0.011717 seconds

Process-1884, Step 3: Get HOG channels: 3.1e-05 seconds

Process-1884, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1884, Step 5: Compute individual channel HOG features for the entire image: 1.475993 seconds

Process-1884, Step 6: Misc initializations: 5e-06 seconds

Process-1884, Step 7: for loop: 0.033123 seconds

Process-1884All steps time: 1.5465050000000002 seconds

Process-1884, Find cars processing time: 1.546649 seconds

Process-1883, Step 1: Divide with 255, processing time: 0.016774 seconds

Process-1883, Step 2: Resize if scale is not 1: 0.01174 seconds

Process-1883, Step 3: Get HOG channels: 2.4e-05 seconds

Process-1883, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1883, Step 5: Compute individual channel HOG features for the entire image: 1.430994 seconds

Process-1883, Step 6: Misc initializations: 1.3e-05 seconds

Process-1883, Step 7: for loop: 0.027607 seconds

Process-1883All steps time: 1.487158 seconds

Process-1883, Find cars processing time: 1.487295 seconds

Length of task list: 5

Number of processes used: 3

Process-1886, Step 1: Divide with 255, processing time: 0.035235 seconds

Process-1886, Step 2: Resize if scale is not 1: 0.02052 seconds

Process-1886, Step 3: Get HOG channels: 6.9e-05 seconds

Process-1886, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1886, Step 5: Compute individual channel HOG features for the entire image: 2.128059 seconds

Process-1886, Step 6: Misc initializations: 9e-06 seconds

Process-1886, Step 7: for loop: 0.09457 seconds

Process-1886All steps time: 2.278473 seconds

Process-1886, Find cars processing time: 2.278778 seconds

Process-1887, Step 1: Divide with 255, processing time: 0.038529 seconds

Process-1887, Step 2: Resize if scale is not 1: 0.018493 seconds

Process-1887, Step 3: Get HOG channels: 7.8e-05 seconds

Process-1887, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1887, Step 5: Compute individual channel HOG features for the entire image: 1.658916 seconds

Process-1887, Step 6: Misc initializations: 7e-06 seconds

Process-1887, Step 7: for loop: 0.048776 seconds

Process-1887All steps time: 1.7648110000000001 seconds

Process-1887, Find cars processing time: 1.765091 seconds

Process-1885, Step 1: Divide with 255, processing time: 0.031971 seconds

Process-1885, Step 2: Resize if scale is not 1: 0.016191 seconds

Process-1885, Step 3: Get HOG channels: 6.9e-05 seconds

Process-1885, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1885, Step 5: Compute individual channel HOG features for the entire image: 2.535609 seconds

Process-1885, Step 6: Misc initializations: 1e-05 seconds

Process-1885, Step 7: for loop: 0.132421 seconds

Process-1885All steps time: 2.716282 seconds

Process-1885, Find cars processing time: 2.71655 seconds

Process-1886, Step 1: Divide with 255, processing time: 0.029497 seconds

Process-1886, Step 2: Resize if scale is not 1: 0.013519 seconds

Process-1886, Step 3: Get HOG channels: 2.1e-05 seconds

Process-1886, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1886, Step 5: Compute individual channel HOG features for the entire image: 1.480022 seconds

Process-1886, Step 6: Misc initializations: 2e-06 seconds

Process-1886, Step 7: for loop: 0.032473 seconds

Process-1886All steps time: 1.555542 seconds

Process-1886, Find cars processing time: 1.555676 seconds

Process-1887, Step 1: Divide with 255, processing time: 0.026933 seconds

Process-1887, Step 2: Resize if scale is not 1: 0.010942 seconds

Process-1887, Step 3: Get HOG channels: 2.7e-05 seconds

Process-1887, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1887, Step 5: Compute individual channel HOG features for the entire image: 1.343606 seconds

Process-1887, Step 6: Misc initializations: 1.4e-05 seconds

Process-1887, Step 7: for loop: 0.036898 seconds

Process-1887All steps time: 1.4184290000000002 seconds

Process-1887, Find cars processing time: 1.418594 seconds

Length of task list: 5

Number of processes used: 3

Process-1889, Step 1: Divide with 255, processing time: 0.027236 seconds

Process-1889, Step 2: Resize if scale is not 1: 0.018276 seconds

Process-1889, Step 3: Get HOG channels: 6.4e-05 seconds

Process-1889, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1889, Step 5: Compute individual channel HOG features for the entire image: 1.923892 seconds

Process-1889, Step 6: Misc initializations: 2e-06 seconds

Process-1889, Step 7: for loop: 0.067639 seconds

Process-1889All steps time: 2.0371189999999997 seconds

Process-1889, Find cars processing time: 2.037383 seconds

Process-1890, Step 1: Divide with 255, processing time: 0.028631 seconds

Process-1890, Step 2: Resize if scale is not 1: 0.019971 seconds

Process-1890, Step 3: Get HOG channels: 6.1e-05 seconds

Process-1890, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1890, Step 5: Compute individual channel HOG features for the entire image: 1.626783 seconds

Process-1890, Step 6: Misc initializations: 8e-06 seconds

Process-1890, Step 7: for loop: 0.051057 seconds

Process-1890All steps time: 1.726521 seconds

Process-1890, Find cars processing time: 1.726743 seconds

Process-1888, Step 1: Divide with 255, processing time: 0.036622 seconds

Process-1888, Step 2: Resize if scale is not 1: 0.018907 seconds

Process-1888, Step 3: Get HOG channels: 6.9e-05 seconds

Process-1888, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-1888, Step 5: Compute individual channel HOG features for the entire image: 3.14297 seconds
Process-1888, Step 6: Misc initializations: 8e-06 seconds
Process-1888, Step 7: for loop: 0.107629 seconds

Process-1888All steps time: 3.306218 seconds

Process-1888, Find cars processing time: 3.30649 seconds

Process-1889, Step 1: Divide with 255, processing time: 0.018404 seconds
Process-1889, Step 2: Resize if scale is not 1: 0.01193 seconds
Process-1889, Step 3: Get HOG channels: 2.3e-05 seconds
Process-1889, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1889, Step 5: Compute individual channel HOG features for the entire image: 1.499226 seconds
Process-1889, Step 6: Misc initializations: 2e-06 seconds
Process-1889, Step 7: for loop: 0.033256 seconds

Process-1889All steps time: 1.562846 seconds

Process-1889, Find cars processing time: 1.562985 seconds

Process-1890, Step 1: Divide with 255, processing time: 0.025339 seconds
Process-1890, Step 2: Resize if scale is not 1: 0.011214 seconds
Process-1890, Step 3: Get HOG channels: 4.5e-05 seconds
Process-1890, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1890, Step 5: Compute individual channel HOG features for the entire image: 1.336608 seconds
Process-1890, Step 6: Misc initializations: 5e-06 seconds
Process-1890, Step 7: for loop: 0.026607 seconds

Process-1890All steps time: 1.3998240000000002 seconds

Process-1890, Find cars processing time: 1.399967 seconds

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]
3 processes used for testing 5 window sizes
Processing times for each image [5.3683, 5.0429, 4.9984, 5.077, 5.0453, 5.0144] with an average

Time elapsed so far... 245.05270000000002

Length of task list: 5
Number of processes used: 4

Process-1892, Step 1: Divide with 255, processing time: 0.031206 seconds
Process-1892, Step 2: Resize if scale is not 1: 0.020439 seconds
Process-1892, Step 3: Get HOG channels: 7.7e-05 seconds
Process-1892, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1892, Step 5: Compute individual channel HOG features for the entire image: 1.917236 seconds
Process-1892, Step 6: Misc initializations: 2e-06 seconds

Process-1892, Step 7: for loop: 0.073061 seconds

Process-1892All steps time: 2.042033 seconds

Process-1892, Find cars processing time: 2.042305 seconds

Process-1894, Step 1: Divide with 255, processing time: 0.023443 seconds

Process-1894, Step 2: Resize if scale is not 1: 0.01672 seconds

Process-1894, Step 3: Get HOG channels: 2.5e-05 seconds

Process-1894, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1894, Step 5: Compute individual channel HOG features for the entire image: 1.441101 seconds

Process-1894, Step 6: Misc initializations: 2e-06 seconds

Process-1894, Step 7: for loop: 0.035765 seconds

Process-1894All steps time: 1.517064 seconds

Process-1894, Find cars processing time: 1.517244 seconds

Process-1893, Step 1: Divide with 255, processing time: 0.027364 seconds

Process-1893, Step 2: Resize if scale is not 1: 0.017524 seconds

Process-1893, Step 3: Get HOG channels: 7.1e-05 seconds

Process-1893, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1893, Step 5: Compute individual channel HOG features for the entire image: 1.689678 seconds

Process-1893, Step 6: Misc initializations: 2e-06 seconds

Process-1893, Step 7: for loop: 0.058042 seconds

Process-1893All steps time: 1.792692 seconds

Process-1893, Find cars processing time: 1.792969 seconds

Process-1891, Step 1: Divide with 255, processing time: 0.028182 seconds

Process-1891, Step 2: Resize if scale is not 1: 0.016713 seconds

Process-1891, Step 3: Get HOG channels: 7.4e-05 seconds

Process-1891, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1891, Step 5: Compute individual channel HOG features for the entire image: 2.539479 seconds

Process-1891, Step 6: Misc initializations: 2e-06 seconds

Process-1891, Step 7: for loop: 0.138603 seconds

Process-1891All steps time: 2.723063 seconds

Process-1891, Find cars processing time: 2.723316 seconds

Process-1892, Step 1: Divide with 255, processing time: 0.023141 seconds

Process-1892, Step 2: Resize if scale is not 1: 0.010563 seconds

Process-1892, Step 3: Get HOG channels: 2.8e-05 seconds

Process-1892, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1892, Step 5: Compute individual channel HOG features for the entire image: 1.34008 seconds

Process-1892, Step 6: Misc initializations: 2e-06 seconds

Process-1892, Step 7: for loop: 0.026566 seconds

Process-1892All steps time: 1.400385 seconds

Process-1892, Find cars processing time: 1.400529 seconds

Length of task list: 5

Number of processes used: 4

Process-1896, Step 1: Divide with 255, processing time: 0.028858 seconds

Process-1896, Step 2: Resize if scale is not 1: 0.027931 seconds

Process-1896, Step 3: Get HOG channels: 0.000118 seconds

Process-1896, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1896, Step 5: Compute individual channel HOG features for the entire image: 2.028176 seconds

Process-1896, Step 6: Misc initializations: 2e-06 seconds

Process-1896, Step 7: for loop: 0.092427 seconds

Process-1896All steps time: 2.177521 seconds

Process-1896, Find cars processing time: 2.177819 seconds

Process-1897, Step 1: Divide with 255, processing time: 0.027525 seconds

Process-1897, Step 2: Resize if scale is not 1: 0.022536 seconds

Process-1897, Step 3: Get HOG channels: 7.7e-05 seconds

Process-1897, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-1897, Step 5: Compute individual channel HOG features for the entire image: 1.709521 seconds

Process-1897, Step 6: Misc initializations: 1e-06 seconds

Process-1897, Step 7: for loop: 0.042311 seconds

Process-1897All steps time: 1.801984 seconds

Process-1897, Find cars processing time: 1.802257 seconds

Process-1898, Step 1: Divide with 255, processing time: 0.030774 seconds

Process-1898, Step 2: Resize if scale is not 1: 0.017512 seconds

Process-1898, Step 3: Get HOG channels: 3.2e-05 seconds

Process-1898, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1898, Step 5: Compute individual channel HOG features for the entire image: 1.477391 seconds

Process-1898, Step 6: Misc initializations: 6e-06 seconds

Process-1898, Step 7: for loop: 0.04774 seconds

Process-1898All steps time: 1.5734629999999998 seconds

Process-1898, Find cars processing time: 1.573667 seconds

Process-1895, Step 1: Divide with 255, processing time: 0.026964 seconds

Process-1895, Step 2: Resize if scale is not 1: 0.025148 seconds

Process-1895, Step 3: Get HOG channels: 7.2e-05 seconds

Process-1895, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1895, Step 5: Compute individual channel HOG features for the entire image: 2.631937 seconds
Process-1895, Step 6: Misc initializations: 3e-06 seconds
Process-1895, Step 7: for loop: 0.147089 seconds

Process-1895All steps time: 2.831225 seconds

Process-1895, Find cars processing time: 2.831474 seconds

Process-1896, Step 1: Divide with 255, processing time: 0.022192 seconds
Process-1896, Step 2: Resize if scale is not 1: 0.011119 seconds
Process-1896, Step 3: Get HOG channels: 2.8e-05 seconds
Process-1896, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1896, Step 5: Compute individual channel HOG features for the entire image: 1.334264 seconds
Process-1896, Step 6: Misc initializations: 6e-06 seconds
Process-1896, Step 7: for loop: 0.025865 seconds

Process-1896All steps time: 1.393481 seconds

Process-1896, Find cars processing time: 1.393613 seconds

Length of task list: 5
Number of processes used: 4

Process-1901, Step 1: Divide with 255, processing time: 0.025436 seconds
Process-1901, Step 2: Resize if scale is not 1: 0.020205 seconds
Process-1901, Step 3: Get HOG channels: 7.7e-05 seconds
Process-1901, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1901, Step 5: Compute individual channel HOG features for the entire image: 1.719608 seconds
Process-1901, Step 6: Misc initializations: 1e-06 seconds
Process-1901, Step 7: for loop: 0.053554 seconds

Process-1901All steps time: 1.818892 seconds

Process-1901, Find cars processing time: 1.8192 seconds

Process-1899, Step 1: Divide with 255, processing time: 0.030959 seconds
Process-1899, Step 2: Resize if scale is not 1: 0.020314 seconds
Process-1899, Step 3: Get HOG channels: 6.8e-05 seconds
Process-1899, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1899, Step 5: Compute individual channel HOG features for the entire image: 2.573291 seconds
Process-1899, Step 6: Misc initializations: 8e-06 seconds
Process-1899, Step 7: for loop: 0.138406 seconds

Process-1899All steps time: 2.763057 seconds

Process-1899, Find cars processing time: 2.763327 seconds

Process-1900, Step 1: Divide with 255, processing time: 0.045019 seconds
Process-1900, Step 2: Resize if scale is not 1: 0.020459 seconds
Process-1900, Step 3: Get HOG channels: 7.1e-05 seconds
Process-1900, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1900, Step 5: Compute individual channel HOG features for the entire image: 2.065505 seconds
Process-1900, Step 6: Misc initializations: 8e-06 seconds
Process-1900, Step 7: for loop: 0.090019 seconds

Process-1900All steps time: 2.22109 seconds

Process-1900, Find cars processing time: 2.221349 seconds

Process-1902, Step 1: Divide with 255, processing time: 0.02877 seconds
Process-1902, Step 2: Resize if scale is not 1: 0.015966 seconds
Process-1902, Step 3: Get HOG channels: 2.5e-05 seconds
Process-1902, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1902, Step 5: Compute individual channel HOG features for the entire image: 1.508553 seconds
Process-1902, Step 6: Misc initializations: 7e-06 seconds
Process-1902, Step 7: for loop: 0.034253 seconds

Process-1902All steps time: 1.5875820000000003 seconds

Process-1902, Find cars processing time: 1.587816 seconds

Process-1901, Step 1: Divide with 255, processing time: 0.029988 seconds
Process-1901, Step 2: Resize if scale is not 1: 0.012616 seconds
Process-1901, Step 3: Get HOG channels: 4.5e-05 seconds
Process-1901, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-1901, Step 5: Compute individual channel HOG features for the entire image: 1.352745 seconds
Process-1901, Step 6: Misc initializations: 6e-06 seconds
Process-1901, Step 7: for loop: 0.028532 seconds

Process-1901All steps time: 1.423945 seconds

Process-1901, Find cars processing time: 1.424116 seconds

Length of task list: 5
Number of processes used: 4

Process-1905, Step 1: Divide with 255, processing time: 0.027675 seconds
Process-1905, Step 2: Resize if scale is not 1: 0.018087 seconds
Process-1905, Step 3: Get HOG channels: 7.3e-05 seconds
Process-1905, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1905, Step 5: Compute individual channel HOG features for the entire image: 1.777308 seconds
Process-1905, Step 6: Misc initializations: 2e-06 seconds
Process-1905, Step 7: for loop: 0.057278 seconds

Process-1905All steps time: 1.8804340000000002 seconds

Process-1905, Find cars processing time: 1.880823 seconds

Process-1903, Step 1: Divide with 255, processing time: 0.025504 seconds

Process-1903, Step 2: Resize if scale is not 1: 0.022061 seconds

Process-1903, Step 3: Get HOG channels: 0.000143 seconds

Process-1903, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1903, Step 5: Compute individual channel HOG features for the entire image: 2.492981 seconds

Process-1903, Step 6: Misc initializations: 5e-06 seconds

Process-1903, Step 7: for loop: 0.109558 seconds

Process-1903All steps time: 2.6502629999999994 seconds

Process-1903, Find cars processing time: 2.650676 seconds

Process-1904, Step 1: Divide with 255, processing time: 0.025469 seconds

Process-1904, Step 2: Resize if scale is not 1: 0.018946 seconds

Process-1904, Step 3: Get HOG channels: 8.2e-05 seconds

Process-1904, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1904, Step 5: Compute individual channel HOG features for the entire image: 1.963014 seconds

Process-1904, Step 6: Misc initializations: 2e-06 seconds

Process-1904, Step 7: for loop: 0.073831 seconds

Process-1904All steps time: 2.081356 seconds

Process-1904, Find cars processing time: 2.081622 seconds

Process-1906, Step 1: Divide with 255, processing time: 0.041687 seconds

Process-1906, Step 2: Resize if scale is not 1: 0.015165 seconds

Process-1906, Step 3: Get HOG channels: 6.3e-05 seconds

Process-1906, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1906, Step 5: Compute individual channel HOG features for the entire image: 1.444683 seconds

Process-1906, Step 6: Misc initializations: 8e-06 seconds

Process-1906, Step 7: for loop: 0.033106 seconds

Process-1906All steps time: 1.534719 seconds

Process-1906, Find cars processing time: 1.534923 seconds

Process-1905, Step 1: Divide with 255, processing time: 0.043592 seconds

Process-1905, Step 2: Resize if scale is not 1: 0.011664 seconds

Process-1905, Step 3: Get HOG channels: 2.8e-05 seconds

Process-1905, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1905, Step 5: Compute individual channel HOG features for the entire image: 1.363577 seconds

Process-1905, Step 6: Misc initializations: 2.1e-05 seconds

Process-1905, Step 7: for loop: 0.026383 seconds

Process-1905All steps time: 1.4452710000000002 seconds

Process-1905, Find cars processing time: 1.445426 seconds

Length of task list: 5

Number of processes used: 4

Process-1909, Step 1: Divide with 255, processing time: 0.031091 seconds

Process-1909, Step 2: Resize if scale is not 1: 0.018204 seconds

Process-1909, Step 3: Get HOG channels: 8.2e-05 seconds

Process-1909, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1909, Step 5: Compute individual channel HOG features for the entire image: 1.640638 seconds

Process-1909, Step 6: Misc initializations: 3e-06 seconds

Process-1909, Step 7: for loop: 0.047237 seconds

Process-1909All steps time: 1.737264 seconds

Process-1909, Find cars processing time: 1.737575 seconds

Process-1908, Step 1: Divide with 255, processing time: 0.02906 seconds

Process-1908, Step 2: Resize if scale is not 1: 0.020295 seconds

Process-1908, Step 3: Get HOG channels: 4.6e-05 seconds

Process-1908, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1908, Step 5: Compute individual channel HOG features for the entire image: 2.114861 seconds

Process-1908, Step 6: Misc initializations: 7e-06 seconds

Process-1908, Step 7: for loop: 0.0653 seconds

Process-1908All steps time: 2.229578 seconds

Process-1907, Step 1: Divide with 255, processing time: 0.033439 seconds

Process-1908, Find cars processing time: 2.229823 seconds

Process-1907, Step 2: Resize if scale is not 1: 0.017028 seconds

Process-1907, Step 3: Get HOG channels: 0.000178 seconds

Process-1907, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1907, Step 5: Compute individual channel HOG features for the entire image: 2.607721 seconds

Process-1907, Step 6: Misc initializations: 1.3e-05 seconds

Process-1907, Step 7: for loop: 0.149886 seconds

Process-1907All steps time: 2.8082770000000004 seconds

Process-1907, Find cars processing time: 2.808595 seconds

Process-1910, Step 1: Divide with 255, processing time: 0.031779 seconds

Process-1910, Step 2: Resize if scale is not 1: 0.0175 seconds

Process-1910, Step 3: Get HOG channels: 4.9e-05 seconds

Process-1910, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1910, Step 5: Compute individual channel HOG features for the entire image: 1.555659 seconds

Process-1910, Step 6: Misc initializations: 2e-06 seconds

Process-1910, Step 7: for loop: 0.029608 seconds

Process-1910All steps time: 1.6346070000000001 seconds

Process-1910, Find cars processing time: 1.634828 seconds

Process-1909, Step 1: Divide with 255, processing time: 0.02803 seconds

Process-1909, Step 2: Resize if scale is not 1: 0.010756 seconds

Process-1909, Step 3: Get HOG channels: 3.2e-05 seconds

Process-1909, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1909, Step 5: Compute individual channel HOG features for the entire image: 1.354407 seconds

Process-1909, Step 6: Misc initializations: 5e-06 seconds

Process-1909, Step 7: for loop: 0.030225 seconds

Process-1909All steps time: 1.4234619999999998 seconds

Process-1909, Find cars processing time: 1.423618 seconds

Length of task list: 5

Number of processes used: 4

Process-1914, Step 1: Divide with 255, processing time: 0.026802 seconds

Process-1914, Step 2: Resize if scale is not 1: 0.014487 seconds

Process-1914, Step 3: Get HOG channels: 3.9e-05 seconds

Process-1914, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1914, Step 5: Compute individual channel HOG features for the entire image: 1.427788 seconds

Process-1914, Step 6: Misc initializations: 7e-06 seconds

Process-1914, Step 7: for loop: 0.042421 seconds

Process-1914All steps time: 1.5115530000000001 seconds

Process-1914, Find cars processing time: 1.511769 seconds

Process-1913, Step 1: Divide with 255, processing time: 0.029111 seconds

Process-1913, Step 2: Resize if scale is not 1: 0.018872 seconds

Process-1913, Step 3: Get HOG channels: 6.6e-05 seconds

Process-1913, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1913, Step 5: Compute individual channel HOG features for the entire image: 1.670471 seconds

Process-1913, Step 6: Misc initializations: 2e-06 seconds

Process-1913, Step 7: for loop: 0.059241 seconds

Process-1913All steps time: 1.7777720000000001 seconds

Process-1913, Find cars processing time: 1.778018 seconds

Process-1912, Step 1: Divide with 255, processing time: 0.036346 seconds

Process-1912, Step 2: Resize if scale is not 1: 0.017618 seconds

Process-1912, Step 3: Get HOG channels: 7.1e-05 seconds

Process-1912, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1912, Step 5: Compute individual channel HOG features for the entire image: 2.177638 seconds
Process-1912, Step 6: Misc initializations: 6e-06 seconds
Process-1912, Step 7: for loop: 0.071006 seconds

Process-1912All steps time: 2.302697 seconds

Process-1912, Find cars processing time: 2.302988 seconds

Process-1911, Step 1: Divide with 255, processing time: 0.028193 seconds
Process-1911, Step 2: Resize if scale is not 1: 0.01686 seconds
Process-1911, Step 3: Get HOG channels: 7.4e-05 seconds
Process-1911, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1911, Step 5: Compute individual channel HOG features for the entire image: 2.535722 seconds
Process-1911, Step 6: Misc initializations: 2e-06 seconds
Process-1911, Step 7: for loop: 0.10742 seconds

Process-1911All steps time: 2.6882819999999996 seconds

Process-1911, Find cars processing time: 2.688523 seconds

Process-1914, Step 1: Divide with 255, processing time: 0.024868 seconds
Process-1914, Step 2: Resize if scale is not 1: 0.011884 seconds
Process-1914, Step 3: Get HOG channels: 5.5e-05 seconds
Process-1914, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1914, Step 5: Compute individual channel HOG features for the entire image: 1.400459 seconds
Process-1914, Step 6: Misc initializations: 2e-06 seconds
Process-1914, Step 7: for loop: 0.027252 seconds

Process-1914All steps time: 1.464529 seconds

Process-1914, Find cars processing time: 1.464704 seconds

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]
4 processes used for testing 5 window sizes
Processing times for each image [4.7485, 4.5815, 4.7886, 4.8715, 4.854, 5.0664] with an average

Time elapsed so far... 273.963200000000003

Length of task list: 5
Number of processes used: 5

Process-1917, Step 1: Divide with 255, processing time: 0.025985 seconds
Process-1917, Step 2: Resize if scale is not 1: 0.020261 seconds
Process-1917, Step 3: Get HOG channels: 8.6e-05 seconds
Process-1917, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1917, Step 5: Compute individual channel HOG features for the entire image: 1.627191 seconds
Process-1917, Step 6: Misc initializations: 2e-06 seconds

Process-1917, Step 7: for loop: 0.05847 seconds

Process-1917All steps time: 1.732007 seconds

Process-1917, Find cars processing time: 1.73225 seconds

Process-1919, Step 1: Divide with 255, processing time: 0.026207 seconds

Process-1919, Step 2: Resize if scale is not 1: 0.015197 seconds

Process-1919, Step 3: Get HOG channels: 4.7e-05 seconds

Process-1919, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1919, Step 5: Compute individual channel HOG features for the entire image: 1.371029 seconds

Process-1919, Step 6: Misc initializations: 8e-06 seconds

Process-1919, Step 7: for loop: 0.029261 seconds

Process-1919All steps time: 1.4417600000000002 seconds

Process-1919, Find cars processing time: 1.441964 seconds

Process-1918, Step 1: Divide with 255, processing time: 0.029758 seconds

Process-1918, Step 2: Resize if scale is not 1: 0.018628 seconds

Process-1918, Step 3: Get HOG channels: 3.8e-05 seconds

Process-1918, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1918, Step 5: Compute individual channel HOG features for the entire image: 1.517253 seconds

Process-1918, Step 6: Misc initializations: 2e-06 seconds

Process-1918, Step 7: for loop: 0.040061 seconds

Process-1918All steps time: 1.6057489999999999 seconds

Process-1918, Find cars processing time: 1.605947 seconds

Process-1916, Step 1: Divide with 255, processing time: 0.029177 seconds

Process-1916, Step 2: Resize if scale is not 1: 0.018574 seconds

Process-1916, Step 3: Get HOG channels: 6.5e-05 seconds

Process-1916, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1916, Step 5: Compute individual channel HOG features for the entire image: 2.20249 seconds

Process-1916, Step 6: Misc initializations: 2e-06 seconds

Process-1916, Step 7: for loop: 0.088197 seconds

Process-1916All steps time: 2.338515 seconds

Process-1916, Find cars processing time: 2.338726 seconds

Process-1915, Step 1: Divide with 255, processing time: 0.027853 seconds

Process-1915, Step 2: Resize if scale is not 1: 0.015448 seconds

Process-1915, Step 3: Get HOG channels: 8.6e-05 seconds

Process-1915, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-1915, Step 5: Compute individual channel HOG features for the entire image: 2.938646 seconds

Process-1915, Step 6: Misc initializations: 1.4e-05 seconds

Process-1915, Step 7: for loop: 0.127573 seconds

Process-1915All steps time: 3.109634 seconds

Process-1915, Find cars processing time: 3.109884 seconds

Length of task list: 5

Number of processes used: 5

Process-1922, Step 1: Divide with 255, processing time: 0.027567 seconds

Process-1922, Step 2: Resize if scale is not 1: 0.019534 seconds

Process-1922, Step 3: Get HOG channels: 6.8e-05 seconds

Process-1922, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1922, Step 5: Compute individual channel HOG features for the entire image: 1.641028 seconds

Process-1922, Step 6: Misc initializations: 8e-06 seconds

Process-1922, Step 7: for loop: 0.057149 seconds

Process-1922All steps time: 1.7453659999999998 seconds

Process-1922, Find cars processing time: 1.745633 seconds

Process-1923, Step 1: Divide with 255, processing time: 0.023739 seconds

Process-1923, Step 2: Resize if scale is not 1: 0.015079 seconds

Process-1923, Step 3: Get HOG channels: 3e-05 seconds

Process-1923, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1923, Step 5: Compute individual channel HOG features for the entire image: 1.522566 seconds

Process-1923, Step 6: Misc initializations: 9e-06 seconds

Process-1923, Step 7: for loop: 0.052635 seconds

Process-1923All steps time: 1.614068 seconds

Process-1923, Find cars processing time: 1.614285 seconds

Process-1924, Step 1: Divide with 255, processing time: 0.026874 seconds

Process-1924, Step 2: Resize if scale is not 1: 0.01467 seconds

Process-1924, Step 3: Get HOG channels: 4.4e-05 seconds

Process-1924, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1924, Step 5: Compute individual channel HOG features for the entire image: 1.255297 seconds

Process-1924, Step 6: Misc initializations: 2e-06 seconds

Process-1924, Step 7: for loop: 0.02682 seconds

Process-1924All steps time: 1.3237160000000003 seconds

Process-1924, Find cars processing time: 1.323933 seconds

Process-1921, Step 1: Divide with 255, processing time: 0.026813 seconds

Process-1921, Step 2: Resize if scale is not 1: 0.021156 seconds

Process-1921, Step 3: Get HOG channels: 7e-05 seconds

Process-1921, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1921, Step 5: Compute individual channel HOG features for the entire image: 2.040839 seconds
Process-1921, Step 6: Misc initializations: 8e-06 seconds
Process-1921, Step 7: for loop: 0.081699 seconds

Process-1921All steps time: 2.1705959999999997 seconds

Process-1921, Find cars processing time: 2.170875 seconds

Process-1920, Step 1: Divide with 255, processing time: 0.041547 seconds
Process-1920, Step 2: Resize if scale is not 1: 0.020848 seconds
Process-1920, Step 3: Get HOG channels: 7.7e-05 seconds
Process-1920, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1920, Step 5: Compute individual channel HOG features for the entire image: 2.671347 seconds
Process-1920, Step 6: Misc initializations: 1.7e-05 seconds
Process-1920, Step 7: for loop: 0.138506 seconds

Process-1920All steps time: 2.872354 seconds

Process-1920, Find cars processing time: 2.872621 seconds

Length of task list: 5
Number of processes used: 5

Process-1927, Step 1: Divide with 255, processing time: 0.027887 seconds
Process-1927, Step 2: Resize if scale is not 1: 0.017592 seconds
Process-1927, Step 3: Get HOG channels: 6.6e-05 seconds
Process-1927, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1927, Step 5: Compute individual channel HOG features for the entire image: 1.640705 seconds
Process-1927, Step 6: Misc initializations: 2e-06 seconds
Process-1927, Step 7: for loop: 0.061092 seconds

Process-1927All steps time: 1.747354 seconds

Process-1927, Find cars processing time: 1.747625 seconds

Process-1926, Step 1: Divide with 255, processing time: 0.025321 seconds
Process-1926, Step 2: Resize if scale is not 1: 0.021099 seconds
Process-1926, Step 3: Get HOG channels: 6.4e-05 seconds
Process-1926, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1926, Step 5: Compute individual channel HOG features for the entire image: 2.156639 seconds
Process-1926, Step 6: Misc initializations: 8e-06 seconds
Process-1926, Step 7: for loop: 0.080574 seconds

Process-1926All steps time: 2.283714 seconds

Process-1925, Step 1: Divide with 255, processing time: 0.025884 seconds
Process-1926, Find cars processing time: 2.28394 seconds

Process-1925, Step 2: Resize if scale is not 1: 0.017262 seconds

Process-1925, Step 3: Get HOG channels: 5.5e-05 seconds

Process-1925, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1925, Step 5: Compute individual channel HOG features for the entire image: 2.481415 seconds

Process-1925, Step 6: Misc initializations: 5e-06 seconds

Process-1925, Step 7: for loop: 0.140914 seconds

Process-1925All steps time: 2.665547 seconds

Process-1925, Find cars processing time: 2.665803 seconds

Process-1929, Step 1: Divide with 255, processing time: 0.027085 seconds

Process-1929, Step 2: Resize if scale is not 1: 0.015483 seconds

Process-1929, Step 3: Get HOG channels: 2.5e-05 seconds

Process-1929, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1929, Step 5: Compute individual channel HOG features for the entire image: 1.307158 seconds

Process-1929, Step 6: Misc initializations: 1.3e-05 seconds

Process-1929, Step 7: for loop: 0.026341 seconds

Process-1929All steps time: 1.376111 seconds

Process-1929, Find cars processing time: 1.3763 seconds

Process-1928, Step 1: Divide with 255, processing time: 0.028954 seconds

Process-1928, Step 2: Resize if scale is not 1: 0.017007 seconds

Process-1928, Step 3: Get HOG channels: 2.6e-05 seconds

Process-1928, Step 4: Define blocks and steps as above: 2.1e-05 seconds

Process-1928, Step 5: Compute individual channel HOG features for the entire image: 1.535856 seconds

Process-1928, Step 6: Misc initializations: 1e-05 seconds

Process-1928, Step 7: for loop: 0.049933 seconds

Process-1928All steps time: 1.6318070000000002 seconds

Process-1928, Find cars processing time: 1.632049 seconds

Length of task list: 5

Number of processes used: 5

Process-1932, Step 1: Divide with 255, processing time: 0.02211 seconds

Process-1932, Step 2: Resize if scale is not 1: 0.016849 seconds

Process-1932, Step 3: Get HOG channels: 9.8e-05 seconds

Process-1932, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1932, Step 5: Compute individual channel HOG features for the entire image: 1.809388 seconds

Process-1932, Step 6: Misc initializations: 2e-06 seconds

Process-1932, Step 7: for loop: 0.055131 seconds

Process-1932All steps time: 1.9035870000000001 seconds

Process-1932, Find cars processing time: 1.903978 seconds

Process-1933, Step 1: Divide with 255, processing time: 0.034498 seconds

Process-1933, Step 2: Resize if scale is not 1: 0.014803 seconds

Process-1933, Step 3: Get HOG channels: 5.5e-05 seconds

Process-1933, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1933, Step 5: Compute individual channel HOG features for the entire image: 1.51567 seconds

Process-1933, Step 6: Misc initializations: 1.3e-05 seconds

Process-1933, Step 7: for loop: 0.035567 seconds

Process-1933All steps time: 1.600616 seconds

Process-1933, Find cars processing time: 1.600829 seconds

Process-1934, Step 1: Divide with 255, processing time: 0.035655 seconds

Process-1934, Step 2: Resize if scale is not 1: 0.01543 seconds

Process-1934, Step 3: Get HOG channels: 5.5e-05 seconds

Process-1934, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1934, Step 5: Compute individual channel HOG features for the entire image: 1.355902 seconds

Process-1934, Step 6: Misc initializations: 8e-06 seconds

Process-1934, Step 7: for loop: 0.030216 seconds

Process-1934All steps time: 1.437275 seconds

Process-1934, Find cars processing time: 1.43751 seconds

Process-1931, Step 1: Divide with 255, processing time: 0.030073 seconds

Process-1931, Step 2: Resize if scale is not 1: 0.018266 seconds

Process-1931, Step 3: Get HOG channels: 7.7e-05 seconds

Process-1931, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1931, Step 5: Compute individual channel HOG features for the entire image: 2.118969 seconds

Process-1931, Step 6: Misc initializations: 1.2e-05 seconds

Process-1931, Step 7: for loop: 0.074011 seconds

Process-1931All steps time: 2.241418 seconds

Process-1931, Find cars processing time: 2.24181 seconds

Process-1930, Step 1: Divide with 255, processing time: 0.030602 seconds

Process-1930, Step 2: Resize if scale is not 1: 0.016565 seconds

Process-1930, Step 3: Get HOG channels: 8.1e-05 seconds

Process-1930, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1930, Step 5: Compute individual channel HOG features for the entire image: 2.526807 seconds

Process-1930, Step 6: Misc initializations: 1.9e-05 seconds

Process-1930, Step 7: for loop: 0.134547 seconds

Process-1930All steps time: 2.70863 seconds

Process-1930, Find cars processing time: 2.708909 seconds

Length of task list: 5

Number of processes used: 5

Process-1937, Step 1: Divide with 255, processing time: 0.038072 seconds

Process-1937, Step 2: Resize if scale is not 1: 0.017124 seconds

Process-1937, Step 3: Get HOG channels: 7.6e-05 seconds

Process-1937, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1937, Step 5: Compute individual channel HOG features for the entire image: 1.707877 seconds

Process-1937, Step 6: Misc initializations: 2e-06 seconds

Process-1937, Step 7: for loop: 0.056664 seconds

Process-1937All steps time: 1.8198260000000002 seconds

Process-1937, Find cars processing time: 1.820092 seconds

Process-1936, Step 1: Divide with 255, processing time: 0.030473 seconds

Process-1936, Step 2: Resize if scale is not 1: 0.017555 seconds

Process-1936, Step 3: Get HOG channels: 0.00011 seconds

Process-1936, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1936, Step 5: Compute individual channel HOG features for the entire image: 2.008114 seconds

Process-1936, Step 6: Misc initializations: 2e-06 seconds

Process-1936, Step 7: for loop: 0.077183 seconds

Process-1936All steps time: 2.1334489999999997 seconds

Process-1936, Find cars processing time: 2.133708 seconds

Process-1939, Step 1: Divide with 255, processing time: 0.029847 seconds

Process-1939, Step 2: Resize if scale is not 1: 0.014462 seconds

Process-1939, Step 3: Get HOG channels: 4.4e-05 seconds

Process-1939, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1939, Step 5: Compute individual channel HOG features for the entire image: 1.301081 seconds

Process-1939, Step 6: Misc initializations: 7e-06 seconds

Process-1939, Step 7: for loop: 0.026157 seconds

Process-1939All steps time: 1.371608 seconds

Process-1939, Find cars processing time: 1.371831 seconds

Process-1938, Step 1: Divide with 255, processing time: 0.030056 seconds

Process-1938, Step 2: Resize if scale is not 1: 0.01321 seconds

Process-1938, Step 3: Get HOG channels: 5.2e-05 seconds

Process-1938, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1938, Step 5: Compute individual channel HOG features for the entire image: 1.462827 seconds

Process-1938, Step 6: Misc initializations: 1.4e-05 seconds

Process-1938, Step 7: for loop: 0.043348 seconds

Process-1938All steps time: 1.549512 seconds

Process-1938, Find cars processing time: 1.549684 seconds

Process-1935, Step 1: Divide with 255, processing time: 0.027237 seconds

Process-1935, Step 2: Resize if scale is not 1: 0.017313 seconds

Process-1935, Step 3: Get HOG channels: 7.3e-05 seconds

Process-1935, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1935, Step 5: Compute individual channel HOG features for the entire image: 2.588391 seconds

Process-1935, Step 6: Misc initializations: 1.2e-05 seconds

Process-1935, Step 7: for loop: 0.105919 seconds

Process-1935All steps time: 2.738956 seconds

Process-1935, Find cars processing time: 2.739263 seconds

Length of task list: 5

Number of processes used: 5

Process-1942, Step 1: Divide with 255, processing time: 0.027365 seconds

Process-1942, Step 2: Resize if scale is not 1: 0.019738 seconds

Process-1942, Step 3: Get HOG channels: 7.4e-05 seconds

Process-1942, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1942, Step 5: Compute individual channel HOG features for the entire image: 1.620603 seconds

Process-1942, Step 6: Misc initializations: 2e-06 seconds

Process-1942, Step 7: for loop: 0.043544 seconds

Process-1942All steps time: 1.711338 seconds

Process-1942, Find cars processing time: 1.711614 seconds

Process-1943, Step 1: Divide with 255, processing time: 0.027768 seconds

Process-1943, Step 2: Resize if scale is not 1: 0.015105 seconds

Process-1943, Step 3: Get HOG channels: 5.3e-05 seconds

Process-1943, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1943, Step 5: Compute individual channel HOG features for the entire image: 1.3802 seconds

Process-1943, Step 6: Misc initializations: 1.5e-05 seconds

Process-1941, Step 1: Divide with 255, processing time: 0.027647 seconds

Process-1941, Step 2: Resize if scale is not 1: 0.018436 seconds

Process-1941, Step 3: Get HOG channels: 6.9e-05 seconds

Process-1943, Step 7: for loop: 0.033787 seconds

Process-1941, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1941, Step 5: Compute individual channel HOG features for the entire image: 2.080707 seconds

Process-1943All steps time: 1.4569390000000002 seconds

Process-1941, Step 6: Misc initializations: 8e-06 seconds

```
Process-1943, Find cars processing time: 1.457173 seconds
Process-1941, Step 7: for loop: 0.081739 seconds
```

```
Process-1941All steps time: 2.2086159999999992 seconds
```

```
Process-1941, Find cars processing time: 2.208883 seconds
```

```
Process-1940, Step 1: Divide with 255, processing time: 0.030359 seconds
Process-1940, Step 2: Resize if scale is not 1: 0.01505 seconds
Process-1940, Step 3: Get HOG channels: 7.1e-05 seconds
Process-1940, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1940, Step 5: Compute individual channel HOG features for the entire image: 2.674135 seconds
Process-1940, Step 6: Misc initializations: 3e-06 seconds
Process-1940, Step 7: for loop: 0.105226 seconds
```

```
Process-1940All steps time: 2.8248550000000003 seconds
```

```
Process-1940, Find cars processing time: 2.825115 seconds
```

```
Process-1944, Step 1: Divide with 255, processing time: 0.030257 seconds
Process-1944, Step 2: Resize if scale is not 1: 0.015754 seconds
Process-1944, Step 3: Get HOG channels: 4.6e-05 seconds
Process-1944, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-1944, Step 5: Compute individual channel HOG features for the entire image: 1.381032 seconds
Process-1944, Step 6: Misc initializations: 1e-05 seconds
Process-1944, Step 7: for loop: 0.038781 seconds
```

```
Process-1944All steps time: 1.4658930000000001 seconds
```

```
Process-1944, Find cars processing time: 1.466091 seconds
```

```
Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]
```

```
5 processes used for testing 5 window sizes
```

```
Processing times for each image [3.7824, 3.5935, 3.4642, 4.1269, 3.5075, 3.679] with an average
```

```
Time elapsed so far... 296.11670000000004
```

```
In [118]: for i in range(len(scaleslist)):
    print(scaleslist[i])
```

```
[0.95]
[1.15, 1.35, 1.55]
[0.75, 0.95, 1.15, 1.35, 1.55]
```

```
[1.15, 1.25, 1.35, 1.45, 1.55]
[0.75, 0.8, 0.85, 0.9, 0.95, 1.05]
[0.75, 0.85, 0.95, 1.05, 1.15, 1.25, 1.35, 1.45, 1.55]
[0.75, 0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.45, 1.5, 1.55, 1.6]
```

```
In [119]: use_sub_windows = 1
use_default_number_of_processes = 0
print_find_car_step_times = 1
find_car_hog_parallelism_enabled = 0
print_task_times = 1

time_elapsed = 0
for i in range(4):
    for j in range(1, len(scaleslist[i])+1):
        time_for_all_images = []
        test_images_with_boxes = []
        number_of_processes_used = j
        for image in test_images:
            print('Number of processes used: ', number_of_processes_used, 'window size')
            img = mpimg.imread(image)
            freeze_support()
            t0 = time.time()
            r = []
            r = parallelize_find_cars(img, number_of_processes_used, i)
            t1 = time.time()
            processing_time = round(t1-t0, 4)
            #print('Time taken for processing each image', processing_time)
            time_for_all_images.append(processing_time)
            img_with_boxes = np.copy(img)
            for box in r:
                cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
            #plt.figure(figsize=(16,9))
            #plt.imshow(img_with_boxes)
            test_images_with_boxes.append(img_with_boxes)
            print('*****')
        print('Window sizes used:', scaleslist[i])
        print(number_of_processes_used, 'processes used for testing', len(scaleslist[i]))
        print('Processing times for each image', time_for_all_images, \
              'with an average of ', round(sum(time_for_all_images)/len(time_for_all_images)))
        print()
        time_elapsed += sum(time_for_all_images)
        print('Time elapsed so far...', time_elapsed)
        print('#####')

        print()
print('=====')
```

```
print()
print()

Number of processes used: 1 window size 260
Length of task list: 4
Number of processes used: 1

Process-1945, Step 1: Divide with 255, processing time: 0.027743 seconds
Process-1945, Step 2: Resize if scale is not 1: 0.002062 seconds
Process-1945, Step 3: Get HOG channels: 2.4e-05 seconds
Process-1945, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1945, Step 5: Compute individual channel HOG features for the entire image: 0.270441 seconds
Process-1945, Step 6: Misc initializations: 0.0 seconds
Process-1945, Step 7: for loop: 0.018548 seconds

Process-1945All steps time: 0.318824 seconds

Process-1945, Find cars processing time: 0.318956 seconds

Process-1945, Step 1: Divide with 255, processing time: 0.010225 seconds
Process-1945, Step 2: Resize if scale is not 1: 0.001161 seconds
Process-1945, Step 3: Get HOG channels: 1e-05 seconds
Process-1945, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1945, Step 5: Compute individual channel HOG features for the entire image: 0.25625 seconds
Process-1945, Step 6: Misc initializations: 0.0 seconds
Process-1945, Step 7: for loop: 0.017654 seconds

Process-1945All steps time: 0.285309 seconds

Process-1945, Find cars processing time: 0.28538 seconds

Process-1945, Step 1: Divide with 255, processing time: 0.011128 seconds
Process-1945, Step 2: Resize if scale is not 1: 0.001141 seconds
Process-1945, Step 3: Get HOG channels: 9e-06 seconds
Process-1945, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1945, Step 5: Compute individual channel HOG features for the entire image: 0.265547 seconds
Process-1945, Step 6: Misc initializations: 1e-06 seconds
Process-1945, Step 7: for loop: 0.017865 seconds

Process-1945All steps time: 0.29569999999999996 seconds

Process-1945, Find cars processing time: 0.295793 seconds

Process-1945, Step 1: Divide with 255, processing time: 0.010604 seconds
Process-1945, Step 2: Resize if scale is not 1: 0.000748 seconds
Process-1945, Step 3: Get HOG channels: 6e-06 seconds
Process-1945, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1945, Step 5: Compute individual channel HOG features for the entire image: 0.264124 seconds
```

Process-1945, Step 6: Misc initializations: 1e-06 seconds
Process-1945, Step 7: for loop: 0.020155 seconds

Process-1945All steps time: 0.295645 seconds

Process-1945, Find cars processing time: 0.295718 seconds

The times for each task are: [0.318956, 0.28538, 0.295793, 0.295718] with:

Minimum: 0.28538 Maximum: 0.318956 Average: 0.299 seconds

Number of processes used: 1 window size 260
Length of task list: 4
Number of processes used: 1

Process-1946, Step 1: Divide with 255, processing time: 0.034114 seconds

Process-1946, Step 2: Resize if scale is not 1: 0.002567 seconds

Process-1946, Step 3: Get HOG channels: 3.4e-05 seconds

Process-1946, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1946, Step 5: Compute individual channel HOG features for the entire image: 0.314884 seconds

Process-1946, Step 6: Misc initializations: 1e-06 seconds

Process-1946, Step 7: for loop: 0.019967 seconds

Process-1946All steps time: 0.3715739999999996 seconds

Process-1946, Find cars processing time: 0.371761 seconds

Process-1946, Step 1: Divide with 255, processing time: 0.011055 seconds

Process-1946, Step 2: Resize if scale is not 1: 0.001311 seconds

Process-1946, Step 3: Get HOG channels: 1.1e-05 seconds

Process-1946, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1946, Step 5: Compute individual channel HOG features for the entire image: 0.27491 seconds

Process-1946, Step 6: Misc initializations: 1e-06 seconds

Process-1946, Step 7: for loop: 0.018793 seconds

Process-1946All steps time: 0.3060909999999995 seconds

Process-1946, Find cars processing time: 0.306186 seconds

Process-1946, Step 1: Divide with 255, processing time: 0.010381 seconds

Process-1946, Step 2: Resize if scale is not 1: 0.000839 seconds

Process-1946, Step 3: Get HOG channels: 7e-06 seconds

Process-1946, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1946, Step 5: Compute individual channel HOG features for the entire image: 0.34789 seconds

Process-1946, Step 6: Misc initializations: 1e-06 seconds

Process-1946, Step 7: for loop: 0.020994 seconds

Process-1946All steps time: 0.380119 seconds

Process-1946, Find cars processing time: 0.380202 seconds

Process-1946, Step 1: Divide with 255, processing time: 0.010294 seconds

Process-1946, Step 2: Resize if scale is not 1: 0.000671 seconds

Process-1946, Step 3: Get HOG channels: 6e-06 seconds

Process-1946, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1946, Step 5: Compute individual channel HOG features for the entire image: 0.322717 seconds

Process-1946, Step 6: Misc initializations: 0.0 seconds

Process-1946, Step 7: for loop: 0.018365 seconds

Process-1946All steps time: 0.35206 seconds

Process-1946, Find cars processing time: 0.352132 seconds

The times for each task are: [0.371761, 0.306186, 0.380202, 0.352132] with:

Minimum: 0.306186 Maximum: 0.380202 Average: 0.3526 seconds

Number of processes used: 1 window size 260

Length of task list: 4

Number of processes used: 1

Process-1947, Step 1: Divide with 255, processing time: 0.024926 seconds

Process-1947, Step 2: Resize if scale is not 1: 0.002215 seconds

Process-1947, Step 3: Get HOG channels: 2.9e-05 seconds

Process-1947, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1947, Step 5: Compute individual channel HOG features for the entire image: 0.301354 seconds

Process-1947, Step 6: Misc initializations: 1e-06 seconds

Process-1947, Step 7: for loop: 0.019766 seconds

Process-1947All steps time: 0.3482969999999997 seconds

Process-1947, Find cars processing time: 0.348417 seconds

Process-1947, Step 1: Divide with 255, processing time: 0.010957 seconds

Process-1947, Step 2: Resize if scale is not 1: 0.000703 seconds

Process-1947, Step 3: Get HOG channels: 6e-06 seconds

Process-1947, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1947, Step 5: Compute individual channel HOG features for the entire image: 0.273445 seconds

Process-1947, Step 6: Misc initializations: 1e-06 seconds

Process-1947, Step 7: for loop: 0.018768 seconds

Process-1947All steps time: 0.303886 seconds

Process-1947, Find cars processing time: 0.303959 seconds

```
Process-1947, Step 1: Divide with 255, processing time: 0.010812 seconds
Process-1947, Step 2: Resize if scale is not 1: 0.000825 seconds
Process-1947, Step 3: Get HOG channels: 6e-06 seconds
Process-1947, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1947, Step 5: Compute individual channel HOG features for the entire image: 0.301176 seconds
Process-1947, Step 6: Misc initializations: 1e-06 seconds
Process-1947, Step 7: for loop: 0.01958 seconds
```

Process-1947All steps time: 0.33240699999999995 seconds

Process-1947, Find cars processing time: 0.332492 seconds

```
Process-1947, Step 1: Divide with 255, processing time: 0.009576 seconds
Process-1947, Step 2: Resize if scale is not 1: 0.000957 seconds
Process-1947, Step 3: Get HOG channels: 8e-06 seconds
Process-1947, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1947, Step 5: Compute individual channel HOG features for the entire image: 0.337093 seconds
Process-1947, Step 6: Misc initializations: 1e-06 seconds
Process-1947, Step 7: for loop: 0.01888 seconds
```

Process-1947All steps time: 0.36652299999999993 seconds

Process-1947, Find cars processing time: 0.366593 seconds

The times for each task are: [0.348417, 0.303959, 0.332492, 0.366593] with:

Minimum: 0.303959 Maximum: 0.366593 Average: 0.3379 seconds

```
*****
Number of processes used: 1 window size 260
Length of task list: 4
Number of processes used: 1
```

```
Process-1948, Step 1: Divide with 255, processing time: 0.028584 seconds
Process-1948, Step 2: Resize if scale is not 1: 0.002339 seconds
Process-1948, Step 3: Get HOG channels: 3.2e-05 seconds
Process-1948, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1948, Step 5: Compute individual channel HOG features for the entire image: 0.366949 seconds
Process-1948, Step 6: Misc initializations: 0.0 seconds
Process-1948, Step 7: for loop: 0.019879 seconds
```

Process-1948All steps time: 0.41779 seconds

Process-1948, Find cars processing time: 0.417926 seconds

```
Process-1948, Step 1: Divide with 255, processing time: 0.010352 seconds
Process-1948, Step 2: Resize if scale is not 1: 0.000962 seconds
```

Process-1948, Step 3: Get HOG channels: 6e-06 seconds
Process-1948, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1948, Step 5: Compute individual channel HOG features for the entire image: 0.35077 seconds
Process-1948, Step 6: Misc initializations: 1e-06 seconds
Process-1948, Step 7: for loop: 0.021889 seconds

Process-1948All steps time: 0.383986 seconds

Process-1948, Find cars processing time: 0.384074 seconds

Process-1948, Step 1: Divide with 255, processing time: 0.011067 seconds
Process-1948, Step 2: Resize if scale is not 1: 0.001142 seconds
Process-1948, Step 3: Get HOG channels: 9e-06 seconds
Process-1948, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1948, Step 5: Compute individual channel HOG features for the entire image: 0.27567 seconds
Process-1948, Step 6: Misc initializations: 1e-06 seconds
Process-1948, Step 7: for loop: 0.019124 seconds

Process-1948All steps time: 0.307023 seconds

Process-1948, Find cars processing time: 0.307123 seconds

Process-1948, Step 1: Divide with 255, processing time: 0.010848 seconds
Process-1948, Step 2: Resize if scale is not 1: 0.001221 seconds
Process-1948, Step 3: Get HOG channels: 1e-05 seconds
Process-1948, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1948, Step 5: Compute individual channel HOG features for the entire image: 0.254375 seconds
Process-1948, Step 6: Misc initializations: 0.0 seconds
Process-1948, Step 7: for loop: 0.017851 seconds

Process-1948All steps time: 0.284316 seconds

Process-1948, Find cars processing time: 0.284399 seconds

The times for each task are: [0.417926, 0.384074, 0.307123, 0.284399] with:

Minimum: 0.284399 Maximum: 0.417926 Average: 0.3484 seconds

Number of processes used: 1 window size 260
Length of task list: 4
Number of processes used: 1

Process-1949, Step 1: Divide with 255, processing time: 0.030277 seconds
Process-1949, Step 2: Resize if scale is not 1: 0.002534 seconds
Process-1949, Step 3: Get HOG channels: 2.8e-05 seconds
Process-1949, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1949, Step 5: Compute individual channel HOG features for the entire image: 0.265566 seconds

Process-1949, Step 6: Misc initializations: 0.0 seconds
Process-1949, Step 7: for loop: 0.019686 seconds

Process-1949All steps time: 0.318097 seconds

Process-1949, Find cars processing time: 0.318259 seconds

Process-1949, Step 1: Divide with 255, processing time: 0.010897 seconds
Process-1949, Step 2: Resize if scale is not 1: 0.000878 seconds
Process-1949, Step 3: Get HOG channels: 9e-06 seconds
Process-1949, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1949, Step 5: Compute individual channel HOG features for the entire image: 0.310696 seconds
Process-1949, Step 6: Misc initializations: 1e-06 seconds
Process-1949, Step 7: for loop: 0.018779 seconds

Process-1949All steps time: 0.34127 seconds

Process-1949, Find cars processing time: 0.341347 seconds

Process-1949, Step 1: Divide with 255, processing time: 0.010428 seconds
Process-1949, Step 2: Resize if scale is not 1: 0.00087 seconds
Process-1949, Step 3: Get HOG channels: 6e-06 seconds
Process-1949, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1949, Step 5: Compute individual channel HOG features for the entire image: 0.337389 seconds
Process-1949, Step 6: Misc initializations: 1e-06 seconds
Process-1949, Step 7: for loop: 0.018591 seconds

Process-1949All steps time: 0.367292 seconds

Process-1949, Find cars processing time: 0.367367 seconds

Process-1949, Step 1: Divide with 255, processing time: 0.010348 seconds
Process-1949, Step 2: Resize if scale is not 1: 0.001015 seconds
Process-1949, Step 3: Get HOG channels: 9e-06 seconds
Process-1949, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1949, Step 5: Compute individual channel HOG features for the entire image: 0.255286 seconds
Process-1949, Step 6: Misc initializations: 0.0 seconds
Process-1949, Step 7: for loop: 0.017594 seconds

Process-1949All steps time: 0.284261 seconds

Process-1949, Find cars processing time: 0.284324 seconds

The times for each task are: [0.318259, 0.341347, 0.367367, 0.284324] with:

Minimum: 0.284324 Maximum: 0.367367 Average: 0.3278 seconds

Number of processes used: 1 window size 260
Length of task list: 4
Number of processes used: 1

Process-1950, Step 1: Divide with 255, processing time: 0.030804 seconds
Process-1950, Step 2: Resize if scale is not 1: 0.002388 seconds
Process-1950, Step 3: Get HOG channels: 3e-05 seconds
Process-1950, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1950, Step 5: Compute individual channel HOG features for the entire image: 0.354508 seconds
Process-1950, Step 6: Misc initializations: 0.0 seconds
Process-1950, Step 7: for loop: 0.019764 seconds

Process-1950All steps time: 0.407501 seconds

Process-1950, Find cars processing time: 0.407654 seconds

Process-1950, Step 1: Divide with 255, processing time: 0.010841 seconds
Process-1950, Step 2: Resize if scale is not 1: 0.001069 seconds
Process-1950, Step 3: Get HOG channels: 7e-06 seconds
Process-1950, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1950, Step 5: Compute individual channel HOG features for the entire image: 0.301195 seconds
Process-1950, Step 6: Misc initializations: 0.0 seconds
Process-1950, Step 7: for loop: 0.018643 seconds

Process-1950All steps time: 0.331762 seconds

Process-1950, Find cars processing time: 0.331851 seconds

Process-1950, Step 1: Divide with 255, processing time: 0.007691 seconds
Process-1950, Step 2: Resize if scale is not 1: 0.000889 seconds
Process-1950, Step 3: Get HOG channels: 8e-06 seconds
Process-1950, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1950, Step 5: Compute individual channel HOG features for the entire image: 0.319133 seconds
Process-1950, Step 6: Misc initializations: 0.0 seconds
Process-1950, Step 7: for loop: 0.019562 seconds

Process-1950All steps time: 0.347291 seconds

Process-1950, Find cars processing time: 0.347406 seconds

Process-1950, Step 1: Divide with 255, processing time: 0.010208 seconds
Process-1950, Step 2: Resize if scale is not 1: 0.001005 seconds
Process-1950, Step 3: Get HOG channels: 5e-06 seconds
Process-1950, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1950, Step 5: Compute individual channel HOG features for the entire image: 0.321721 seconds
Process-1950, Step 6: Misc initializations: 1e-06 seconds
Process-1950, Step 7: for loop: 0.019574 seconds

Process-1950All steps time: 0.3525199999999994 seconds

Process-1950, Find cars processing time: 0.352602 seconds

The times for each task are: [0.407654, 0.331851, 0.347406, 0.352602] with:

Minimum: 0.331851 Maximum: 0.407654 Average: 0.3599 seconds

Window sizes used: [0.95]

1 processes used for testing 1 window sizes

Processing times for each image [2.5874, 3.1292, 2.4443, 2.6289, 2.6061, 2.9521] with an average

Time elapsed so far... 16.348

#####

=====

Number of processes used: 1 window size 260

Length of task list: 13

Number of processes used: 1

Process-1951, Step 1: Divide with 255, processing time: 0.029157 seconds

Process-1951, Step 2: Resize if scale is not 1: 0.002048 seconds

Process-1951, Step 3: Get HOG channels: 2.3e-05 seconds

Process-1951, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.243011 seconds

Process-1951, Step 6: Misc initializations: 1e-06 seconds

Process-1951, Step 7: for loop: 0.016845 seconds

Process-1951All steps time: 0.2910899999999996 seconds

Process-1951, Find cars processing time: 0.291226 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.009746 seconds

Process-1951, Step 2: Resize if scale is not 1: 0.000867 seconds

Process-1951, Step 3: Get HOG channels: 8e-06 seconds

Process-1951, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.185163 seconds

Process-1951, Step 6: Misc initializations: 0.0 seconds

Process-1951, Step 7: for loop: 0.012066 seconds

Process-1951All steps time: 0.207858 seconds

Process-1951, Find cars processing time: 0.207937 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.010739 seconds

Process-1951, Step 2: Resize if scale is not 1: 0.000777 seconds
Process-1951, Step 3: Get HOG channels: 7e-06 seconds
Process-1951, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.234046 seconds
Process-1951, Step 6: Misc initializations: 0.0 seconds
Process-1951, Step 7: for loop: 0.012288 seconds

Process-1951All steps time: 0.257865 seconds

Process-1951, Find cars processing time: 0.257943 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.010887 seconds
Process-1951, Step 2: Resize if scale is not 1: 0.000885 seconds
Process-1951, Step 3: Get HOG channels: 9e-06 seconds
Process-1951, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.182706 seconds
Process-1951, Step 6: Misc initializations: 0.0 seconds
Process-1951, Step 7: for loop: 0.012018 seconds

Process-1951All steps time: 0.206513 seconds

Process-1951, Find cars processing time: 0.206608 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.010804 seconds
Process-1951, Step 2: Resize if scale is not 1: 0.001067 seconds
Process-1951, Step 3: Get HOG channels: 7e-06 seconds
Process-1951, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.136169 seconds
Process-1951, Step 6: Misc initializations: 0.0 seconds
Process-1951, Step 7: for loop: 0.007505 seconds

Process-1951All steps time: 0.15555900000000003 seconds

Process-1951, Find cars processing time: 0.155637 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.010786 seconds
Process-1951, Step 2: Resize if scale is not 1: 0.000906 seconds
Process-1951, Step 3: Get HOG channels: 9e-06 seconds
Process-1951, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.14421 seconds
Process-1951, Step 6: Misc initializations: 1e-06 seconds
Process-1951, Step 7: for loop: 0.00772 seconds

Process-1951All steps time: 0.163642 seconds

Process-1951, Find cars processing time: 0.16372 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.009758 seconds

Process-1951, Step 2: Resize if scale is not 1: 0.000607 seconds
Process-1951, Step 3: Get HOG channels: 7e-06 seconds
Process-1951, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.13865 seconds
Process-1951, Step 6: Misc initializations: 0.0 seconds
Process-1951, Step 7: for loop: 0.007377 seconds

Process-1951All steps time: 0.156407 seconds

Process-1951, Find cars processing time: 0.156475 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.011755 seconds
Process-1951, Step 2: Resize if scale is not 1: 0.000871 seconds
Process-1951, Step 3: Get HOG channels: 8e-06 seconds
Process-1951, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.135156 seconds
Process-1951, Step 6: Misc initializations: 0.0 seconds
Process-1951, Step 7: for loop: 0.00757 seconds

Process-1951All steps time: 0.15536899999999998 seconds

Process-1951, Find cars processing time: 0.155452 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.011063 seconds
Process-1951, Step 2: Resize if scale is not 1: 0.001055 seconds
Process-1951, Step 3: Get HOG channels: 6e-06 seconds
Process-1951, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.108069 seconds
Process-1951, Step 6: Misc initializations: 3e-06 seconds
Process-1951, Step 7: for loop: 0.00601 seconds

Process-1951All steps time: 0.126212 seconds

Process-1951, Find cars processing time: 0.126313 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.007164 seconds
Process-1951, Step 2: Resize if scale is not 1: 0.000652 seconds
Process-1951, Step 3: Get HOG channels: 1e-05 seconds
Process-1951, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.101087 seconds
Process-1951, Step 6: Misc initializations: 0.0 seconds
Process-1951, Step 7: for loop: 0.00588 seconds

Process-1951All steps time: 0.1147999999999999 seconds

Process-1951, Find cars processing time: 0.114882 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.010532 seconds

```
Process-1951, Step 2: Resize if scale is not 1: 0.000841 seconds
Process-1951, Step 3: Get HOG channels: 1e-05 seconds
Process-1951, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.126419 seconds
Process-1951, Step 6: Misc initializations: 1e-06 seconds
Process-1951, Step 7: for loop: 0.007089 seconds

Process-1951All steps time: 0.14490100000000003 seconds

Process-1951, Find cars processing time: 0.144969 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.010532 seconds
Process-1951, Step 2: Resize if scale is not 1: 0.000517 seconds
Process-1951, Step 3: Get HOG channels: 6e-06 seconds
Process-1951, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.129424 seconds
Process-1951, Step 6: Misc initializations: 0.0 seconds
Process-1951, Step 7: for loop: 0.007506 seconds

Process-1951All steps time: 0.147992 seconds

Process-1951, Find cars processing time: 0.148059 seconds

Process-1951, Step 1: Divide with 255, processing time: 0.014794 seconds
Process-1951, Step 2: Resize if scale is not 1: 0.001379 seconds
Process-1951, Step 3: Get HOG channels: 7e-06 seconds
Process-1951, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1951, Step 5: Compute individual channel HOG features for the entire image: 0.116642 seconds
Process-1951, Step 6: Misc initializations: 0.0 seconds
Process-1951, Step 7: for loop: 0.00652 seconds

Process-1951All steps time: 0.139349 seconds

Process-1951, Find cars processing time: 0.139424 seconds

The times for each task are: [0.291226, 0.207937, 0.257943, 0.206608, 0.155637, 0.16372, 0.15647

Minimum: 0.114882 Maximum: 0.291226 Average: 0.1745 seconds

*****
Number of processes used: 1 window size 260
Length of task list: 13
Number of processes used: 1

Process-1952, Step 1: Divide with 255, processing time: 0.022827 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.00172 seconds
Process-1952, Step 3: Get HOG channels: 2e-05 seconds
Process-1952, Step 4: Define blocks and steps as above: 6e-06 seconds
```

Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.182836 seconds
Process-1952, Step 6: Misc initializations: 0.0 seconds
Process-1952, Step 7: for loop: 0.013604 seconds

Process-1952All steps time: 0.22101300000000001 seconds

Process-1952, Find cars processing time: 0.221138 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.010797 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.00098 seconds
Process-1952, Step 3: Get HOG channels: 9e-06 seconds
Process-1952, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.229016 seconds
Process-1952, Step 6: Misc initializations: 1e-06 seconds
Process-1952, Step 7: for loop: 0.012204 seconds

Process-1952All steps time: 0.253016 seconds

Process-1952, Find cars processing time: 0.253092 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.00995 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.000938 seconds
Process-1952, Step 3: Get HOG channels: 6e-06 seconds
Process-1952, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.192756 seconds
Process-1952, Step 6: Misc initializations: 5e-06 seconds
Process-1952, Step 7: for loop: 0.01299 seconds

Process-1952All steps time: 0.216652 seconds

Process-1952, Find cars processing time: 0.216754 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.009953 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.000981 seconds
Process-1952, Step 3: Get HOG channels: 1.3e-05 seconds
Process-1952, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.184649 seconds
Process-1952, Step 6: Misc initializations: 1e-06 seconds
Process-1952, Step 7: for loop: 0.012323 seconds

Process-1952All steps time: 0.207929 seconds

Process-1952, Find cars processing time: 0.208008 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.018242 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.000706 seconds
Process-1952, Step 3: Get HOG channels: 7e-06 seconds
Process-1952, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.177121 seconds
Process-1952, Step 6: Misc initializations: 0.0 seconds
Process-1952, Step 7: for loop: 0.009255 seconds

Process-1952All steps time: 0.20533800000000002 seconds

Process-1952, Find cars processing time: 0.205439 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.013891 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.000409 seconds
Process-1952, Step 3: Get HOG channels: 5e-06 seconds
Process-1952, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.127116 seconds
Process-1952, Step 6: Misc initializations: 0.0 seconds
Process-1952, Step 7: for loop: 0.007213 seconds

Process-1952All steps time: 0.14864 seconds

Process-1952, Find cars processing time: 0.148687 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.009968 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.000785 seconds
Process-1952, Step 3: Get HOG channels: 8e-06 seconds
Process-1952, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.131065 seconds
Process-1952, Step 6: Misc initializations: 0.0 seconds
Process-1952, Step 7: for loop: 0.007287 seconds

Process-1952All steps time: 0.1491219999999998 seconds

Process-1952, Find cars processing time: 0.149191 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.0109 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.000934 seconds
Process-1952, Step 3: Get HOG channels: 1.4e-05 seconds
Process-1952, Step 4: Define blocks and steps as above: 1.4e-05 seconds
Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.139967 seconds
Process-1952, Step 6: Misc initializations: 0.0 seconds
Process-1952, Step 7: for loop: 0.007492 seconds

Process-1952All steps time: 0.15932100000000002 seconds

Process-1952, Find cars processing time: 0.159413 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.010882 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.001457 seconds
Process-1952, Step 3: Get HOG channels: 1e-05 seconds
Process-1952, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.132666 seconds
Process-1952, Step 6: Misc initializations: 1e-06 seconds
Process-1952, Step 7: for loop: 0.007197 seconds

Process-1952All steps time: 0.15222200000000002 seconds

Process-1952, Find cars processing time: 0.152302 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.010444 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.000782 seconds
Process-1952, Step 3: Get HOG channels: 8e-06 seconds
Process-1952, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.128709 seconds
Process-1952, Step 6: Misc initializations: 0.0 seconds
Process-1952, Step 7: for loop: 0.007116 seconds

Process-1952All steps time: 0.147068 seconds

Process-1952, Find cars processing time: 0.14714 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.010256 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.000603 seconds
Process-1952, Step 3: Get HOG channels: 7e-06 seconds
Process-1952, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.138706 seconds
Process-1952, Step 6: Misc initializations: 0.0 seconds
Process-1952, Step 7: for loop: 0.007279 seconds

Process-1952All steps time: 0.156859 seconds

Process-1952, Find cars processing time: 0.156936 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.010853 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.000885 seconds
Process-1952, Step 3: Get HOG channels: 1e-05 seconds
Process-1952, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.128963 seconds
Process-1952, Step 6: Misc initializations: 0.0 seconds
Process-1952, Step 7: for loop: 0.007118 seconds

Process-1952All steps time: 0.147839 seconds

Process-1952, Find cars processing time: 0.14792 seconds

Process-1952, Step 1: Divide with 255, processing time: 0.009565 seconds
Process-1952, Step 2: Resize if scale is not 1: 0.001478 seconds
Process-1952, Step 3: Get HOG channels: 6e-06 seconds
Process-1952, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1952, Step 5: Compute individual channel HOG features for the entire image: 0.120316 seconds
Process-1952, Step 6: Misc initializations: 2e-06 seconds
Process-1952, Step 7: for loop: 0.006243 seconds

Process-1952All steps time: 0.13761600000000002 seconds

Process-1952, Find cars processing time: 0.137685 seconds

The times for each task are: [0.221138, 0.253092, 0.216754, 0.208008, 0.205439, 0.148687, 0.1491

Minimum: 0.137685 Maximum: 0.253092 Average: 0.1772 seconds

Number of processes used: 1 window size 260

Length of task list: 13

Number of processes used: 1

Process-1953, Step 1: Divide with 255, processing time: 0.033826 seconds
Process-1953, Step 2: Resize if scale is not 1: 0.002384 seconds
Process-1953, Step 3: Get HOG channels: 2.9e-05 seconds
Process-1953, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.246453 seconds
Process-1953, Step 6: Misc initializations: 1e-06 seconds
Process-1953, Step 7: for loop: 0.014285 seconds

Process-1953All steps time: 0.296985 seconds

Process-1953, Find cars processing time: 0.29716 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.010234 seconds
Process-1953, Step 2: Resize if scale is not 1: 0.000715 seconds
Process-1953, Step 3: Get HOG channels: 7e-06 seconds
Process-1953, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.239601 seconds
Process-1953, Step 6: Misc initializations: 1e-06 seconds
Process-1953, Step 7: for loop: 0.013685 seconds

Process-1953All steps time: 0.264252 seconds

Process-1953, Find cars processing time: 0.264323 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.010357 seconds
Process-1953, Step 2: Resize if scale is not 1: 0.000594 seconds
Process-1953, Step 3: Get HOG channels: 6e-06 seconds
Process-1953, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.218191 seconds
Process-1953, Step 6: Misc initializations: 0.0 seconds
Process-1953, Step 7: for loop: 0.01187 seconds

Process-1953All steps time: 0.241025 seconds

Process-1953, Find cars processing time: 0.241082 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.005689 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.000563 seconds

Process-1953, Step 3: Get HOG channels: 5e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.196328 seconds

Process-1953, Step 6: Misc initializations: 0.0 seconds

Process-1953, Step 7: for loop: 0.012533 seconds

Process-1953All steps time: 0.2151239999999998 seconds

Process-1953, Find cars processing time: 0.2152 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.012586 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.000712 seconds

Process-1953, Step 3: Get HOG channels: 6e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.172418 seconds

Process-1953, Step 6: Misc initializations: 0.0 seconds

Process-1953, Step 7: for loop: 0.00953 seconds

Process-1953All steps time: 0.195259 seconds

Process-1953, Find cars processing time: 0.195338 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.009296 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.000536 seconds

Process-1953, Step 3: Get HOG channels: 6e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.166147 seconds

Process-1953, Step 6: Misc initializations: 0.0 seconds

Process-1953, Step 7: for loop: 0.007896 seconds

Process-1953All steps time: 0.1838869999999997 seconds

Process-1953, Find cars processing time: 0.183941 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.008057 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.000681 seconds

Process-1953, Step 3: Get HOG channels: 8e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.173378 seconds

Process-1953, Step 6: Misc initializations: 0.0 seconds

Process-1953, Step 7: for loop: 0.010176 seconds

Process-1953All steps time: 0.1923089999999998 seconds

Process-1953, Find cars processing time: 0.192383 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.018632 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.000764 seconds

Process-1953, Step 3: Get HOG channels: 7e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.174151 seconds

Process-1953, Step 6: Misc initializations: 0.0 seconds

Process-1953, Step 7: for loop: 0.009522 seconds

Process-1953All steps time: 0.203082 seconds

Process-1953, Find cars processing time: 0.203157 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.01038 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.001277 seconds

Process-1953, Step 3: Get HOG channels: 6e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.138767 seconds

Process-1953, Step 6: Misc initializations: 0.0 seconds

Process-1953, Step 7: for loop: 0.007196 seconds

Process-1953All steps time: 0.15763300000000002 seconds

Process-1953, Find cars processing time: 0.157709 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.01064 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.000635 seconds

Process-1953, Step 3: Get HOG channels: 7e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.128489 seconds

Process-1953, Step 6: Misc initializations: 0.0 seconds

Process-1953, Step 7: for loop: 0.006361 seconds

Process-1953All steps time: 0.146139 seconds

Process-1953, Find cars processing time: 0.146208 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.013083 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.000581 seconds

Process-1953, Step 3: Get HOG channels: 6e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.115858 seconds

Process-1953, Step 6: Misc initializations: 1e-06 seconds

Process-1953, Step 7: for loop: 0.006646 seconds

Process-1953All steps time: 0.13618100000000002 seconds

Process-1953, Find cars processing time: 0.136273 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.011156 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.000523 seconds

Process-1953, Step 3: Get HOG channels: 6e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.139231 seconds

Process-1953, Step 6: Misc initializations: 1e-06 seconds

Process-1953, Step 7: for loop: 0.007364 seconds

Process-1953All steps time: 0.158287 seconds

Process-1953, Find cars processing time: 0.158385 seconds

Process-1953, Step 1: Divide with 255, processing time: 0.010113 seconds

Process-1953, Step 2: Resize if scale is not 1: 0.001042 seconds

Process-1953, Step 3: Get HOG channels: 5e-06 seconds

Process-1953, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1953, Step 5: Compute individual channel HOG features for the entire image: 0.085668 seconds

Process-1953, Step 6: Misc initializations: 0.0 seconds

Process-1953, Step 7: for loop: 0.004665 seconds

Process-1953All steps time: 0.1014979999999999 seconds

Process-1953, Find cars processing time: 0.101541 seconds

The times for each task are: [0.29716, 0.264323, 0.241082, 0.2152, 0.195338, 0.183941, 0.192383,

Minimum: 0.101541 Maximum: 0.29716 Average: 0.1917 seconds

Number of processes used: 1 window size 260

Length of task list: 13

Number of processes used: 1

Process-1954, Step 1: Divide with 255, processing time: 0.028633 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.002985 seconds

Process-1954, Step 3: Get HOG channels: 4.4e-05 seconds

Process-1954, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.231011 seconds

Process-1954, Step 6: Misc initializations: 0.0 seconds

Process-1954, Step 7: for loop: 0.013168 seconds

Process-1954All steps time: 0.275851 seconds

Process-1954, Find cars processing time: 0.276017 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.008296 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.000735 seconds

Process-1954, Step 3: Get HOG channels: 7e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.233846 seconds

Process-1954, Step 6: Misc initializations: 1e-06 seconds

Process-1954, Step 7: for loop: 0.015779 seconds

Process-1954All steps time: 0.258672 seconds

Process-1954, Find cars processing time: 0.258736 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.010229 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.000907 seconds

Process-1954, Step 3: Get HOG channels: 1.2e-05 seconds

Process-1954, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.220332 seconds

Process-1954, Step 6: Misc initializations: 0.0 seconds

Process-1954, Step 7: for loop: 0.01184 seconds

Process-1954All steps time: 0.24333 seconds

Process-1954, Find cars processing time: 0.243402 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.011204 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.000873 seconds

Process-1954, Step 3: Get HOG channels: 9e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.235938 seconds

Process-1954, Step 6: Misc initializations: 1e-06 seconds

Process-1954, Step 7: for loop: 0.014317 seconds

Process-1954All steps time: 0.26235 seconds

Process-1954, Find cars processing time: 0.262426 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.014932 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.000622 seconds

Process-1954, Step 3: Get HOG channels: 6e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.133732 seconds

Process-1954, Step 6: Misc initializations: 0.0 seconds

Process-1954, Step 7: for loop: 0.008312 seconds

Process-1954All steps time: 0.1576099999999997 seconds

Process-1954, Find cars processing time: 0.157675 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.014857 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.000615 seconds

Process-1954, Step 3: Get HOG channels: 7e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.161676 seconds

Process-1954, Step 6: Misc initializations: 0.0 seconds

Process-1954, Step 7: for loop: 0.007691 seconds

Process-1954All steps time: 0.184852 seconds

Process-1954, Find cars processing time: 0.184936 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.010725 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.000529 seconds

Process-1954, Step 3: Get HOG channels: 6e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.13143 seconds

Process-1954, Step 6: Misc initializations: 0.0 seconds

Process-1954, Step 7: for loop: 0.0072 seconds

Process-1954All steps time: 0.149897 seconds

Process-1954, Find cars processing time: 0.14996 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.010875 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.000879 seconds

Process-1954, Step 3: Get HOG channels: 9e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.136038 seconds

Process-1954, Step 6: Misc initializations: 0.0 seconds

Process-1954, Step 7: for loop: 0.007773 seconds

Process-1954All steps time: 0.155584 seconds

Process-1954, Find cars processing time: 0.155667 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.010394 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.00134 seconds

Process-1954, Step 3: Get HOG channels: 9e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.094907 seconds

Process-1954, Step 6: Misc initializations: 1e-06 seconds

Process-1954, Step 7: for loop: 0.005383 seconds

Process-1954All steps time: 0.112043 seconds

Process-1954, Find cars processing time: 0.11211 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.005578 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.000432 seconds

Process-1954, Step 3: Get HOG channels: 5e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.101787 seconds

Process-1954, Step 6: Misc initializations: 0.0 seconds

Process-1954, Step 7: for loop: 0.005419 seconds

Process-1954All steps time: 0.113227 seconds

Process-1954, Find cars processing time: 0.113279 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.010752 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.000869 seconds

Process-1954, Step 3: Get HOG channels: 9e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.093016 seconds

Process-1954, Step 6: Misc initializations: 0.0 seconds

Process-1954, Step 7: for loop: 0.005499 seconds

Process-1954All steps time: 0.110154 seconds

Process-1954, Find cars processing time: 0.110221 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.006365 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.00056 seconds

Process-1954, Step 3: Get HOG channels: 8e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.109047 seconds

Process-1954, Step 6: Misc initializations: 1e-06 seconds

Process-1954, Step 7: for loop: 0.005496 seconds

Process-1954All steps time: 0.1214850000000001 seconds

Process-1954, Find cars processing time: 0.12156 seconds

Process-1954, Step 1: Divide with 255, processing time: 0.008078 seconds

Process-1954, Step 2: Resize if scale is not 1: 0.001222 seconds

Process-1954, Step 3: Get HOG channels: 5e-06 seconds

Process-1954, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1954, Step 5: Compute individual channel HOG features for the entire image: 0.081733 seconds

Process-1954, Step 6: Misc initializations: 0.0 seconds

Process-1954, Step 7: for loop: 0.004582 seconds

Process-1954All steps time: 0.095626 seconds

Process-1954, Find cars processing time: 0.095672 seconds

The times for each task are: [0.276017, 0.258736, 0.243402, 0.262426, 0.157675, 0.184936, 0.1499

Minimum: 0.095672 Maximum: 0.276017 Average: 0.1724 seconds

Number of processes used: 1 window size 260

Length of task list: 13

Number of processes used: 1

Process-1955, Step 1: Divide with 255, processing time: 0.033716 seconds

Process-1955, Step 2: Resize if scale is not 1: 0.002416 seconds

Process-1955, Step 3: Get HOG channels: 3.4e-05 seconds

Process-1955, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.230964 sec

Process-1955, Step 6: Misc initializations: 0.0 seconds

Process-1955, Step 7: for loop: 0.013087 seconds

Process-1955All steps time: 0.2802240000000003 seconds

Process-1955, Find cars processing time: 0.280411 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.011007 seconds

Process-1955, Step 2: Resize if scale is not 1: 0.000955 seconds

Process-1955, Step 3: Get HOG channels: 9e-06 seconds

Process-1955, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.229017 sec

Process-1955, Step 6: Misc initializations: 0.0 seconds

Process-1955, Step 7: for loop: 0.01203 seconds

Process-1955All steps time: 0.253027 seconds

Process-1955, Find cars processing time: 0.253109 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.008229 seconds

Process-1955, Step 2: Resize if scale is not 1: 0.000739 seconds

Process-1955, Step 3: Get HOG channels: 7e-06 seconds

Process-1955, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.177733 sec

Process-1955, Step 6: Misc initializations: 1e-06 seconds

Process-1955, Step 7: for loop: 0.012477 seconds

Process-1955All steps time: 0.1991939999999998 seconds

Process-1955, Find cars processing time: 0.199261 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.014848 seconds

Process-1955, Step 2: Resize if scale is not 1: 0.000739 seconds
Process-1955, Step 3: Get HOG channels: 3e-05 seconds
Process-1955, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.181497 seconds
Process-1955, Step 6: Misc initializations: 0.0 seconds
Process-1955, Step 7: for loop: 0.011789 seconds

Process-1955All steps time: 0.2089089999999998 seconds

Process-1955, Find cars processing time: 0.208992 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.007404 seconds
Process-1955, Step 2: Resize if scale is not 1: 0.000672 seconds
Process-1955, Step 3: Get HOG channels: 6e-06 seconds
Process-1955, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.137783 seconds
Process-1955, Step 6: Misc initializations: 0.0 seconds
Process-1955, Step 7: for loop: 0.007153 seconds

Process-1955All steps time: 0.153025 seconds

Process-1955, Find cars processing time: 0.153089 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.006426 seconds
Process-1955, Step 2: Resize if scale is not 1: 0.000498 seconds
Process-1955, Step 3: Get HOG channels: 5e-06 seconds
Process-1955, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.122854 seconds
Process-1955, Step 6: Misc initializations: 0.0 seconds
Process-1955, Step 7: for loop: 0.007041 seconds

Process-1955All steps time: 0.13683 seconds

Process-1955, Find cars processing time: 0.136873 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.010885 seconds
Process-1955, Step 2: Resize if scale is not 1: 0.000731 seconds
Process-1955, Step 3: Get HOG channels: 1.1e-05 seconds
Process-1955, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.129281 seconds
Process-1955, Step 6: Misc initializations: 0.0 seconds
Process-1955, Step 7: for loop: 0.007426 seconds

Process-1955All steps time: 0.14834 seconds

Process-1955, Find cars processing time: 0.148408 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.018505 seconds

Process-1955, Step 2: Resize if scale is not 1: 0.000489 seconds
Process-1955, Step 3: Get HOG channels: 5e-06 seconds
Process-1955, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.125571 seconds
Process-1955, Step 6: Misc initializations: 0.0 seconds
Process-1955, Step 7: for loop: 0.00713 seconds

Process-1955All steps time: 0.1517049999999998 seconds

Process-1955, Find cars processing time: 0.151769 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.010745 seconds
Process-1955, Step 2: Resize if scale is not 1: 0.000881 seconds
Process-1955, Step 3: Get HOG channels: 6e-06 seconds
Process-1955, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.100313 seconds
Process-1955, Step 6: Misc initializations: 0.0 seconds
Process-1955, Step 7: for loop: 0.006019 seconds

Process-1955All steps time: 0.1179699999999999 seconds

Process-1955, Find cars processing time: 0.118039 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.008598 seconds
Process-1955, Step 2: Resize if scale is not 1: 0.00064 seconds
Process-1955, Step 3: Get HOG channels: 7e-06 seconds
Process-1955, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.104048 seconds
Process-1955, Step 6: Misc initializations: 0.0 seconds
Process-1955, Step 7: for loop: 0.005713 seconds

Process-1955All steps time: 0.119014 seconds

Process-1955, Find cars processing time: 0.119076 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.006469 seconds
Process-1955, Step 2: Resize if scale is not 1: 0.000458 seconds
Process-1955, Step 3: Get HOG channels: 5e-06 seconds
Process-1955, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.099147 seconds
Process-1955, Step 6: Misc initializations: 1e-06 seconds
Process-1955, Step 7: for loop: 0.00586 seconds

Process-1955All steps time: 0.111946 seconds

Process-1955, Find cars processing time: 0.112018 seconds

Process-1955, Step 1: Divide with 255, processing time: 0.007607 seconds

```
Process-1955, Step 2: Resize if scale is not 1: 0.000585 seconds
Process-1955, Step 3: Get HOG channels: 6e-06 seconds
Process-1955, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.126621 seconds
Process-1955, Step 6: Misc initializations: 0.0 seconds
Process-1955, Step 7: for loop: 0.006908 seconds
```

```
Process-1955All steps time: 0.141735 seconds
```

```
Process-1955, Find cars processing time: 0.141805 seconds
```

```
Process-1955, Step 1: Divide with 255, processing time: 0.014772 seconds
Process-1955, Step 2: Resize if scale is not 1: 0.001385 seconds
Process-1955, Step 3: Get HOG channels: 7e-06 seconds
Process-1955, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1955, Step 5: Compute individual channel HOG features for the entire image: 0.130227 seconds
Process-1955, Step 6: Misc initializations: 1e-06 seconds
Process-1955, Step 7: for loop: 0.006303 seconds
```

```
Process-1955All steps time: 0.152702 seconds
```

```
Process-1955, Find cars processing time: 0.152791 seconds
```

```
The times for each task are: [0.280411, 0.253109, 0.199261, 0.208992, 0.153089, 0.136873, 0.1484
```

```
Minimum: 0.112018 Maximum: 0.280411 Average: 0.1674 seconds
```

```
*****
Number of processes used: 1 window size 260
Length of task list: 13
Number of processes used: 1
```

```
Process-1956, Step 1: Divide with 255, processing time: 0.033476 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.002348 seconds
Process-1956, Step 3: Get HOG channels: 2.6e-05 seconds
Process-1956, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.252231 seconds
Process-1956, Step 6: Misc initializations: 1e-06 seconds
Process-1956, Step 7: for loop: 0.01737 seconds
```

```
Process-1956All steps time: 0.305459 seconds
```

```
Process-1956, Find cars processing time: 0.305631 seconds
```

```
Process-1956, Step 1: Divide with 255, processing time: 0.006065 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.000536 seconds
Process-1956, Step 3: Get HOG channels: 5e-06 seconds
Process-1956, Step 4: Define blocks and steps as above: 6e-06 seconds
```

Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.238796 seconds
Process-1956, Step 6: Misc initializations: 0.0 seconds
Process-1956, Step 7: for loop: 0.015451 seconds

Process-1956All steps time: 0.260859 seconds

Process-1956, Find cars processing time: 0.260917 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.0105 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.000906 seconds
Process-1956, Step 3: Get HOG channels: 9e-06 seconds
Process-1956, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.196748 seconds
Process-1956, Step 6: Misc initializations: 1e-06 seconds
Process-1956, Step 7: for loop: 0.013263 seconds

Process-1956All steps time: 0.221437 seconds

Process-1956, Find cars processing time: 0.22152 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.006477 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.00056 seconds
Process-1956, Step 3: Get HOG channels: 5e-06 seconds
Process-1956, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.191462 seconds
Process-1956, Step 6: Misc initializations: 1e-06 seconds
Process-1956, Step 7: for loop: 0.013035 seconds

Process-1956All steps time: 0.2115459999999998 seconds

Process-1956, Find cars processing time: 0.211605 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.010644 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.000784 seconds
Process-1956, Step 3: Get HOG channels: 5e-06 seconds
Process-1956, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.136848 seconds
Process-1956, Step 6: Misc initializations: 0.0 seconds
Process-1956, Step 7: for loop: 0.0117 seconds

Process-1956All steps time: 0.159987 seconds

Process-1956, Find cars processing time: 0.160062 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.010666 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.00083 seconds
Process-1956, Step 3: Get HOG channels: 9e-06 seconds
Process-1956, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.132073 seconds
Process-1956, Step 6: Misc initializations: 0.0 seconds
Process-1956, Step 7: for loop: 0.011487 seconds

Process-1956All steps time: 0.155074 seconds

Process-1956, Find cars processing time: 0.155157 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.010695 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.000741 seconds
Process-1956, Step 3: Get HOG channels: 6e-06 seconds
Process-1956, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.168686 seconds
Process-1956, Step 6: Misc initializations: 1e-06 seconds
Process-1956, Step 7: for loop: 0.008244 seconds

Process-1956All steps time: 0.18838 seconds

Process-1956, Find cars processing time: 0.188474 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.011117 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.00086 seconds
Process-1956, Step 3: Get HOG channels: 8e-06 seconds
Process-1956, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.153409 seconds
Process-1956, Step 6: Misc initializations: 1e-06 seconds
Process-1956, Step 7: for loop: 0.008638 seconds

Process-1956All steps time: 0.174042 seconds

Process-1956, Find cars processing time: 0.174121 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.011016 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.001443 seconds
Process-1956, Step 3: Get HOG channels: 1.3e-05 seconds
Process-1956, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.116254 seconds
Process-1956, Step 6: Misc initializations: 1e-06 seconds
Process-1956, Step 7: for loop: 0.006225 seconds

Process-1956All steps time: 0.134961 seconds

Process-1956, Find cars processing time: 0.135044 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.010617 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.000774 seconds
Process-1956, Step 3: Get HOG channels: 8e-06 seconds
Process-1956, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.13161 seconds
Process-1956, Step 6: Misc initializations: 1e-06 seconds
Process-1956, Step 7: for loop: 0.007143 seconds

Process-1956All steps time: 0.15016200000000002 seconds

Process-1956, Find cars processing time: 0.150239 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.008597 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.000698 seconds
Process-1956, Step 3: Get HOG channels: 8e-06 seconds
Process-1956, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.156384 seconds
Process-1956, Step 6: Misc initializations: 0.0 seconds
Process-1956, Step 7: for loop: 0.007782 seconds

Process-1956All steps time: 0.173477 seconds

Process-1956, Find cars processing time: 0.17357 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.010854 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.000813 seconds
Process-1956, Step 3: Get HOG channels: 1e-05 seconds
Process-1956, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.16885 seconds
Process-1956, Step 6: Misc initializations: 1e-06 seconds
Process-1956, Step 7: for loop: 0.007663 seconds

Process-1956All steps time: 0.1882 seconds

Process-1956, Find cars processing time: 0.188291 seconds

Process-1956, Step 1: Divide with 255, processing time: 0.010114 seconds
Process-1956, Step 2: Resize if scale is not 1: 0.001816 seconds
Process-1956, Step 3: Get HOG channels: 1e-05 seconds
Process-1956, Step 4: Define blocks and steps as above: 1.8e-05 seconds
Process-1956, Step 5: Compute individual channel HOG features for the entire image: 0.133253 seconds
Process-1956, Step 6: Misc initializations: 1e-06 seconds
Process-1956, Step 7: for loop: 0.006372 seconds

Process-1956All steps time: 0.151584 seconds

Process-1956, Find cars processing time: 0.151657 seconds

The times for each task are: [0.305631, 0.260917, 0.22152, 0.211605, 0.160062, 0.155157, 0.18847]

Minimum: 0.135044 Maximum: 0.305631 Average: 0.1905 seconds

```
*****
Window sizes used: [1.15, 1.35, 1.55]
1 processes used for testing 3 window sizes
Processing times for each image [6.3024, 6.2057, 5.9439, 4.9933, 6.6752, 6.0111] with an average

Time elapsed so far... 52.4796
#####
Number of processes used: 2 window size 260
Length of task list: 13
Number of processes used: 2

Process-1957, Step 1: Divide with 255, processing time: 0.032969 seconds
Process-1957, Step 2: Resize if scale is not 1: 0.002618 seconds
Process-1957, Step 3: Get HOG channels: 3.1e-05 seconds
Process-1957, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1957, Step 5: Compute individual channel HOG features for the entire image: 0.241436 seconds
Process-1957, Step 6: Misc initializations: 0.0 seconds
Process-1957, Step 7: for loop: 0.014042 seconds

Process-1957All steps time: 0.291102 seconds

Process-1957, Find cars processing time: 0.291272 seconds

Process-1958, Step 1: Divide with 255, processing time: 0.03326 seconds
Process-1958, Step 2: Resize if scale is not 1: 0.002401 seconds
Process-1958, Step 3: Get HOG channels: 3.2e-05 seconds
Process-1958, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1958, Step 5: Compute individual channel HOG features for the entire image: 0.271478 seconds
Process-1958, Step 6: Misc initializations: 1e-06 seconds
Process-1958, Step 7: for loop: 0.015231 seconds

Process-1958All steps time: 0.3224089999999995 seconds

Process-1958, Find cars processing time: 0.322596 seconds

Process-1958, Step 1: Divide with 255, processing time: 0.010767 seconds
Process-1958, Step 2: Resize if scale is not 1: 0.000941 seconds
Process-1958, Step 3: Get HOG channels: 9e-06 seconds
Process-1958, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1958, Step 5: Compute individual channel HOG features for the entire image: 0.190899 seconds
Process-1958, Step 6: Misc initializations: 1e-06 seconds
Process-1958, Step 7: for loop: 0.013518 seconds

Process-1958All steps time: 0.216144 seconds

Process-1958, Find cars processing time: 0.216223 seconds
```

Process-1957, Step 1: Divide with 255, processing time: 0.010572 seconds
Process-1957, Step 2: Resize if scale is not 1: 0.000512 seconds
Process-1957, Step 3: Get HOG channels: 5e-06 seconds
Process-1957, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1957, Step 5: Compute individual channel HOG features for the entire image: 0.189491 seconds
Process-1957, Step 6: Misc initializations: 1e-06 seconds
Process-1957, Step 7: for loop: 0.012182 seconds

Process-1957All steps time: 0.2127679999999998 seconds

Process-1957, Find cars processing time: 0.212821 seconds

Process-1957, Step 1: Divide with 255, processing time: 0.00958 seconds
Process-1957, Step 2: Resize if scale is not 1: 0.000908 seconds
Process-1957, Step 3: Get HOG channels: 8e-06 seconds
Process-1957, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1957, Step 5: Compute individual channel HOG features for the entire image: 0.137416 seconds
Process-1957, Step 6: Misc initializations: 0.0 seconds
Process-1957, Step 7: for loop: 0.00756 seconds

Process-1957All steps time: 0.1554810000000004 seconds

Process-1957, Find cars processing time: 0.155546 seconds

Process-1958, Step 1: Divide with 255, processing time: 0.010755 seconds
Process-1958, Step 2: Resize if scale is not 1: 0.000938 seconds
Process-1958, Step 3: Get HOG channels: 8e-06 seconds
Process-1958, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1958, Step 5: Compute individual channel HOG features for the entire image: 0.172358 seconds
Process-1958, Step 6: Misc initializations: 0.0 seconds
Process-1958, Step 7: for loop: 0.009544 seconds

Process-1958All steps time: 0.193611 seconds

Process-1958, Find cars processing time: 0.193674 seconds

Process-1957, Step 1: Divide with 255, processing time: 0.007254 seconds
Process-1957, Step 2: Resize if scale is not 1: 0.000589 seconds
Process-1957, Step 3: Get HOG channels: 6e-06 seconds
Process-1957, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1957, Step 5: Compute individual channel HOG features for the entire image: 0.133324 seconds
Process-1957, Step 6: Misc initializations: 1e-06 seconds
Process-1957, Step 7: for loop: 0.007612 seconds

Process-1957All steps time: 0.148792 seconds

Process-1957, Find cars processing time: 0.148847 seconds

Process-1957, Step 1: Divide with 255, processing time: 0.010222 seconds
Process-1957, Step 2: Resize if scale is not 1: 0.001389 seconds
Process-1957, Step 3: Get HOG channels: 9e-06 seconds
Process-1957, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1957, Step 5: Compute individual channel HOG features for the entire image: 0.095321 seconds
Process-1957, Step 6: Misc initializations: 0.0 seconds
Process-1957, Step 7: for loop: 0.005322 seconds

Process-1957All steps time: 0.11227200000000001 seconds

Process-1957, Find cars processing time: 0.112331 seconds

Process-1958, Step 1: Divide with 255, processing time: 0.010724 seconds
Process-1958, Step 2: Resize if scale is not 1: 0.000587 seconds
Process-1958, Step 3: Get HOG channels: 6e-06 seconds
Process-1958, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1958, Step 5: Compute individual channel HOG features for the entire image: 0.141237 seconds
Process-1958, Step 6: Misc initializations: 0.0 seconds
Process-1958, Step 7: for loop: 0.007721 seconds

Process-1958All steps time: 0.160281 seconds

Process-1958, Find cars processing time: 0.160347 seconds

Process-1957, Step 1: Divide with 255, processing time: 0.010037 seconds
Process-1957, Step 2: Resize if scale is not 1: 0.000711 seconds
Process-1957, Step 3: Get HOG channels: 6e-06 seconds
Process-1957, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1957, Step 5: Compute individual channel HOG features for the entire image: 0.102576 seconds
Process-1957, Step 6: Misc initializations: 0.0 seconds
Process-1957, Step 7: for loop: 0.005568 seconds

Process-1957All steps time: 0.11890500000000001 seconds

Process-1957, Find cars processing time: 0.118967 seconds

Process-1958, Step 1: Divide with 255, processing time: 0.010418 seconds
Process-1958, Step 2: Resize if scale is not 1: 0.001416 seconds
Process-1958, Step 3: Get HOG channels: 8e-06 seconds
Process-1958, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1958, Step 5: Compute individual channel HOG features for the entire image: 0.097367 seconds
Process-1958, Step 6: Misc initializations: 1e-06 seconds
Process-1958, Step 7: for loop: 0.006054 seconds

Process-1958All steps time: 0.115273 seconds

Process-1958, Find cars processing time: 0.115352 seconds

Process-1957, Step 1: Divide with 255, processing time: 0.01063 seconds
Process-1957, Step 2: Resize if scale is not 1: 0.000705 seconds
Process-1957, Step 3: Get HOG channels: 6e-06 seconds
Process-1957, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1957, Step 5: Compute individual channel HOG features for the entire image: 0.132612 seconds
Process-1957, Step 6: Misc initializations: 0.0 seconds
Process-1957, Step 7: for loop: 0.007134 seconds

Process-1957All steps time: 0.151093 seconds

Process-1957, Find cars processing time: 0.151155 seconds

Process-1958, Step 1: Divide with 255, processing time: 0.008429 seconds
Process-1958, Step 2: Resize if scale is not 1: 0.001547 seconds
Process-1958, Step 3: Get HOG channels: 7e-06 seconds
Process-1958, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1958, Step 5: Compute individual channel HOG features for the entire image: 0.091156 seconds
Process-1958, Step 6: Misc initializations: 1e-06 seconds
Process-1958, Step 7: for loop: 0.004812 seconds

Process-1958All steps time: 0.105958 seconds

Process-1958, Find cars processing time: 0.106012 seconds

The times for each task are: [0.291272, 0.322596, 0.216223, 0.212821, 0.155546, 0.193674, 0.1488

Minimum: 0.106012 Maximum: 0.322596 Average: 0.1773 seconds

Number of processes used: 2 window size 260
Length of task list: 13
Number of processes used: 2

Process-1959, Step 1: Divide with 255, processing time: 0.031843 seconds
Process-1959, Step 2: Resize if scale is not 1: 0.002367 seconds
Process-1959, Step 3: Get HOG channels: 2.4e-05 seconds
Process-1959, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1959, Step 5: Compute individual channel HOG features for the entire image: 0.228096 seconds
Process-1959, Step 6: Misc initializations: 0.0 seconds
Process-1959, Step 7: for loop: 0.014386 seconds

Process-1959All steps time: 0.276722 seconds

Process-1959, Find cars processing time: 0.276884 seconds

Process-1959, Step 1: Divide with 255, processing time: 0.006414 seconds
Process-1959, Step 2: Resize if scale is not 1: 0.000654 seconds
Process-1959, Step 3: Get HOG channels: 6e-06 seconds

Process-1959, Step 4: Define blocks and steps as above: 1.7e-05 seconds
Process-1959, Step 5: Compute individual channel HOG features for the entire image: 0.185516 seconds
Process-1959, Step 6: Misc initializations: 0.0 seconds
Process-1959, Step 7: for loop: 0.012302 seconds

Process-1959All steps time: 0.2049089999999998 seconds

Process-1959, Find cars processing time: 0.204983 seconds

Process-1960, Step 1: Divide with 255, processing time: 0.03505 seconds
Process-1960, Step 2: Resize if scale is not 1: 0.002327 seconds
Process-1960, Step 3: Get HOG channels: 2.2e-05 seconds
Process-1960, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1960, Step 5: Compute individual channel HOG features for the entire image: 0.25497 seconds
Process-1960, Step 6: Misc initializations: 1e-06 seconds
Process-1960, Step 7: for loop: 0.017647 seconds

Process-1960All steps time: 0.310023 seconds

Process-1960, Find cars processing time: 0.310186 seconds

Process-1959, Step 1: Divide with 255, processing time: 0.010643 seconds
Process-1959, Step 2: Resize if scale is not 1: 0.000567 seconds
Process-1959, Step 3: Get HOG channels: 6e-06 seconds
Process-1959, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1959, Step 5: Compute individual channel HOG features for the entire image: 0.182231 seconds
Process-1959, Step 6: Misc initializations: 1e-06 seconds
Process-1959, Step 7: for loop: 0.012125 seconds

Process-1959All steps time: 0.205579 seconds

Process-1959, Find cars processing time: 0.205656 seconds

Process-1960, Step 1: Divide with 255, processing time: 0.010906 seconds
Process-1960, Step 2: Resize if scale is not 1: 0.000891 seconds
Process-1960, Step 3: Get HOG channels: 9e-06 seconds
Process-1960, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1960, Step 5: Compute individual channel HOG features for the entire image: 0.177793 seconds
Process-1960, Step 6: Misc initializations: 1e-06 seconds
Process-1960, Step 7: for loop: 0.009784 seconds

Process-1960All steps time: 0.1993919999999999 seconds
Process-1959, Step 1: Divide with 255, processing time: 0.009509 seconds

Process-1959, Step 2: Resize if scale is not 1: 0.000781 seconds
Process-1960, Find cars processing time: 0.199463 seconds
Process-1959, Step 3: Get HOG channels: 8e-06 seconds

Process-1959, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1959, Step 5: Compute individual channel HOG features for the entire image: 0.171892 seconds

Process-1959, Step 6: Misc initializations: 1e-06 seconds

Process-1959, Step 7: for loop: 0.009988 seconds

Process-1959, Find cars processing time: 0.192248 seconds

Process-1959All steps time: 0.192186 seconds

Process-1959, Step 1: Divide with 255, processing time: 0.011455 seconds

Process-1959, Step 2: Resize if scale is not 1: 0.00083 seconds

Process-1959, Step 3: Get HOG channels: 8e-06 seconds

Process-1959, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1960, Step 1: Divide with 255, processing time: 0.011566 seconds

Process-1959, Step 5: Compute individual channel HOG features for the entire image: 0.167239 seconds

Process-1960, Step 2: Resize if scale is not 1: 0.000875 seconds

Process-1959, Step 6: Misc initializations: 1e-06 seconds

Process-1960, Step 3: Get HOG channels: 9e-06 seconds

Process-1959, Step 7: for loop: 0.009627 seconds

Process-1960, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1960, Step 5: Compute individual channel HOG features for the entire image: 0.171197 seconds

Process-1959All steps time: 0.189167 seconds

Process-1960, Step 6: Misc initializations: 1e-06 seconds

Process-1960, Step 7: for loop: 0.010069 seconds

Process-1959, Find cars processing time: 0.189236 seconds

Process-1960All steps time: 0.1937249999999998 seconds

Process-1960, Find cars processing time: 0.193807 seconds

Process-1959, Step 1: Divide with 255, processing time: 0.011 seconds

Process-1959, Step 2: Resize if scale is not 1: 0.000964 seconds

Process-1959, Step 3: Get HOG channels: 6e-06 seconds

Process-1959, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1959, Step 5: Compute individual channel HOG features for the entire image: 0.125887 seconds

Process-1959, Step 6: Misc initializations: 0.0 seconds

Process-1959, Step 7: for loop: 0.006484 seconds

Process-1959All steps time: 0.144347 seconds

Process-1959, Find cars processing time: 0.144427 seconds

Process-1960, Step 1: Divide with 255, processing time: 0.007466 seconds

Process-1960, Step 2: Resize if scale is not 1: 0.001014 seconds

Process-1960, Step 3: Get HOG channels: 6e-06 seconds

Process-1960, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1960, Step 5: Compute individual channel HOG features for the entire image: 0.137292 seconds
Process-1960, Step 6: Misc initializations: 0.0 seconds
Process-1960, Step 7: for loop: 0.007145 seconds

Process-1960All steps time: 0.152928 seconds

Process-1960, Find cars processing time: 0.15298 seconds

Process-1960, Step 1: Divide with 255, processing time: 0.009751 seconds
Process-1960, Step 2: Resize if scale is not 1: 0.000692 seconds
Process-1960, Step 3: Get HOG channels: 2e-05 seconds
Process-1960, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1960, Step 5: Compute individual channel HOG features for the entire image: 0.126667 seconds
Process-1960, Step 6: Misc initializations: 1e-06 seconds
Process-1960, Step 7: for loop: 0.007383 seconds

Process-1960All steps time: 0.144523 seconds

Process-1960, Find cars processing time: 0.144587 seconds

Process-1959, Step 1: Divide with 255, processing time: 0.009629 seconds
Process-1959, Step 2: Resize if scale is not 1: 0.000752 seconds
Process-1959, Step 3: Get HOG channels: 5e-06 seconds
Process-1959, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1959, Step 5: Compute individual channel HOG features for the entire image: 0.126098 seconds
Process-1959, Step 6: Misc initializations: 1e-06 seconds
Process-1959, Step 7: for loop: 0.007336 seconds

Process-1959All steps time: 0.143826 seconds

Process-1959, Find cars processing time: 0.143891 seconds

Process-1960, Step 1: Divide with 255, processing time: 0.009584 seconds
Process-1960, Step 2: Resize if scale is not 1: 0.001786 seconds
Process-1960, Step 3: Get HOG channels: 8e-06 seconds
Process-1960, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1960, Step 5: Compute individual channel HOG features for the entire image: 0.113946 seconds
Process-1960, Step 6: Misc initializations: 1e-06 seconds
Process-1960, Step 7: for loop: 0.00612 seconds

Process-1960All steps time: 0.131453 seconds

Process-1960, Find cars processing time: 0.131532 seconds

The times for each task are: [0.276884, 0.204983, 0.310186, 0.205656, 0.199463, 0.192248, 0.189241]
Minimum: 0.131532 Maximum: 0.310186 Average: 0.1915 seconds

```
*****
Number of processes used: 2 window size 260
Length of task list: 13
Number of processes used: 2

Process-1961, Step 1: Divide with 255, processing time: 0.032986 seconds
Process-1961, Step 2: Resize if scale is not 1: 0.002416 seconds
Process-1961, Step 3: Get HOG channels: 2.4e-05 seconds
Process-1961, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1961, Step 5: Compute individual channel HOG features for the entire image: 0.261226 sec
Process-1961, Step 6: Misc initializations: 1e-06 seconds
Process-1961, Step 7: for loop: 0.016402 seconds

Process-1961All steps time: 0.31306100000000003 seconds

Process-1961, Find cars processing time: 0.313224 seconds

Process-1962, Step 1: Divide with 255, processing time: 0.044513 seconds
Process-1962, Step 2: Resize if scale is not 1: 0.003114 seconds
Process-1962, Step 3: Get HOG channels: 5.4e-05 seconds
Process-1962, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1962, Step 5: Compute individual channel HOG features for the entire image: 0.210565 sec
Process-1962, Step 6: Misc initializations: 1e-06 seconds
Process-1962, Step 7: for loop: 0.015398 seconds

Process-1962All steps time: 0.27365 seconds

Process-1962, Find cars processing time: 0.27378 seconds

Process-1961, Step 1: Divide with 255, processing time: 0.006598 seconds
Process-1961, Step 2: Resize if scale is not 1: 0.00069 seconds
Process-1961, Step 3: Get HOG channels: 7e-06 seconds
Process-1961, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1961, Step 5: Compute individual channel HOG features for the entire image: 0.237012 sec
Process-1961, Step 6: Misc initializations: 1e-06 seconds
Process-1961, Step 7: for loop: 0.01615 seconds

Process-1961All steps time: 0.26046400000000003 seconds

Process-1961, Find cars processing time: 0.260528 seconds

Process-1962, Step 1: Divide with 255, processing time: 0.010669 seconds
Process-1962, Step 2: Resize if scale is not 1: 0.000916 seconds
Process-1962, Step 3: Get HOG channels: 9e-06 seconds
Process-1962, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1962, Step 5: Compute individual channel HOG features for the entire image: 0.206096 sec
Process-1962, Step 6: Misc initializations: 1e-06 seconds
```

Process-1962, Step 7: for loop: 0.022107 seconds

Process-1962All steps time: 0.23980600000000002 seconds

Process-1962, Find cars processing time: 0.239907 seconds

Process-1961, Step 1: Divide with 255, processing time: 0.010854 seconds

Process-1961, Step 2: Resize if scale is not 1: 0.000734 seconds

Process-1961, Step 3: Get HOG channels: 6e-06 seconds

Process-1961, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1961, Step 5: Compute individual channel HOG features for the entire image: 0.168605 seconds

Process-1961, Step 6: Misc initializations: 0.0 seconds

Process-1961, Step 7: for loop: 0.008113 seconds

Process-1961All steps time: 0.188318 seconds

Process-1961, Find cars processing time: 0.18838 seconds

Process-1962, Step 1: Divide with 255, processing time: 0.010753 seconds

Process-1962, Step 2: Resize if scale is not 1: 0.000702 seconds

Process-1962, Step 3: Get HOG channels: 7e-06 seconds

Process-1962, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1962, Step 5: Compute individual channel HOG features for the entire image: 0.172583 seconds

Process-1962, Step 6: Misc initializations: 1e-06 seconds

Process-1962, Step 7: for loop: 0.009608 seconds

Process-1962All steps time: 0.193661 seconds

Process-1962, Find cars processing time: 0.193744 seconds

Process-1961, Step 1: Divide with 255, processing time: 0.010816 seconds

Process-1961, Step 2: Resize if scale is not 1: 0.000885 seconds

Process-1961, Step 3: Get HOG channels: 8e-06 seconds

Process-1961, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1961, Step 5: Compute individual channel HOG features for the entire image: 0.192441 seconds

Process-1961, Step 6: Misc initializations: 1e-06 seconds

Process-1961, Step 7: for loop: 0.010191 seconds

Process-1961All steps time: 0.21435 seconds

Process-1961, Find cars processing time: 0.214433 seconds

Process-1962, Step 1: Divide with 255, processing time: 0.009737 seconds

Process-1962, Step 2: Resize if scale is not 1: 0.000542 seconds

Process-1962, Step 3: Get HOG channels: 6e-06 seconds

Process-1962, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1962, Step 5: Compute individual channel HOG features for the entire image: 0.210936 seconds

Process-1962, Step 6: Misc initializations: 1e-06 seconds

Process-1962, Step 7: for loop: 0.010656 seconds

Process-1962All steps time: 0.231884 seconds

Process-1962, Find cars processing time: 0.231951 seconds

Process-1961, Step 1: Divide with 255, processing time: 0.009632 seconds

Process-1961, Step 2: Resize if scale is not 1: 0.001144 seconds

Process-1961, Step 3: Get HOG channels: 7e-06 seconds

Process-1961, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1961, Step 5: Compute individual channel HOG features for the entire image: 0.12824 seconds

Process-1961, Step 6: Misc initializations: 0.0 seconds

Process-1961, Step 7: for loop: 0.007276 seconds

Process-1961All steps time: 0.146305 seconds

Process-1961, Find cars processing time: 0.146372 seconds

Process-1961, Step 1: Divide with 255, processing time: 0.007313 seconds

Process-1961, Step 2: Resize if scale is not 1: 0.000556 seconds

Process-1961, Step 3: Get HOG channels: 7e-06 seconds

Process-1961, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1961, Step 5: Compute individual channel HOG features for the entire image: 0.127444 seconds

Process-1961, Step 6: Misc initializations: 0.0 seconds

Process-1961, Step 7: for loop: 0.007157 seconds

Process-1961All steps time: 0.142484 seconds

Process-1961, Find cars processing time: 0.142551 seconds

Process-1962, Step 1: Divide with 255, processing time: 0.010431 seconds

Process-1962, Step 2: Resize if scale is not 1: 0.000887 seconds

Process-1962, Step 3: Get HOG channels: 6e-06 seconds

Process-1962, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1962, Step 5: Compute individual channel HOG features for the entire image: 0.135959 seconds

Process-1962, Step 6: Misc initializations: 1e-06 seconds

Process-1962, Step 7: for loop: 0.00838 seconds

Process-1962All steps time: 0.15567 seconds

Process-1962, Find cars processing time: 0.155745 seconds

Process-1961, Step 1: Divide with 255, processing time: 0.010941 seconds

Process-1961, Step 2: Resize if scale is not 1: 0.000775 seconds

Process-1961, Step 3: Get HOG channels: 8e-06 seconds

Process-1961, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1961, Step 5: Compute individual channel HOG features for the entire image: 0.135611 seconds

Process-1961, Step 6: Misc initializations: 1e-06 seconds

Process-1961, Step 7: for loop: 0.007427 seconds

Process-1961All steps time: 0.15477 seconds

Process-1961, Find cars processing time: 0.15484 seconds

Process-1962, Step 1: Divide with 255, processing time: 0.0108 seconds

Process-1962, Step 2: Resize if scale is not 1: 0.002026 seconds

Process-1962, Step 3: Get HOG channels: 9e-06 seconds

Process-1962, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1962, Step 5: Compute individual channel HOG features for the entire image: 0.121463 seconds

Process-1962, Step 6: Misc initializations: 1e-06 seconds

Process-1962, Step 7: for loop: 0.006354 seconds

Process-1962All steps time: 0.140663 seconds

Process-1962, Find cars processing time: 0.140756 seconds

The times for each task are: [0.313224, 0.27378, 0.260528, 0.239907, 0.18838, 0.193744, 0.214433]

Minimum: 0.140756 Maximum: 0.313224 Average: 0.2043 seconds

Number of processes used: 2 window size 260

Length of task list: 13

Number of processes used: 2

Process-1963, Step 1: Divide with 255, processing time: 0.026781 seconds

Process-1963, Step 2: Resize if scale is not 1: 0.002079 seconds

Process-1963, Step 3: Get HOG channels: 2e-05 seconds

Process-1963, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1963, Step 5: Compute individual channel HOG features for the entire image: 0.24414 seconds

Process-1963, Step 6: Misc initializations: 1e-06 seconds

Process-1963, Step 7: for loop: 0.013585 seconds

Process-1963All steps time: 0.286612 seconds

Process-1963, Find cars processing time: 0.286781 seconds

Process-1964, Step 1: Divide with 255, processing time: 0.023285 seconds

Process-1964, Step 2: Resize if scale is not 1: 0.001678 seconds

Process-1964, Step 3: Get HOG channels: 1.6e-05 seconds

Process-1964, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1964, Step 5: Compute individual channel HOG features for the entire image: 0.181806 seconds

Process-1964, Step 6: Misc initializations: 1e-06 seconds

Process-1964, Step 7: for loop: 0.013622 seconds

Process-1964All steps time: 0.220414 seconds

Process-1964, Find cars processing time: 0.220542 seconds

Process-1963, Step 1: Divide with 255, processing time: 0.007859 seconds

Process-1963, Step 2: Resize if scale is not 1: 0.00068 seconds

Process-1963, Step 3: Get HOG channels: 7e-06 seconds

Process-1963, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1963, Step 5: Compute individual channel HOG features for the entire image: 0.185327 seconds

Process-1963, Step 6: Misc initializations: 1e-06 seconds

Process-1963, Step 7: for loop: 0.013037 seconds

Process-1963All steps time: 0.206917 seconds

Process-1963, Find cars processing time: 0.20698 seconds

Process-1964, Step 1: Divide with 255, processing time: 0.010541 seconds

Process-1964, Step 2: Resize if scale is not 1: 0.000496 seconds

Process-1964, Step 3: Get HOG channels: 5e-06 seconds

Process-1964, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1964, Step 5: Compute individual channel HOG features for the entire image: 0.173638 seconds

Process-1964, Step 6: Misc initializations: 0.0 seconds

Process-1964, Step 7: for loop: 0.01194 seconds

Process-1964All steps time: 0.196625 seconds

Process-1964, Find cars processing time: 0.196671 seconds

Process-1964, Step 1: Divide with 255, processing time: 0.010302 seconds

Process-1964, Step 2: Resize if scale is not 1: 0.00093 seconds

Process-1964, Step 3: Get HOG channels: 9e-06 seconds

Process-1964, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1964, Step 5: Compute individual channel HOG features for the entire image: 0.128842 seconds

Process-1964, Step 6: Misc initializations: 0.0 seconds

Process-1964, Step 7: for loop: 0.00747 seconds

Process-1964All steps time: 0.14756200000000003 seconds

Process-1964, Find cars processing time: 0.147625 seconds

Process-1963, Step 1: Divide with 255, processing time: 0.010469 seconds

Process-1963, Step 2: Resize if scale is not 1: 0.000966 seconds

Process-1963, Step 3: Get HOG channels: 8e-06 seconds

Process-1963, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1963, Step 5: Compute individual channel HOG features for the entire image: 0.168171 seconds

Process-1963, Step 6: Misc initializations: 1e-06 seconds

Process-1963, Step 7: for loop: 0.007487 seconds

Process-1963All steps time: 0.1871089999999997 seconds

Process-1963, Find cars processing time: 0.187212 seconds

Process-1964, Step 1: Divide with 255, processing time: 0.010151 seconds

Process-1964, Step 2: Resize if scale is not 1: 0.000866 seconds

Process-1964, Step 3: Get HOG channels: 9e-06 seconds

Process-1964, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1964, Step 5: Compute individual channel HOG features for the entire image: 0.128071 seconds

Process-1964, Step 6: Misc initializations: 0.0 seconds

Process-1964, Step 7: for loop: 0.007681 seconds

Process-1964All steps time: 0.14678799999999997 seconds

Process-1964, Find cars processing time: 0.146863 seconds

Process-1963, Step 1: Divide with 255, processing time: 0.010555 seconds

Process-1963, Step 2: Resize if scale is not 1: 0.000898 seconds

Process-1963, Step 3: Get HOG channels: 8e-06 seconds

Process-1963, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1963, Step 5: Compute individual channel HOG features for the entire image: 0.131078 seconds

Process-1963, Step 6: Misc initializations: 0.0 seconds

Process-1963, Step 7: for loop: 0.007235 seconds

Process-1963All steps time: 0.149782 seconds

Process-1963, Find cars processing time: 0.149848 seconds

Process-1964, Step 1: Divide with 255, processing time: 0.010514 seconds

Process-1964, Step 2: Resize if scale is not 1: 0.001032 seconds

Process-1964, Step 3: Get HOG channels: 7e-06 seconds

Process-1964, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1964, Step 5: Compute individual channel HOG features for the entire image: 0.157187 seconds

Process-1964, Step 6: Misc initializations: 1e-06 seconds

Process-1964, Step 7: for loop: 0.007593 seconds

Process-1964All steps time: 0.17634099999999997 seconds

Process-1964, Find cars processing time: 0.176446 seconds

Process-1963, Step 1: Divide with 255, processing time: 0.010783 seconds

Process-1963, Step 2: Resize if scale is not 1: 0.00088 seconds

Process-1963, Step 3: Get HOG channels: 6e-06 seconds

Process-1963, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1963, Step 5: Compute individual channel HOG features for the entire image: 0.094292 seconds

Process-1963, Step 6: Misc initializations: 1e-06 seconds

Process-1963, Step 7: for loop: 0.005546 seconds

Process-1963All steps time: 0.1111515 seconds

Process-1963, Find cars processing time: 0.111579 seconds

Process-1964, Step 1: Divide with 255, processing time: 0.010859 seconds

Process-1964, Step 2: Resize if scale is not 1: 0.000786 seconds

Process-1964, Step 3: Get HOG channels: 6e-06 seconds

Process-1964, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1964, Step 5: Compute individual channel HOG features for the entire image: 0.140558 seconds

Process-1964, Step 6: Misc initializations: 1e-06 seconds

Process-1964, Step 7: for loop: 0.007246 seconds

Process-1964All steps time: 0.159461 seconds

Process-1964, Find cars processing time: 0.159523 seconds

Process-1963, Step 1: Divide with 255, processing time: 0.010339 seconds

Process-1963, Step 2: Resize if scale is not 1: 0.0008 seconds

Process-1963, Step 3: Get HOG channels: 9e-06 seconds

Process-1963, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1963, Step 5: Compute individual channel HOG features for the entire image: 0.10741 seconds

Process-1963, Step 6: Misc initializations: 1e-06 seconds

Process-1963, Step 7: for loop: 0.005297 seconds

Process-1963All steps time: 0.123864 seconds

Process-1963, Find cars processing time: 0.123934 seconds

Process-1964, Step 1: Divide with 255, processing time: 0.010503 seconds

Process-1964, Step 2: Resize if scale is not 1: 0.001348 seconds

Process-1964, Step 3: Get HOG channels: 7e-06 seconds

Process-1964, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1964, Step 5: Compute individual channel HOG features for the entire image: 0.111723 seconds

Process-1964, Step 6: Misc initializations: 1e-06 seconds

Process-1964, Step 7: for loop: 0.006006 seconds

Process-1964All steps time: 0.12959500000000002 seconds

Process-1964, Find cars processing time: 0.129665 seconds

The times for each task are: [0.286781, 0.220542, 0.20698, 0.196671, 0.147625, 0.187212, 0.14686

Minimum: 0.111579 Maximum: 0.286781 Average: 0.1726 seconds

Number of processes used: 2 window size 260

Length of task list: 13

Number of processes used: 2

Process-1965, Step 1: Divide with 255, processing time: 0.029198 seconds
Process-1965, Step 2: Resize if scale is not 1: 0.002102 seconds
Process-1965, Step 3: Get HOG channels: 2.6e-05 seconds
Process-1965, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1965, Step 5: Compute individual channel HOG features for the entire image: 0.234996 seconds
Process-1965, Step 6: Misc initializations: 0.0 seconds
Process-1965, Step 7: for loop: 0.013556 seconds

Process-1965All steps time: 0.279884 seconds

Process-1965, Find cars processing time: 0.280056 seconds

Process-1966, Step 1: Divide with 255, processing time: 0.026534 seconds
Process-1966, Step 2: Resize if scale is not 1: 0.002069 seconds
Process-1966, Step 3: Get HOG channels: 2.1e-05 seconds
Process-1966, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1966, Step 5: Compute individual channel HOG features for the entire image: 0.226539 seconds
Process-1966, Step 6: Misc initializations: 1e-06 seconds
Process-1966, Step 7: for loop: 0.017139 seconds

Process-1966All steps time: 0.2723089999999997 seconds

Process-1966, Find cars processing time: 0.272467 seconds

Process-1965, Step 1: Divide with 255, processing time: 0.010148 seconds
Process-1965, Step 2: Resize if scale is not 1: 0.000722 seconds
Process-1965, Step 3: Get HOG channels: 5e-06 seconds
Process-1965, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1965, Step 5: Compute individual channel HOG features for the entire image: 0.177068 seconds
Process-1965, Step 6: Misc initializations: 0.0 seconds
Process-1965, Step 7: for loop: 0.012161 seconds

Process-1965All steps time: 0.200109 seconds

Process-1965, Find cars processing time: 0.200167 seconds

Process-1966, Step 1: Divide with 255, processing time: 0.010672 seconds
Process-1966, Step 2: Resize if scale is not 1: 0.000862 seconds
Process-1966, Step 3: Get HOG channels: 8e-06 seconds
Process-1966, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1966, Step 5: Compute individual channel HOG features for the entire image: 0.230055 seconds
Process-1966, Step 6: Misc initializations: 1e-06 seconds
Process-1966, Step 7: for loop: 0.012583 seconds

Process-1966All steps time: 0.254188 seconds

Process-1966, Find cars processing time: 0.254252 seconds

Process-1965, Step 1: Divide with 255, processing time: 0.010425 seconds
Process-1965, Step 2: Resize if scale is not 1: 0.000603 seconds
Process-1965, Step 3: Get HOG channels: 6e-06 seconds
Process-1965, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1965, Step 5: Compute individual channel HOG features for the entire image: 0.134102 seconds
Process-1965, Step 6: Misc initializations: 0.0 seconds
Process-1965, Step 7: for loop: 0.007348 seconds

Process-1965All steps time: 0.15249 seconds

Process-1965, Find cars processing time: 0.152564 seconds

Process-1966, Step 1: Divide with 255, processing time: 0.010565 seconds
Process-1966, Step 2: Resize if scale is not 1: 0.00073 seconds
Process-1966, Step 3: Get HOG channels: 7e-06 seconds
Process-1966, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1966, Step 5: Compute individual channel HOG features for the entire image: 0.173376 seconds
Process-1966, Step 6: Misc initializations: 1e-06 seconds
Process-1966, Step 7: for loop: 0.00956 seconds

Process-1966All steps time: 0.1942450000000003 seconds

Process-1966, Find cars processing time: 0.194328 seconds

Process-1966, Step 1: Divide with 255, processing time: 0.030423 seconds
Process-1966, Step 2: Resize if scale is not 1: 0.000614 seconds
Process-1966, Step 3: Get HOG channels: 6e-06 seconds
Process-1966, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1966, Step 5: Compute individual channel HOG features for the entire image: 0.16612 seconds
Process-1966, Step 6: Misc initializations: 1e-06 seconds
Process-1966, Step 7: for loop: 0.007832 seconds

Process-1966All steps time: 0.205003 seconds

Process-1966, Find cars processing time: 0.205083 seconds

Process-1965, Step 1: Divide with 255, processing time: 0.010207 seconds
Process-1965, Step 2: Resize if scale is not 1: 0.000772 seconds
Process-1965, Step 3: Get HOG channels: 8e-06 seconds
Process-1965, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1965, Step 5: Compute individual channel HOG features for the entire image: 0.15258 seconds
Process-1965, Step 6: Misc initializations: 1e-06 seconds
Process-1965, Step 7: for loop: 0.007673 seconds

Process-1965All steps time: 0.171248 seconds

Process-1965, Find cars processing time: 0.171314 seconds

Process-1966, Step 1: Divide with 255, processing time: 0.010672 seconds
Process-1966, Step 2: Resize if scale is not 1: 0.001333 seconds
Process-1966, Step 3: Get HOG channels: 9e-06 seconds
Process-1966, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1966, Step 5: Compute individual channel HOG features for the entire image: 0.101254 seconds
Process-1966, Step 6: Misc initializations: 0.0 seconds
Process-1966, Step 7: for loop: 0.005639 seconds

Process-1966All steps time: 0.118916 seconds

Process-1966, Find cars processing time: 0.118984 seconds

Process-1965, Step 1: Divide with 255, processing time: 0.010211 seconds
Process-1965, Step 2: Resize if scale is not 1: 0.000828 seconds
Process-1965, Step 3: Get HOG channels: 5e-06 seconds
Process-1965, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1965, Step 5: Compute individual channel HOG features for the entire image: 0.09812 seconds
Process-1965, Step 6: Misc initializations: 0.0 seconds
Process-1965, Step 7: for loop: 0.005668 seconds

Process-1965All steps time: 0.11483700000000001 seconds

Process-1965, Find cars processing time: 0.114894 seconds

Process-1966, Step 1: Divide with 255, processing time: 0.010835 seconds
Process-1966, Step 2: Resize if scale is not 1: 0.000564 seconds
Process-1966, Step 3: Get HOG channels: 6e-06 seconds
Process-1966, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1966, Step 5: Compute individual channel HOG features for the entire image: 0.133196 seconds
Process-1966, Step 6: Misc initializations: 1e-06 seconds
Process-1966, Step 7: for loop: 0.007216 seconds

Process-1966All steps time: 0.15182600000000002 seconds

Process-1966, Find cars processing time: 0.151902 seconds

Process-1965, Step 1: Divide with 255, processing time: 0.010223 seconds
Process-1965, Step 2: Resize if scale is not 1: 0.000765 seconds
Process-1965, Step 3: Get HOG channels: 8e-06 seconds
Process-1965, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1965, Step 5: Compute individual channel HOG features for the entire image: 0.094448 seconds
Process-1965, Step 6: Misc initializations: 0.0 seconds
Process-1965, Step 7: for loop: 0.005457 seconds

Process-1965All steps time: 0.11090900000000001 seconds

Process-1965, Find cars processing time: 0.110968 seconds

Process-1966, Step 1: Divide with 255, processing time: 0.010471 seconds
Process-1966, Step 2: Resize if scale is not 1: 0.001904 seconds
Process-1966, Step 3: Get HOG channels: 8e-06 seconds
Process-1966, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1966, Step 5: Compute individual channel HOG features for the entire image: 0.129951 seconds
Process-1966, Step 6: Misc initializations: 1e-06 seconds
Process-1966, Step 7: for loop: 0.006291 seconds

Process-1966All steps time: 0.148635 seconds

Process-1966, Find cars processing time: 0.148716 seconds

The times for each task are: [0.280056, 0.272467, 0.200167, 0.254252, 0.152564, 0.194328, 0.205000]

Minimum: 0.110968 Maximum: 0.280056 Average: 0.1827 seconds

Number of processes used: 2 window size 260
Length of task list: 13
Number of processes used: 2

Process-1967, Step 1: Divide with 255, processing time: 0.031838 seconds
Process-1967, Step 2: Resize if scale is not 1: 0.002338 seconds
Process-1967, Step 3: Get HOG channels: 2.8e-05 seconds
Process-1967, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1967, Step 5: Compute individual channel HOG features for the entire image: 0.2527 seconds
Process-1967, Step 6: Misc initializations: 1e-06 seconds
Process-1967, Step 7: for loop: 0.017669 seconds

Process-1967All steps time: 0.3045809999999994 seconds

Process-1967, Find cars processing time: 0.304811 seconds

Process-1968, Step 1: Divide with 255, processing time: 0.028835 seconds
Process-1968, Step 2: Resize if scale is not 1: 0.002141 seconds
Process-1968, Step 3: Get HOG channels: 3.9e-05 seconds
Process-1968, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1968, Step 5: Compute individual channel HOG features for the entire image: 0.206718 seconds
Process-1968, Step 6: Misc initializations: 0.0 seconds
Process-1968, Step 7: for loop: 0.013727 seconds

Process-1968All steps time: 0.251466 seconds

Process-1968, Find cars processing time: 0.251642 seconds

Process-1968, Step 1: Divide with 255, processing time: 0.009548 seconds
Process-1968, Step 2: Resize if scale is not 1: 0.00083 seconds
Process-1968, Step 3: Get HOG channels: 8e-06 seconds

Process-1968, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1968, Step 5: Compute individual channel HOG features for the entire image: 0.230577 seconds
Process-1968, Step 6: Misc initializations: 1e-06 seconds
Process-1968, Step 7: for loop: 0.01296 seconds

Process-1968All steps time: 0.25393299999999996 seconds

Process-1968, Find cars processing time: 0.254014 seconds

Process-1967, Step 1: Divide with 255, processing time: 0.010498 seconds
Process-1967, Step 2: Resize if scale is not 1: 0.000896 seconds
Process-1967, Step 3: Get HOG channels: 8e-06 seconds
Process-1967, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1967, Step 5: Compute individual channel HOG features for the entire image: 0.227719 seconds
Process-1967, Step 6: Misc initializations: 1e-06 seconds
Process-1967, Step 7: for loop: 0.012386 seconds

Process-1967All steps time: 0.251516 seconds

Process-1967, Find cars processing time: 0.251585 seconds

Process-1968, Step 1: Divide with 255, processing time: 0.010239 seconds
Process-1968, Step 2: Resize if scale is not 1: 0.000928 seconds
Process-1968, Step 3: Get HOG channels: 6e-06 seconds
Process-1968, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1968, Step 5: Compute individual channel HOG features for the entire image: 0.170498 seconds
Process-1968, Step 6: Misc initializations: 1e-06 seconds
Process-1968, Step 7: for loop: 0.009538 seconds

Process-1968All steps time: 0.191216 seconds

Process-1968, Find cars processing time: 0.191282 seconds

Process-1967, Step 1: Divide with 255, processing time: 0.009113 seconds
Process-1967, Step 2: Resize if scale is not 1: 0.000784 seconds
Process-1967, Step 3: Get HOG channels: 8e-06 seconds
Process-1967, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1967, Step 5: Compute individual channel HOG features for the entire image: 0.147429 seconds
Process-1967, Step 6: Misc initializations: 0.0 seconds
Process-1967, Step 7: for loop: 0.007587 seconds

Process-1967All steps time: 0.16492900000000002 seconds

Process-1967, Find cars processing time: 0.165009 seconds

Process-1968, Step 1: Divide with 255, processing time: 0.009353 seconds
Process-1968, Step 2: Resize if scale is not 1: 0.00071 seconds
Process-1968, Step 3: Get HOG channels: 8e-06 seconds

Process-1968, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1968, Step 5: Compute individual channel HOG features for the entire image: 0.146601 seconds

Process-1968, Step 6: Misc initializations: 0.0 seconds

Process-1968, Step 7: for loop: 0.007713 seconds

Process-1968All steps time: 0.164393 seconds

Process-1968, Find cars processing time: 0.164459 seconds

Process-1967, Step 1: Divide with 255, processing time: 0.011 seconds

Process-1967, Step 2: Resize if scale is not 1: 0.00087 seconds

Process-1967, Step 3: Get HOG channels: 8e-06 seconds

Process-1967, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1967, Step 5: Compute individual channel HOG features for the entire image: 0.167091 seconds

Process-1967, Step 6: Misc initializations: 1e-06 seconds

Process-1967, Step 7: for loop: 0.009711 seconds

Process-1967All steps time: 0.188689 seconds

Process-1967, Find cars processing time: 0.188774 seconds

Process-1967, Step 1: Divide with 255, processing time: 0.010283 seconds

Process-1967, Step 2: Resize if scale is not 1: 0.001379 seconds

Process-1967, Step 3: Get HOG channels: 9e-06 seconds

Process-1967, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1967, Step 5: Compute individual channel HOG features for the entire image: 0.127805 seconds

Process-1967, Step 6: Misc initializations: 0.0 seconds

Process-1967, Step 7: for loop: 0.007218 seconds

Process-1967All steps time: 0.146703 seconds

Process-1967, Find cars processing time: 0.146775 seconds

Process-1968, Step 1: Divide with 255, processing time: 0.01032 seconds

Process-1968, Step 2: Resize if scale is not 1: 0.001113 seconds

Process-1968, Step 3: Get HOG channels: 7e-06 seconds

Process-1968, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1968, Step 5: Compute individual channel HOG features for the entire image: 0.105221 seconds

Process-1968, Step 6: Misc initializations: 1e-06 seconds

Process-1968, Step 7: for loop: 0.006202 seconds

Process-1968All steps time: 0.122872 seconds

Process-1968, Find cars processing time: 0.122955 seconds

Process-1967, Step 1: Divide with 255, processing time: 0.010421 seconds

Process-1967, Step 2: Resize if scale is not 1: 0.00051 seconds

Process-1967, Step 3: Get HOG channels: 5e-06 seconds

Process-1967, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1967, Step 5: Compute individual channel HOG features for the entire image: 0.139445 seconds
Process-1967, Step 6: Misc initializations: 0.0 seconds
Process-1967, Step 7: for loop: 0.007336 seconds

Process-1967All steps time: 0.15772200000000003 seconds

Process-1967, Find cars processing time: 0.157786 seconds

Process-1967, Step 1: Divide with 255, processing time: 0.010486 seconds
Process-1967, Step 2: Resize if scale is not 1: 0.001723 seconds
Process-1967, Step 3: Get HOG channels: 9e-06 seconds
Process-1967, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1967, Step 5: Compute individual channel HOG features for the entire image: 0.113627 seconds
Process-1967, Step 6: Misc initializations: 1e-06 seconds
Process-1967, Step 7: for loop: 0.006196 seconds

Process-1967All steps time: 0.13205100000000003 seconds

Process-1967, Find cars processing time: 0.132133 seconds

Process-1968, Step 1: Divide with 255, processing time: 0.009639 seconds
Process-1968, Step 2: Resize if scale is not 1: 0.000498 seconds
Process-1968, Step 3: Get HOG channels: 6e-06 seconds
Process-1968, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1968, Step 5: Compute individual channel HOG features for the entire image: 0.126302 seconds
Process-1968, Step 6: Misc initializations: 0.0 seconds
Process-1968, Step 7: for loop: 0.007329 seconds

Process-1968All steps time: 0.14378 seconds

Process-1968, Find cars processing time: 0.143837 seconds

The times for each task are: [0.304811, 0.251642, 0.254014, 0.251585, 0.191282, 0.165009, 0.1644

Minimum: 0.122955 Maximum: 0.304811 Average: 0.1904 seconds

Window sizes used: [1.15, 1.35, 1.55]
2 processes used for testing 3 window sizes
Processing times for each image [3.5017, 3.5796, 3.3636, 3.504, 4.1611, 3.8529] with an average

Time elapsed so far... 74.4425
#####

Number of processes used: 3 window size 260
Length of task list: 13
Number of processes used: 3

Process-1969, Step 1: Divide with 255, processing time: 0.029802 seconds
Process-1969, Step 2: Resize if scale is not 1: 0.002139 seconds
Process-1969, Step 3: Get HOG channels: 2.7e-05 seconds
Process-1969, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1969, Step 5: Compute individual channel HOG features for the entire image: 0.249145 seconds
Process-1969, Step 6: Misc initializations: 1e-06 seconds
Process-1969, Step 7: for loop: 0.017445 seconds

Process-1969All steps time: 0.29856499999999997 seconds

Process-1969, Find cars processing time: 0.298754 seconds

Process-1970, Step 1: Divide with 255, processing time: 0.02795 seconds
Process-1970, Step 2: Resize if scale is not 1: 0.00205 seconds
Process-1970, Step 3: Get HOG channels: 2.3e-05 seconds
Process-1970, Step 4: Define blocks and steps as above: 1.6e-05 seconds
Process-1970, Step 5: Compute individual channel HOG features for the entire image: 0.240637 seconds
Process-1970, Step 6: Misc initializations: 1e-06 seconds
Process-1970, Step 7: for loop: 0.017837 seconds

Process-1970All steps time: 0.2885139999999994 seconds

Process-1970, Find cars processing time: 0.288663 seconds

Process-1971, Step 1: Divide with 255, processing time: 0.028896 seconds
Process-1971, Step 2: Resize if scale is not 1: 0.002077 seconds
Process-1971, Step 3: Get HOG channels: 2.1e-05 seconds
Process-1971, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1971, Step 5: Compute individual channel HOG features for the entire image: 0.274558 seconds
Process-1971, Step 6: Misc initializations: 0.0 seconds
Process-1971, Step 7: for loop: 0.014626 seconds

Process-1971All steps time: 0.320184 seconds

Process-1971, Find cars processing time: 0.320355 seconds

Process-1969, Step 1: Divide with 255, processing time: 0.009673 seconds
Process-1969, Step 2: Resize if scale is not 1: 0.000576 seconds
Process-1969, Step 3: Get HOG channels: 5e-06 seconds
Process-1969, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1969, Step 5: Compute individual channel HOG features for the entire image: 0.184361 seconds
Process-1969, Step 6: Misc initializations: 1e-06 seconds
Process-1969, Step 7: for loop: 0.013092 seconds

Process-1969All steps time: 0.2077139999999998 seconds

Process-1969, Find cars processing time: 0.207777 seconds

Process-1970, Step 1: Divide with 255, processing time: 0.010155 seconds
Process-1970, Step 2: Resize if scale is not 1: 0.000602 seconds
Process-1970, Step 3: Get HOG channels: 5e-06 seconds
Process-1970, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1970, Step 5: Compute individual channel HOG features for the entire image: 0.170235 seconds
Process-1970, Step 6: Misc initializations: 0.0 seconds
Process-1970, Step 7: for loop: 0.007359 seconds

Process-1970All steps time: 0.188361 seconds

Process-1970, Find cars processing time: 0.188424 seconds

Process-1970, Step 1: Divide with 255, processing time: 0.010289 seconds
Process-1970, Step 2: Resize if scale is not 1: 0.000895 seconds
Process-1970, Step 3: Get HOG channels: 9e-06 seconds
Process-1970, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1970, Step 5: Compute individual channel HOG features for the entire image: 0.127169 seconds
Process-1970, Step 6: Misc initializations: 0.0 seconds
Process-1970, Step 7: for loop: 0.00801 seconds

Process-1970All steps time: 0.1463809999999998 seconds

Process-1970, Find cars processing time: 0.146446 seconds

Process-1969, Step 1: Divide with 255, processing time: 0.011106 seconds
Process-1971, Step 1: Divide with 255, processing time: 0.010609 seconds
Process-1969, Step 2: Resize if scale is not 1: 0.000975 seconds
Process-1971, Step 2: Resize if scale is not 1: 0.000548 seconds
Process-1969, Step 3: Get HOG channels: 8e-06 seconds
Process-1971, Step 3: Get HOG channels: 5e-06 seconds
Process-1969, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1971, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1971, Step 5: Compute individual channel HOG features for the entire image: 0.13592 seconds
Process-1969, Step 5: Compute individual channel HOG features for the entire image: 0.139823 seconds
Process-1971, Step 6: Misc initializations: 0.0 seconds
Process-1969, Step 6: Misc initializations: 0.0 seconds
Process-1971, Step 7: for loop: 0.007532 seconds
Process-1969, Step 7: for loop: 0.007539 seconds

Process-1971All steps time: 0.1546190000000003 seconds

Process-1971, Find cars processing time: 0.154668 seconds

Process-1969All steps time: 0.159459 seconds

Process-1969, Find cars processing time: 0.159529 seconds

Process-1971, Step 1: Divide with 255, processing time: 0.010337 seconds
Process-1971, Step 2: Resize if scale is not 1: 0.001386 seconds
Process-1971, Step 3: Get HOG channels: 9e-06 seconds
Process-1971, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1971, Step 5: Compute individual channel HOG features for the entire image: 0.098243 seconds
Process-1971, Step 6: Misc initializations: 0.0 seconds
Process-1971, Step 7: for loop: 0.005934 seconds

Process-1971All steps time: 0.115919 seconds

Process-1971, Find cars processing time: 0.115993 seconds

Process-1969, Step 1: Divide with 255, processing time: 0.01064 seconds
Process-1969, Step 2: Resize if scale is not 1: 0.001339 seconds
Process-1969, Step 3: Get HOG channels: 6e-06 seconds
Process-1969, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1969, Step 5: Compute individual channel HOG features for the entire image: 0.140714 seconds
Process-1969, Step 6: Misc initializations: 1e-06 seconds
Process-1969, Step 7: for loop: 0.007745 seconds

Process-1969All steps time: 0.16045 seconds

Process-1969, Find cars processing time: 0.160521 seconds

Process-1970, Step 1: Divide with 255, processing time: 0.010191 seconds
Process-1970, Step 2: Resize if scale is not 1: 0.000766 seconds
Process-1970, Step 3: Get HOG channels: 4e-06 seconds
Process-1970, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1970, Step 5: Compute individual channel HOG features for the entire image: 0.109774 seconds
Process-1970, Step 6: Misc initializations: 1e-06 seconds
Process-1970, Step 7: for loop: 0.00674 seconds

Process-1970All steps time: 0.127481 seconds

Process-1970, Find cars processing time: 0.127569 seconds

Process-1971, Step 1: Divide with 255, processing time: 0.010756 seconds
Process-1971, Step 2: Resize if scale is not 1: 0.000776 seconds
Process-1971, Step 3: Get HOG channels: 8e-06 seconds
Process-1971, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1971, Step 5: Compute individual channel HOG features for the entire image: 0.104302 seconds
Process-1971, Step 6: Misc initializations: 0.0 seconds
Process-1971, Step 7: for loop: 0.005704 seconds

Process-1971All steps time: 0.12155400000000001 seconds

Process-1971, Find cars processing time: 0.121624 seconds

Process-1969, Step 1: Divide with 255, processing time: 0.010149 seconds
Process-1969, Step 2: Resize if scale is not 1: 0.001205 seconds
Process-1969, Step 3: Get HOG channels: 6e-06 seconds
Process-1969, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1969, Step 5: Compute individual channel HOG features for the entire image: 0.109319 seconds
Process-1969, Step 6: Misc initializations: 1e-06 seconds
Process-1969, Step 7: for loop: 0.00609 seconds

Process-1969All steps time: 0.126776 seconds

Process-1969, Find cars processing time: 0.126835 seconds

The times for each task are: [0.298754, 0.288663, 0.320355, 0.207777, 0.188424, 0.146446, 0.1546

Minimum: 0.115993 Maximum: 0.320355 Average: 0.1859 seconds

Number of processes used: 3 window size 260
Length of task list: 13
Number of processes used: 3

Process-1972, Step 1: Divide with 255, processing time: 0.030632 seconds
Process-1972, Step 2: Resize if scale is not 1: 0.002097 seconds
Process-1972, Step 3: Get HOG channels: 2.4e-05 seconds
Process-1972, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1972, Step 5: Compute individual channel HOG features for the entire image: 0.244706 seconds
Process-1972, Step 6: Misc initializations: 1e-06 seconds
Process-1972, Step 7: for loop: 0.017478 seconds

Process-1972All steps time: 0.29494499999999996 seconds

Process-1972, Find cars processing time: 0.295144 seconds

Process-1973, Step 1: Divide with 255, processing time: 0.042546 seconds
Process-1973, Step 2: Resize if scale is not 1: 0.002536 seconds
Process-1973, Step 3: Get HOG channels: 4.1e-05 seconds
Process-1973, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1973, Step 5: Compute individual channel HOG features for the entire image: 0.247541 seconds
Process-1973, Step 6: Misc initializations: 1e-06 seconds
Process-1973, Step 7: for loop: 0.017897 seconds

Process-1973All steps time: 0.31056799999999996 seconds

Process-1973, Find cars processing time: 0.310728 seconds

Process-1974, Step 1: Divide with 255, processing time: 0.041928 seconds
Process-1974, Step 2: Resize if scale is not 1: 0.002977 seconds

Process-1974, Step 3: Get HOG channels: 1.9e-05 seconds
Process-1974, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1974, Step 5: Compute individual channel HOG features for the entire image: 0.247441 seconds
Process-1974, Step 6: Misc initializations: 1e-06 seconds
Process-1974, Step 7: for loop: 0.017386 seconds

Process-1974All steps time: 0.309757 seconds

Process-1974, Find cars processing time: 0.309936 seconds

Process-1972, Step 1: Divide with 255, processing time: 0.010682 seconds
Process-1972, Step 2: Resize if scale is not 1: 0.0008 seconds
Process-1972, Step 3: Get HOG channels: 7e-06 seconds
Process-1972, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1972, Step 5: Compute individual channel HOG features for the entire image: 0.245211 seconds
Process-1972, Step 6: Misc initializations: 1e-06 seconds
Process-1972, Step 7: for loop: 0.016808 seconds

Process-1972All steps time: 0.273516 seconds

Process-1972, Find cars processing time: 0.273603 seconds

Process-1973, Step 1: Divide with 255, processing time: 0.010638 seconds
Process-1973, Step 2: Resize if scale is not 1: 0.000958 seconds
Process-1973, Step 3: Get HOG channels: 8e-06 seconds
Process-1973, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1973, Step 5: Compute individual channel HOG features for the entire image: 0.171878 seconds
Process-1973, Step 6: Misc initializations: 0.0 seconds
Process-1973, Step 7: for loop: 0.009589 seconds

Process-1973All steps time: 0.1930800000000003 seconds

Process-1973, Find cars processing time: 0.193157 seconds

Process-1974, Step 1: Divide with 255, processing time: 0.009624 seconds
Process-1974, Step 2: Resize if scale is not 1: 0.000879 seconds
Process-1974, Step 3: Get HOG channels: 8e-06 seconds
Process-1974, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1974, Step 5: Compute individual channel HOG features for the entire image: 0.172984 seconds
Process-1974, Step 6: Misc initializations: 5e-06 seconds
Process-1974, Step 7: for loop: 0.00994 seconds

Process-1974All steps time: 0.193448 seconds

Process-1974, Find cars processing time: 0.193517 seconds

Process-1972, Step 1: Divide with 255, processing time: 0.010559 seconds
Process-1972, Step 2: Resize if scale is not 1: 0.000623 seconds

Process-1972, Step 3: Get HOG channels: 6e-06 seconds
Process-1972, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1972, Step 5: Compute individual channel HOG features for the entire image: 0.175726 seconds
Process-1972, Step 6: Misc initializations: 1e-06 seconds
Process-1972, Step 7: for loop: 0.012282 seconds

Process-1972All steps time: 0.199202 seconds

Process-1972, Find cars processing time: 0.199269 seconds

Process-1974, Step 1: Divide with 255, processing time: 0.010617 seconds
Process-1974, Step 2: Resize if scale is not 1: 0.001453 seconds
Process-1974, Step 3: Get HOG channels: 1e-05 seconds
Process-1974, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1974, Step 5: Compute individual channel HOG features for the entire image: 0.103282 seconds
Process-1974, Step 6: Misc initializations: 0.0 seconds
Process-1974, Step 7: for loop: 0.005723 seconds

Process-1974All steps time: 0.121093 seconds

Process-1974, Find cars processing time: 0.121116 seconds

Process-1973, Step 1: Divide with 255, processing time: 0.010324 seconds
Process-1973, Step 2: Resize if scale is not 1: 0.000923 seconds
Process-1973, Step 3: Get HOG channels: 1e-05 seconds
Process-1973, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1973, Step 5: Compute individual channel HOG features for the entire image: 0.172461 seconds
Process-1973, Step 6: Misc initializations: 1e-06 seconds
Process-1973, Step 7: for loop: 0.009562 seconds

Process-1973All steps time: 0.19329 seconds

Process-1973, Find cars processing time: 0.193379 seconds

Process-1974, Step 1: Divide with 255, processing time: 0.008318 seconds
Process-1974, Step 2: Resize if scale is not 1: 0.000561 seconds
Process-1974, Step 3: Get HOG channels: 7e-06 seconds
Process-1974, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1974, Step 5: Compute individual channel HOG features for the entire image: 0.117463 seconds
Process-1974, Step 6: Misc initializations: 0.0 seconds
Process-1974, Step 7: for loop: 0.006045 seconds

Process-1974All steps time: 0.1324 seconds

Process-1974, Find cars processing time: 0.132464 seconds

Process-1973, Step 1: Divide with 255, processing time: 0.006906 seconds
Process-1973, Step 2: Resize if scale is not 1: 0.001062 seconds

Process-1973, Step 3: Get HOG channels: 7e-06 seconds
Process-1973, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1973, Step 5: Compute individual channel HOG features for the entire image: 0.131431 seconds
Process-1973, Step 6: Misc initializations: 1e-06 seconds
Process-1973, Step 7: for loop: 0.007168 seconds

Process-1973All steps time: 0.146583 seconds

Process-1973, Find cars processing time: 0.146658 seconds

Process-1972, Step 1: Divide with 255, processing time: 0.022497 seconds
Process-1972, Step 2: Resize if scale is not 1: 0.000864 seconds
Process-1972, Step 3: Get HOG channels: 6e-06 seconds
Process-1972, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1972, Step 5: Compute individual channel HOG features for the entire image: 0.130881 seconds
Process-1972, Step 6: Misc initializations: 1e-06 seconds
Process-1972, Step 7: for loop: 0.008041 seconds

Process-1972All steps time: 0.162295 seconds

Process-1972, Find cars processing time: 0.162369 seconds

Process-1974, Step 1: Divide with 255, processing time: 0.007606 seconds
Process-1974, Step 2: Resize if scale is not 1: 0.00187 seconds
Process-1974, Step 3: Get HOG channels: 9e-06 seconds
Process-1974, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1974, Step 5: Compute individual channel HOG features for the entire image: 0.113818 seconds
Process-1974, Step 6: Misc initializations: 1e-06 seconds
Process-1974, Step 7: for loop: 0.006272 seconds

Process-1974All steps time: 0.129584 seconds

Process-1974, Find cars processing time: 0.129658 seconds

The times for each task are: [0.295144, 0.310728, 0.309936, 0.273603, 0.193157, 0.193517, 0.199211]

Minimum: 0.12116 Maximum: 0.310728 Average: 0.2047 seconds

Number of processes used: 3 window size 260
Length of task list: 13
Number of processes used: 3

Process-1975, Step 1: Divide with 255, processing time: 0.027148 seconds
Process-1976, Step 1: Divide with 255, processing time: 0.020738 seconds
Process-1975, Step 2: Resize if scale is not 1: 0.002678 seconds
Process-1976, Step 2: Resize if scale is not 1: 0.001801 seconds
Process-1975, Step 3: Get HOG channels: 1.8e-05 seconds

Process-1976, Step 3: Get HOG channels: 1.7e-05 seconds
Process-1975, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1976, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1975, Step 5: Compute individual channel HOG features for the entire image: 0.256093 seconds
Process-1976, Step 5: Compute individual channel HOG features for the entire image: 0.202621 seconds
Process-1975, Step 6: Misc initializations: 1e-06 seconds
Process-1976, Step 6: Misc initializations: 1e-06 seconds
Process-1975, Step 7: for loop: 0.016366 seconds
Process-1976, Step 7: for loop: 0.017254 seconds

Process-1975All steps time: 0.30230999999999997 seconds
Process-1976All steps time: 0.2424369999999999 seconds

Process-1975, Find cars processing time: 0.302511 seconds

Process-1976, Find cars processing time: 0.242589 seconds

Process-1976, Step 1: Divide with 255, processing time: 0.009216 seconds
Process-1976, Step 2: Resize if scale is not 1: 0.000801 seconds
Process-1976, Step 3: Get HOG channels: 8e-06 seconds
Process-1976, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1976, Step 5: Compute individual channel HOG features for the entire image: 0.153351 seconds
Process-1976, Step 6: Misc initializations: 1e-06 seconds
Process-1976, Step 7: for loop: 0.008119 seconds

Process-1976All steps time: 0.171502 seconds

Process-1976, Find cars processing time: 0.171563 seconds

Process-1977, Step 1: Divide with 255, processing time: 0.024752 seconds
Process-1977, Step 2: Resize if scale is not 1: 0.002177 seconds
Process-1977, Step 3: Get HOG channels: 1.8e-05 seconds
Process-1977, Step 4: Define blocks and steps as above: 1.6e-05 seconds
Process-1977, Step 5: Compute individual channel HOG features for the entire image: 0.248108 seconds
Process-1977, Step 6: Misc initializations: 1e-06 seconds
Process-1977, Step 7: for loop: 0.015377 seconds

Process-1977All steps time: 0.2904489999999996 seconds

Process-1977, Find cars processing time: 0.290612 seconds
Process-1975, Step 1: Divide with 255, processing time: 0.008811 seconds

Process-1975, Step 2: Resize if scale is not 1: 0.000591 seconds
Process-1975, Step 3: Get HOG channels: 5e-06 seconds
Process-1975, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1975, Step 5: Compute individual channel HOG features for the entire image: 0.236979 seconds

Process-1975, Step 6: Misc initializations: 1e-06 seconds
Process-1975, Step 7: for loop: 0.01645 seconds

Process-1975All steps time: 0.262842 seconds

Process-1975, Find cars processing time: 0.2629 seconds

Process-1976, Step 1: Divide with 255, processing time: 0.010663 seconds
Process-1976, Step 2: Resize if scale is not 1: 0.000491 seconds
Process-1976, Step 3: Get HOG channels: 5e-06 seconds
Process-1976, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1976, Step 5: Compute individual channel HOG features for the entire image: 0.146464 seconds
Process-1976, Step 6: Misc initializations: 0.0 seconds
Process-1976, Step 7: for loop: 0.007704 seconds

Process-1976All steps time: 0.165332 seconds

Process-1976, Find cars processing time: 0.165384 seconds

Process-1976, Step 1: Divide with 255, processing time: 0.010829 seconds
Process-1976, Step 2: Resize if scale is not 1: 0.000833 seconds
Process-1976, Step 3: Get HOG channels: 6e-06 seconds
Process-1976, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1976, Step 5: Compute individual channel HOG features for the entire image: 0.106931 seconds
Process-1976, Step 6: Misc initializations: 1e-06 seconds
Process-1976, Step 7: for loop: 0.005773 seconds

Process-1976All steps time: 0.124378 seconds

Process-1976, Find cars processing time: 0.124435 seconds

Process-1977, Step 1: Divide with 255, processing time: 0.010548 seconds
Process-1977, Step 2: Resize if scale is not 1: 0.000488 seconds
Process-1977, Step 3: Get HOG channels: 5e-06 seconds
Process-1977, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1977, Step 5: Compute individual channel HOG features for the entire image: 0.146537 seconds
Process-1977, Step 6: Misc initializations: 1e-06 seconds
Process-1977, Step 7: for loop: 0.008484 seconds

Process-1977All steps time: 0.166068 seconds

Process-1977, Find cars processing time: 0.166126 seconds

Process-1975, Step 1: Divide with 255, processing time: 0.007456 seconds
Process-1975, Step 2: Resize if scale is not 1: 0.00059 seconds
Process-1975, Step 3: Get HOG channels: 5e-06 seconds
Process-1975, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1975, Step 5: Compute individual channel HOG features for the entire image: 0.17063 seconds

Process-1975, Step 6: Misc initializations: 0.0 seconds
Process-1975, Step 7: for loop: 0.009346 seconds

Process-1975All steps time: 0.188033 seconds

Process-1975, Find cars processing time: 0.188112 seconds

Process-1977, Step 1: Divide with 255, processing time: 0.009627 seconds
Process-1977, Step 2: Resize if scale is not 1: 0.000882 seconds
Process-1977, Step 3: Get HOG channels: 6e-06 seconds
Process-1977, Step 4: Define blocks and steps as above: 1.4e-05 seconds
Process-1977, Step 5: Compute individual channel HOG features for the entire image: 0.10585 seconds
Process-1977, Step 6: Misc initializations: 1e-06 seconds
Process-1977, Step 7: for loop: 0.005961 seconds

Process-1977All steps time: 0.1223409999999999 seconds

Process-1977, Find cars processing time: 0.122408 seconds

Process-1975, Step 1: Divide with 255, processing time: 0.009852 seconds
Process-1975, Step 2: Resize if scale is not 1: 0.000913 seconds
Process-1975, Step 3: Get HOG channels: 5e-06 seconds
Process-1975, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1975, Step 5: Compute individual channel HOG features for the entire image: 0.127673 seconds
Process-1975, Step 6: Misc initializations: 1e-06 seconds
Process-1975, Step 7: for loop: 0.007149 seconds

Process-1975All steps time: 0.145598 seconds

Process-1975, Find cars processing time: 0.14566 seconds

Process-1976, Step 1: Divide with 255, processing time: 0.010643 seconds
Process-1976, Step 2: Resize if scale is not 1: 0.000682 seconds
Process-1976, Step 3: Get HOG channels: 5e-06 seconds
Process-1976, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1976, Step 5: Compute individual channel HOG features for the entire image: 0.126594 seconds
Process-1976, Step 6: Misc initializations: 1e-06 seconds
Process-1976, Step 7: for loop: 0.008388 seconds

Process-1976All steps time: 0.1463180000000003 seconds

Process-1976, Find cars processing time: 0.146383 seconds

Process-1977, Step 1: Divide with 255, processing time: 0.010905 seconds
Process-1977, Step 2: Resize if scale is not 1: 0.001967 seconds
Process-1977, Step 3: Get HOG channels: 9e-06 seconds
Process-1977, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1977, Step 5: Compute individual channel HOG features for the entire image: 0.087148 seconds

Process-1977, Step 6: Misc initializations: 0.0 seconds
Process-1977, Step 7: for loop: 0.004846 seconds

Process-1977All steps time: 0.104884 seconds

Process-1977, Find cars processing time: 0.104963 seconds

The times for each task are: [0.302511, 0.242589, 0.171563, 0.290612, 0.2629, 0.165384, 0.124435]

Minimum: 0.104963 Maximum: 0.302511 Average: 0.1872 seconds

Number of processes used: 3 window size 260

Length of task list: 13

Number of processes used: 3

Process-1978, Step 1: Divide with 255, processing time: 0.02906 seconds

Process-1978, Step 2: Resize if scale is not 1: 0.002031 seconds

Process-1978, Step 3: Get HOG channels: 1.8e-05 seconds

Process-1978, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1978, Step 5: Compute individual channel HOG features for the entire image: 0.271732 seconds

Process-1978, Step 6: Misc initializations: 1e-06 seconds

Process-1978, Step 7: for loop: 0.01926 seconds

Process-1978All steps time: 0.3221079999999995 seconds

Process-1978, Find cars processing time: 0.322303 seconds

Process-1979, Step 1: Divide with 255, processing time: 0.024208 seconds

Process-1979, Step 2: Resize if scale is not 1: 0.001681 seconds

Process-1979, Step 3: Get HOG channels: 1.7e-05 seconds

Process-1979, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1979, Step 5: Compute individual channel HOG features for the entire image: 0.20163 seconds

Process-1979, Step 6: Misc initializations: 1e-06 seconds

Process-1979, Step 7: for loop: 0.01513 seconds

Process-1979All steps time: 0.242673 seconds

Process-1979, Find cars processing time: 0.242806 seconds

Process-1980, Step 1: Divide with 255, processing time: 0.031555 seconds

Process-1980, Step 2: Resize if scale is not 1: 0.002295 seconds

Process-1980, Step 3: Get HOG channels: 2.2e-05 seconds

Process-1980, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1980, Step 5: Compute individual channel HOG features for the entire image: 0.234471 seconds

Process-1980, Step 6: Misc initializations: 1e-06 seconds

Process-1980, Step 7: for loop: 0.01465 seconds

Process-1980All steps time: 0.283002 seconds

Process-1980, Find cars processing time: 0.283176 seconds

Process-1978, Step 1: Divide with 255, processing time: 0.010196 seconds

Process-1978, Step 2: Resize if scale is not 1: 0.000899 seconds

Process-1978, Step 3: Get HOG channels: 8e-06 seconds

Process-1978, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1978, Step 5: Compute individual channel HOG features for the entire image: 0.203895 seconds

Process-1978, Step 6: Misc initializations: 1e-06 seconds

Process-1978, Step 7: for loop: 0.013874 seconds

Process-1978All steps time: 0.228881 seconds

Process-1978, Find cars processing time: 0.228952 seconds

Process-1979, Step 1: Divide with 255, processing time: 0.010017 seconds

Process-1979, Step 2: Resize if scale is not 1: 0.000913 seconds

Process-1979, Step 3: Get HOG channels: 7e-06 seconds

Process-1979, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1979, Step 5: Compute individual channel HOG features for the entire image: 0.173301 seconds

Process-1979, Step 6: Misc initializations: 1e-06 seconds

Process-1979, Step 7: for loop: 0.010007 seconds

Process-1979All steps time: 0.194252 seconds

Process-1979, Find cars processing time: 0.194319 seconds

Process-1980, Step 1: Divide with 255, processing time: 0.010533 seconds

Process-1980, Step 2: Resize if scale is not 1: 0.00085 seconds

Process-1980, Step 3: Get HOG channels: 8e-06 seconds

Process-1980, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1980, Step 5: Compute individual channel HOG features for the entire image: 0.1516 seconds

Process-1980, Step 6: Misc initializations: 1e-06 seconds

Process-1980, Step 7: for loop: 0.008154 seconds

Process-1980All steps time: 0.171155 seconds

Process-1980, Find cars processing time: 0.171227 seconds

Process-1978, Step 1: Divide with 255, processing time: 0.009779 seconds

Process-1978, Step 2: Resize if scale is not 1: 0.000526 seconds

Process-1978, Step 3: Get HOG channels: 5e-06 seconds

Process-1978, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1978, Step 5: Compute individual channel HOG features for the entire image: 0.174444 seconds

Process-1978, Step 6: Misc initializations: 1e-06 seconds

Process-1978, Step 7: for loop: 0.009807 seconds

Process-1978All steps time: 0.194567 seconds

Process-1978, Find cars processing time: 0.194626 seconds

Process-1979, Step 1: Divide with 255, processing time: 0.011073 seconds

Process-1979, Step 2: Resize if scale is not 1: 0.000772 seconds

Process-1979, Step 3: Get HOG channels: 5e-06 seconds

Process-1979, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1979, Step 5: Compute individual channel HOG features for the entire image: 0.16905 seconds

Process-1979, Step 6: Misc initializations: 1e-06 seconds

Process-1979, Step 7: for loop: 0.009931 seconds

Process-1979All steps time: 0.190837 seconds

Process-1979, Find cars processing time: 0.190907 seconds

Process-1980, Step 1: Divide with 255, processing time: 0.010258 seconds

Process-1980, Step 2: Resize if scale is not 1: 0.001325 seconds

Process-1980, Step 3: Get HOG channels: 8e-06 seconds

Process-1980, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1980, Step 5: Compute individual channel HOG features for the entire image: 0.109448 seconds

Process-1980, Step 6: Misc initializations: 0.0 seconds

Process-1980, Step 7: for loop: 0.00583 seconds

Process-1979, Step 1: Divide with 255, processing time: 0.010425 seconds

Process-1979, Step 2: Resize if scale is not 1: 0.001303 seconds

Process-1980All steps time: 0.126877 seconds

Process-1979, Step 3: Get HOG channels: 8e-06 seconds

Process-1979, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1980, Find cars processing time: 0.126954 seconds

Process-1979, Step 5: Compute individual channel HOG features for the entire image: 0.134341 seconds

Process-1979, Step 6: Misc initializations: 1e-06 seconds

Process-1979, Step 7: for loop: 0.007411 seconds

Process-1979All steps time: 0.153496 seconds

Process-1979, Find cars processing time: 0.153577 seconds

Process-1978, Step 1: Divide with 255, processing time: 0.016259 seconds

Process-1978, Step 2: Resize if scale is not 1: 0.000921 seconds

Process-1978, Step 3: Get HOG channels: 6e-06 seconds

Process-1978, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1978, Step 5: Compute individual channel HOG features for the entire image: 0.129752 seconds

Process-1978, Step 6: Misc initializations: 1e-06 seconds

Process-1978, Step 7: for loop: 0.00738 seconds

Process-1978All steps time: 0.15432400000000002 seconds

Process-1978, Find cars processing time: 0.154414 seconds

Process-1980, Step 1: Divide with 255, processing time: 0.009902 seconds

Process-1980, Step 2: Resize if scale is not 1: 0.000692 seconds

Process-1980, Step 3: Get HOG channels: 7e-06 seconds

Process-1980, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1980, Step 5: Compute individual channel HOG features for the entire image: 0.114144 sec

Process-1980, Step 6: Misc initializations: 1e-06 seconds

Process-1980, Step 7: for loop: 0.006335 seconds

Process-1980All steps time: 0.1310889999999998 seconds

Process-1980, Find cars processing time: 0.131116 seconds

Process-1979, Step 1: Divide with 255, processing time: 0.010235 seconds

Process-1979, Step 2: Resize if scale is not 1: 0.00159 seconds

Process-1979, Step 3: Get HOG channels: 7e-06 seconds

Process-1979, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1979, Step 5: Compute individual channel HOG features for the entire image: 0.133314 sec

Process-1979, Step 6: Misc initializations: 1e-06 seconds

Process-1979, Step 7: for loop: 0.006194 seconds

Process-1979All steps time: 0.1513489999999998 seconds

Process-1979, Find cars processing time: 0.151434 seconds

The times for each task are: [0.322303, 0.242806, 0.283176, 0.228952, 0.194319, 0.171227, 0.1946

Minimum: 0.126954 Maximum: 0.322303 Average: 0.1958 seconds

Number of processes used: 3 window size 260

Length of task list: 13

Number of processes used: 3

Process-1981, Step 1: Divide with 255, processing time: 0.027399 seconds

Process-1981, Step 2: Resize if scale is not 1: 0.002364 seconds

Process-1981, Step 3: Get HOG channels: 3e-05 seconds

Process-1981, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1981, Step 5: Compute individual channel HOG features for the entire image: 0.24854 sec

Process-1981, Step 6: Misc initializations: 1e-06 seconds

Process-1981, Step 7: for loop: 0.016841 seconds

Process-1981All steps time: 0.29518 seconds

Process-1981, Find cars processing time: 0.295351 seconds

Process-1982, Step 1: Divide with 255, processing time: 0.027665 seconds
Process-1982, Step 2: Resize if scale is not 1: 0.001907 seconds
Process-1982, Step 3: Get HOG channels: 2.3e-05 seconds
Process-1982, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1982, Step 5: Compute individual channel HOG features for the entire image: 0.241285 seconds
Process-1982, Step 6: Misc initializations: 1e-06 seconds
Process-1982, Step 7: for loop: 0.01485 seconds

Process-1982All steps time: 0.2857379999999994 seconds

Process-1982, Find cars processing time: 0.285891 seconds

Process-1981, Step 1: Divide with 255, processing time: 0.010372 seconds
Process-1981, Step 2: Resize if scale is not 1: 0.000852 seconds
Process-1981, Step 3: Get HOG channels: 7e-06 seconds
Process-1981, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1981, Step 5: Compute individual channel HOG features for the entire image: 0.238786 seconds
Process-1981, Step 6: Misc initializations: 1e-06 seconds
Process-1981, Step 7: for loop: 0.015056 seconds

Process-1981All steps time: 0.265081 seconds

Process-1981, Find cars processing time: 0.265156 seconds

Process-1982, Step 1: Divide with 255, processing time: 0.009502 seconds
Process-1982, Step 2: Resize if scale is not 1: 0.000858 seconds
Process-1982, Step 3: Get HOG channels: 7e-06 seconds
Process-1982, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1982, Step 5: Compute individual channel HOG features for the entire image: 0.154323 seconds
Process-1982, Step 6: Misc initializations: 1e-06 seconds
Process-1982, Step 7: for loop: 0.008466 seconds

Process-1982All steps time: 0.1731629999999998 seconds

Process-1982, Find cars processing time: 0.173226 seconds

Process-1983, Step 1: Divide with 255, processing time: 0.027452 seconds
Process-1983, Step 2: Resize if scale is not 1: 0.00193 seconds
Process-1983, Step 3: Get HOG channels: 2e-05 seconds
Process-1983, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-1983, Step 5: Compute individual channel HOG features for the entire image: 0.246037 seconds
Process-1983, Step 6: Misc initializations: 1e-06 seconds
Process-1983, Step 7: for loop: 0.01547 seconds

Process-1983All steps time: 0.2909139999999995 seconds

Process-1983, Find cars processing time: 0.291051 seconds

Process-1982, Step 1: Divide with 255, processing time: 0.009444 seconds
Process-1982, Step 2: Resize if scale is not 1: 0.000737 seconds
Process-1982, Step 3: Get HOG channels: 7e-06 seconds
Process-1982, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1982, Step 5: Compute individual channel HOG features for the entire image: 0.15186 seconds
Process-1982, Step 6: Misc initializations: 0.0 seconds
Process-1982, Step 7: for loop: 0.00842 seconds

Process-1982All steps time: 0.17047400000000001 seconds

Process-1982, Find cars processing time: 0.170528 seconds

Process-1983, Step 1: Divide with 255, processing time: 0.009405 seconds
Process-1981, Step 1: Divide with 255, processing time: 0.010536 seconds
Process-1983, Step 2: Resize if scale is not 1: 0.000675 seconds
Process-1981, Step 2: Resize if scale is not 1: 0.000819 seconds
Process-1983, Step 3: Get HOG channels: 5e-06 seconds
Process-1981, Step 3: Get HOG channels: 8e-06 seconds
Process-1983, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1981, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1983, Step 5: Compute individual channel HOG features for the entire image: 0.176722 seconds
Process-1981, Step 5: Compute individual channel HOG features for the entire image: 0.163126 seconds
Process-1983, Step 6: Misc initializations: 1e-06 seconds
Process-1981, Step 6: Misc initializations: 0.0 seconds
Process-1981, Step 7: for loop: 0.008134 seconds
Process-1983, Step 7: for loop: 0.009478 seconds

Process-1981All steps time: 0.18263 seconds
Process-1983All steps time: 0.196291 seconds

Process-1981, Find cars processing time: 0.182688 seconds
Process-1983, Find cars processing time: 0.196354 seconds

Process-1981, Step 1: Divide with 255, processing time: 0.010388 seconds
Process-1981, Step 2: Resize if scale is not 1: 0.00155 seconds
Process-1981, Step 3: Get HOG channels: 9e-06 seconds
Process-1981, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1981, Step 5: Compute individual channel HOG features for the entire image: 0.10779 seconds
Process-1981, Step 6: Misc initializations: 0.0 seconds
Process-1981, Step 7: for loop: 0.00598 seconds

Process-1981All steps time: 0.125725 seconds

Process-1981, Find cars processing time: 0.125811 seconds

Process-1982, Step 1: Divide with 255, processing time: 0.006743 seconds
Process-1982, Step 2: Resize if scale is not 1: 0.000863 seconds
Process-1982, Step 3: Get HOG channels: 5e-06 seconds
Process-1982, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1982, Step 5: Compute individual channel HOG features for the entire image: 0.118989 seconds
Process-1982, Step 6: Misc initializations: 1e-06 seconds
Process-1982, Step 7: for loop: 0.006085 seconds

Process-1982All steps time: 0.132692 seconds

Process-1982, Find cars processing time: 0.13277 seconds

Process-1983, Step 1: Divide with 255, processing time: 0.009641 seconds
Process-1983, Step 2: Resize if scale is not 1: 0.001048 seconds
Process-1983, Step 3: Get HOG channels: 7e-06 seconds
Process-1983, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1983, Step 5: Compute individual channel HOG features for the entire image: 0.111267 seconds
Process-1983, Step 6: Misc initializations: 1e-06 seconds
Process-1983, Step 7: for loop: 0.006254 seconds

Process-1983All steps time: 0.128224 seconds

Process-1983, Find cars processing time: 0.128281 seconds

Process-1982, Step 1: Divide with 255, processing time: 0.010061 seconds
Process-1982, Step 2: Resize if scale is not 1: 0.00157 seconds
Process-1982, Step 3: Get HOG channels: 5e-06 seconds
Process-1982, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1982, Step 5: Compute individual channel HOG features for the entire image: 0.095458 seconds
Process-1982, Step 6: Misc initializations: 1e-06 seconds
Process-1982, Step 7: for loop: 0.005472 seconds

Process-1982All steps time: 0.112572 seconds

Process-1982, Find cars processing time: 0.112633 seconds

Process-1981, Step 1: Divide with 255, processing time: 0.007267 seconds
Process-1981, Step 2: Resize if scale is not 1: 0.000496 seconds
Process-1981, Step 3: Get HOG channels: 6e-06 seconds
Process-1981, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1981, Step 5: Compute individual channel HOG features for the entire image: 0.106671 seconds
Process-1981, Step 6: Misc initializations: 1e-06 seconds
Process-1981, Step 7: for loop: 0.006481 seconds

Process-1981All steps time: 0.1209290000000001 seconds

Process-1981, Find cars processing time: 0.121007 seconds

The times for each task are: [0.295351, 0.285891, 0.265156, 0.173226, 0.291051, 0.170528, 0.1826

Minimum: 0.112633 Maximum: 0.295351 Average: 0.1908 seconds

Number of processes used: 3 window size 260

Length of task list: 13

Number of processes used: 3

Process-1984, Step 1: Divide with 255, processing time: 0.027339 seconds

Process-1984, Step 2: Resize if scale is not 1: 0.002029 seconds

Process-1984, Step 3: Get HOG channels: 1.7e-05 seconds

Process-1984, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1984, Step 5: Compute individual channel HOG features for the entire image: 0.244109 seconds

Process-1984, Step 6: Misc initializations: 1e-06 seconds

Process-1984, Step 7: for loop: 0.01756 seconds

Process-1984All steps time: 0.2910609999999996 seconds

Process-1984, Find cars processing time: 0.291227 seconds

Process-1985, Step 1: Divide with 255, processing time: 0.024504 seconds

Process-1985, Step 2: Resize if scale is not 1: 0.001871 seconds

Process-1985, Step 3: Get HOG channels: 1.7e-05 seconds

Process-1985, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1985, Step 5: Compute individual channel HOG features for the entire image: 0.256517 seconds

Process-1985, Step 6: Misc initializations: 1e-06 seconds

Process-1985, Step 7: for loop: 0.020046 seconds

Process-1985All steps time: 0.302961 seconds

Process-1985, Find cars processing time: 0.303127 seconds

Process-1986, Step 1: Divide with 255, processing time: 0.026045 seconds

Process-1986, Step 2: Resize if scale is not 1: 0.001863 seconds

Process-1986, Step 3: Get HOG channels: 1.9e-05 seconds

Process-1986, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1986, Step 5: Compute individual channel HOG features for the entire image: 0.247279 seconds

Process-1986, Step 6: Misc initializations: 1e-06 seconds

Process-1986, Step 7: for loop: 0.016869 seconds

Process-1986All steps time: 0.292082 seconds

Process-1986, Find cars processing time: 0.292246 seconds

Process-1984, Step 1: Divide with 255, processing time: 0.00699 seconds

Process-1984, Step 2: Resize if scale is not 1: 0.000717 seconds

Process-1984, Step 3: Get HOG channels: 7e-06 seconds
Process-1984, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1984, Step 5: Compute individual channel HOG features for the entire image: 0.265181 seconds
Process-1984, Step 6: Misc initializations: 1e-06 seconds
Process-1984, Step 7: for loop: 0.016464 seconds

Process-1984All steps time: 0.28936699999999993 seconds

Process-1984, Find cars processing time: 0.289438 seconds

Process-1985, Step 1: Divide with 255, processing time: 0.007328 seconds
Process-1985, Step 2: Resize if scale is not 1: 0.000688 seconds
Process-1985, Step 3: Get HOG channels: 6e-06 seconds
Process-1985, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1985, Step 5: Compute individual channel HOG features for the entire image: 0.149826 seconds
Process-1985, Step 6: Misc initializations: 0.0 seconds
Process-1985, Step 7: for loop: 0.008262 seconds

Process-1985All steps time: 0.1661159999999999 seconds

Process-1985, Find cars processing time: 0.166171 seconds

Process-1986, Step 1: Divide with 255, processing time: 0.010303 seconds
Process-1986, Step 2: Resize if scale is not 1: 0.000554 seconds
Process-1986, Step 3: Get HOG channels: 5e-06 seconds
Process-1986, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1986, Step 5: Compute individual channel HOG features for the entire image: 0.172031 seconds
Process-1986, Step 6: Misc initializations: 1e-06 seconds
Process-1986, Step 7: for loop: 0.008242 seconds

Process-1986All steps time: 0.1911409999999998 seconds

Process-1986, Find cars processing time: 0.19119 seconds

Process-1986, Step 1: Divide with 255, processing time: 0.008334 seconds
Process-1986, Step 2: Resize if scale is not 1: 0.000958 seconds
Process-1986, Step 3: Get HOG channels: 6e-06 seconds
Process-1986, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1986, Step 5: Compute individual channel HOG features for the entire image: 0.128729 seconds
Process-1986, Step 6: Misc initializations: 1e-06 seconds
Process-1986, Step 7: for loop: 0.007161 seconds

Process-1986All steps time: 0.1451940000000002 seconds

Process-1986, Find cars processing time: 0.145251 seconds

Process-1984, Step 1: Divide with 255, processing time: 0.010792 seconds
Process-1984, Step 2: Resize if scale is not 1: 0.00104 seconds

Process-1984, Step 3: Get HOG channels: 1.1e-05 seconds
Process-1984, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1984, Step 5: Compute individual channel HOG features for the entire image: 0.170401 seconds
Process-1984, Step 6: Misc initializations: 1e-06 seconds
Process-1984, Step 7: for loop: 0.009622 seconds

Process-1984All steps time: 0.191879 seconds

Process-1984, Find cars processing time: 0.191965 seconds

Process-1985, Step 1: Divide with 255, processing time: 0.009288 seconds
Process-1985, Step 2: Resize if scale is not 1: 0.00079 seconds
Process-1985, Step 3: Get HOG channels: 7e-06 seconds
Process-1985, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1985, Step 5: Compute individual channel HOG features for the entire image: 0.140952 seconds
Process-1985, Step 6: Misc initializations: 1e-06 seconds
Process-1985, Step 7: for loop: 0.008408 seconds

Process-1985All steps time: 0.1594539999999998 seconds

Process-1985, Find cars processing time: 0.159525 seconds

Process-1986, Step 1: Divide with 255, processing time: 0.010042 seconds
Process-1986, Step 2: Resize if scale is not 1: 0.000487 seconds
Process-1986, Step 3: Get HOG channels: 5e-06 seconds
Process-1986, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1986, Step 5: Compute individual channel HOG features for the entire image: 0.12685 seconds
Process-1986, Step 6: Misc initializations: 1e-06 seconds
Process-1986, Step 7: for loop: 0.007529 seconds

Process-1986All steps time: 0.144919 seconds

Process-1986, Find cars processing time: 0.144983 seconds

Process-1984, Step 1: Divide with 255, processing time: 0.009753 seconds
Process-1984, Step 2: Resize if scale is not 1: 0.001208 seconds
Process-1984, Step 3: Get HOG channels: 7e-06 seconds
Process-1984, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1984, Step 5: Compute individual channel HOG features for the entire image: 0.142508 seconds
Process-1984, Step 6: Misc initializations: 1e-06 seconds
Process-1984, Step 7: for loop: 0.007551 seconds

Process-1984All steps time: 0.161036 seconds

Process-1984, Find cars processing time: 0.161112 seconds

Process-1985, Step 1: Divide with 255, processing time: 0.010238 seconds
Process-1985, Step 2: Resize if scale is not 1: 0.001251 seconds

```
Process-1985, Step 3: Get HOG channels: 1e-05 seconds
Process-1985, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1985, Step 5: Compute individual channel HOG features for the entire image: 0.146547 seconds
Process-1985, Step 6: Misc initializations: 1e-06 seconds
Process-1985, Step 7: for loop: 0.00733 seconds
```

```
Process-1985All steps time: 0.165386 seconds
```

```
Process-1985, Find cars processing time: 0.165463 seconds
```

```
Process-1986, Step 1: Divide with 255, processing time: 0.006079 seconds
Process-1986, Step 2: Resize if scale is not 1: 0.001113 seconds
Process-1986, Step 3: Get HOG channels: 5e-06 seconds
Process-1986, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1986, Step 5: Compute individual channel HOG features for the entire image: 0.127752 seconds
Process-1986, Step 6: Misc initializations: 1e-06 seconds
Process-1986, Step 7: for loop: 0.006013 seconds
```

```
Process-1986All steps time: 0.1409689999999998 seconds
```

```
Process-1986, Find cars processing time: 0.141034 seconds
```

```
The times for each task are: [0.291227, 0.303127, 0.292246, 0.289438, 0.166171, 0.19119, 0.14525]
```

```
Minimum: 0.141034 Maximum: 0.303127 Average: 0.2033 seconds
```

```
*****
Window sizes used: [1.15, 1.35, 1.55]
3 processes used for testing 3 window sizes
Processing times for each image [2.8356, 3.175, 2.8873, 3.0865, 2.9081, 2.9937] with an average
```

```
Time elapsed so far... 92.3287
```

```
#####
=====
```

```
Number of processes used: 1 window size 260
```

```
Length of task list: 21
```

```
Number of processes used: 1
```

```
Process-1987, Step 1: Divide with 255, processing time: 0.028987 seconds
Process-1987, Step 2: Resize if scale is not 1: 0.01239 seconds
Process-1987, Step 3: Get HOG channels: 7.2e-05 seconds
Process-1987, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.545233 seconds
Process-1987, Step 6: Misc initializations: 0.0 seconds
Process-1987, Step 7: for loop: 0.034443 seconds
```

Process-1987All steps time: 0.6211329999999999 seconds

Process-1987, Find cars processing time: 0.621463 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010334 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.003334 seconds

Process-1987, Step 3: Get HOG channels: 1.1e-05 seconds

Process-1987, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.513815 seconds

Process-1987, Step 6: Misc initializations: 1e-06 seconds

Process-1987, Step 7: for loop: 0.033849 seconds

Process-1987All steps time: 0.561355 seconds

Process-1987, Find cars processing time: 0.56144 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010558 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.003923 seconds

Process-1987, Step 3: Get HOG channels: 1e-05 seconds

Process-1987, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.537652 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.035596 seconds

Process-1987All steps time: 0.587749 seconds

Process-1987, Find cars processing time: 0.587842 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010715 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.005377 seconds

Process-1987, Step 3: Get HOG channels: 9e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.430094 seconds

Process-1987, Step 6: Misc initializations: 1e-06 seconds

Process-1987, Step 7: for loop: 0.032027 seconds

Process-1987All steps time: 0.478232 seconds

Process-1987, Find cars processing time: 0.47831 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010398 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.001094 seconds

Process-1987, Step 3: Get HOG channels: 9e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.350932 seconds

Process-1987, Step 6: Misc initializations: 1e-06 seconds

Process-1987, Step 7: for loop: 0.021995 seconds

Process-1987All steps time: 0.384438 seconds

Process-1987, Find cars processing time: 0.384556 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.013705 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000799 seconds

Process-1987, Step 3: Get HOG channels: 5e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.276295 seconds

Process-1987, Step 6: Misc initializations: 1e-06 seconds

Process-1987, Step 7: for loop: 0.018251 seconds

Process-1987All steps time: 0.309062 seconds

Process-1987, Find cars processing time: 0.309147 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.015519 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.001172 seconds

Process-1987, Step 3: Get HOG channels: 6e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.305441 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.01844 seconds

Process-1987All steps time: 0.340585 seconds

Process-1987, Find cars processing time: 0.340663 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010833 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000744 seconds

Process-1987, Step 3: Get HOG channels: 6e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.276444 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.01798 seconds

Process-1987All steps time: 0.306014 seconds

Process-1987, Find cars processing time: 0.306099 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.01096 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.001044 seconds

Process-1987, Step 3: Get HOG channels: 9e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.252864 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.015753 seconds

Process-1987All steps time: 0.28063899999999997 seconds

Process-1987, Find cars processing time: 0.28073 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010409 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000721 seconds

Process-1987, Step 3: Get HOG channels: 7e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.217788 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.01245 seconds

Process-1987All steps time: 0.241383 seconds

Process-1987, Find cars processing time: 0.241474 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.011096 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000643 seconds

Process-1987, Step 3: Get HOG channels: 6e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.25664 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.01587 seconds

Process-1987All steps time: 0.284263 seconds

Process-1987, Find cars processing time: 0.284349 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.009933 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000669 seconds

Process-1987, Step 3: Get HOG channels: 7e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.229156 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.012416 seconds

Process-1987All steps time: 0.25218799999999997 seconds

Process-1987, Find cars processing time: 0.252269 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010977 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000914 seconds

Process-1987, Step 3: Get HOG channels: 9e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.134259 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.016593 seconds

Process-1987All steps time: 0.162762 seconds

Process-1987, Find cars processing time: 0.162863 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.00947 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000673 seconds

Process-1987, Step 3: Get HOG channels: 7e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.170615 seconds

Process-1987, Step 6: Misc initializations: 1e-06 seconds

Process-1987, Step 7: for loop: 0.009558 seconds

Process-1987All steps time: 0.190332 seconds

Process-1987, Find cars processing time: 0.190396 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010879 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000895 seconds

Process-1987, Step 3: Get HOG channels: 9e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.175833 seconds

Process-1987, Step 6: Misc initializations: 1e-06 seconds

Process-1987, Step 7: for loop: 0.0099 seconds

Process-1987All steps time: 0.1975259999999998 seconds

Process-1987, Find cars processing time: 0.197613 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.011156 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000874 seconds

Process-1987, Step 3: Get HOG channels: 9e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.185186 seconds

Process-1987, Step 6: Misc initializations: 1e-06 seconds

Process-1987, Step 7: for loop: 0.007792 seconds

Process-1987All steps time: 0.205027 seconds

Process-1987, Find cars processing time: 0.205113 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010864 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.00145 seconds

Process-1987, Step 3: Get HOG channels: 9e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.104331 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.005823 seconds

Process-1987All steps time: 0.1224859999999998 seconds

Process-1987, Find cars processing time: 0.122559 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010512 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000492 seconds

Process-1987, Step 3: Get HOG channels: 6e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.102249 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.005836 seconds

Process-1987All steps time: 0.119101 seconds

Process-1987, Find cars processing time: 0.119166 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.009508 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000443 seconds

Process-1987, Step 3: Get HOG channels: 5e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.105748 seconds

Process-1987, Step 6: Misc initializations: 1e-06 seconds

Process-1987, Step 7: for loop: 0.005701 seconds

Process-1987All steps time: 0.1214119999999999 seconds

Process-1987, Find cars processing time: 0.121468 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.006545 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000579 seconds

Process-1987, Step 3: Get HOG channels: 7e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.143547 seconds

Process-1987, Step 6: Misc initializations: 1e-06 seconds

Process-1987, Step 7: for loop: 0.008795 seconds

Process-1987All steps time: 0.159482 seconds

Process-1987, Find cars processing time: 0.159571 seconds

Process-1987, Step 1: Divide with 255, processing time: 0.010241 seconds

Process-1987, Step 2: Resize if scale is not 1: 0.000857 seconds

Process-1987, Step 3: Get HOG channels: 8e-06 seconds

Process-1987, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1987, Step 5: Compute individual channel HOG features for the entire image: 0.114692 seconds

Process-1987, Step 6: Misc initializations: 0.0 seconds

Process-1987, Step 7: for loop: 0.006046 seconds

Process-1987All steps time: 0.131853 seconds

Process-1987, Find cars processing time: 0.13193 seconds

The times for each task are: [0.621463, 0.56144, 0.587842, 0.47831, 0.384556, 0.309147, 0.340663]

Minimum: 0.119166 Maximum: 0.621463 Average: 0.2885 seconds

Number of processes used: 1 window size 260

Length of task list: 21

Number of processes used: 1

Process-1988, Step 1: Divide with 255, processing time: 0.026032 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.012351 seconds

Process-1988, Step 3: Get HOG channels: 0.000103 seconds

Process-1988, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.501341 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.035181 seconds

Process-1988All steps time: 0.575018 seconds

Process-1988, Find cars processing time: 0.575277 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010293 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.005877 seconds

Process-1988, Step 3: Get HOG channels: 1.3e-05 seconds

Process-1988, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.508579 seconds

Process-1988, Step 6: Misc initializations: 0.0 seconds

Process-1988, Step 7: for loop: 0.033926 seconds

Process-1988All steps time: 0.5586990000000001 seconds

Process-1988, Find cars processing time: 0.558802 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.01091 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.005646 seconds

Process-1988, Step 3: Get HOG channels: 1e-05 seconds

Process-1988, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.440538 seconds

Process-1988, Step 6: Misc initializations: 0.0 seconds

Process-1988, Step 7: for loop: 0.037166 seconds

Process-1988All steps time: 0.4942799999999994 seconds

Process-1988, Find cars processing time: 0.494386 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.016054 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.01139 seconds

Process-1988, Step 3: Get HOG channels: 1.3e-05 seconds

Process-1988, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.492849 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.030906 seconds

Process-1988All steps time: 0.551224 seconds

Process-1988, Find cars processing time: 0.551336 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.012408 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.001241 seconds

Process-1988, Step 3: Get HOG channels: 8e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.345379 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.020048 seconds

Process-1988All steps time: 0.3790949999999996 seconds

Process-1988, Find cars processing time: 0.379186 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010442 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000677 seconds

Process-1988, Step 3: Get HOG channels: 5e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.302672 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.019485 seconds

Process-1988All steps time: 0.3332869999999994 seconds

Process-1988, Find cars processing time: 0.333356 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010626 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.00107 seconds

Process-1988, Step 3: Get HOG channels: 9e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.289403 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.018118 seconds

Process-1988All steps time: 0.319237 seconds

Process-1988, Find cars processing time: 0.319324 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010502 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000834 seconds

Process-1988, Step 3: Get HOG channels: 6e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.321313 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.019221 seconds

Process-1988All steps time: 0.351884 seconds

Process-1988, Find cars processing time: 0.351972 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010689 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000654 seconds

Process-1988, Step 3: Get HOG channels: 8e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.239133 seconds

Process-1988, Step 6: Misc initializations: 0.0 seconds

Process-1988, Step 7: for loop: 0.012194 seconds

Process-1988All steps time: 0.262686 seconds

Process-1988, Find cars processing time: 0.262762 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010798 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000925 seconds

Process-1988, Step 3: Get HOG channels: 1e-05 seconds

Process-1988, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.196826 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.012194 seconds

Process-1988All steps time: 0.2207640000000002 seconds

Process-1988, Find cars processing time: 0.220844 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010803 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000914 seconds

Process-1988, Step 3: Get HOG channels: 9e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.186229 seconds

Process-1988, Step 6: Misc initializations: 0.0 seconds

Process-1988, Step 7: for loop: 0.011918 seconds

Process-1988All steps time: 0.209882 seconds

Process-1988, Find cars processing time: 0.209965 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.01059 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000882 seconds

Process-1988, Step 3: Get HOG channels: 9e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.237102 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.012955 seconds

Process-1988All steps time: 0.261548 seconds

Process-1988, Find cars processing time: 0.261636 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.009124 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000516 seconds

Process-1988, Step 3: Get HOG channels: 5e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.188311 seconds

Process-1988, Step 6: Misc initializations: 0.0 seconds

Process-1988, Step 7: for loop: 0.008406 seconds

Process-1988All steps time: 0.206368 seconds

Process-1988, Find cars processing time: 0.206443 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.01101 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000862 seconds

Process-1988, Step 3: Get HOG channels: 9e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.170987 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.009599 seconds

Process-1988All steps time: 0.192477 seconds

Process-1988, Find cars processing time: 0.192565 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.008147 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000539 seconds

Process-1988, Step 3: Get HOG channels: 6e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.186926 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.009661 seconds

Process-1988All steps time: 0.2052860000000002 seconds

Process-1988, Find cars processing time: 0.205351 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.009342 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000571 seconds

Process-1988, Step 3: Get HOG channels: 7e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.198569 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.010288 seconds

Process-1988All steps time: 0.2187849999999998 seconds

Process-1988, Find cars processing time: 0.21887 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010718 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000925 seconds

Process-1988, Step 3: Get HOG channels: 8e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.150339 seconds

Process-1988, Step 6: Misc initializations: 0.0 seconds

Process-1988, Step 7: for loop: 0.007587 seconds

Process-1988All steps time: 0.169584 seconds

Process-1988, Find cars processing time: 0.169659 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010709 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000693 seconds

Process-1988, Step 3: Get HOG channels: 9e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.168577 seconds

Process-1988, Step 6: Misc initializations: 0.0 seconds

Process-1988, Step 7: for loop: 0.007708 seconds

Process-1988All steps time: 0.1877039999999998 seconds

Process-1988, Find cars processing time: 0.187795 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010753 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000809 seconds

Process-1988, Step 3: Get HOG channels: 8e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.130461 seconds

Process-1988, Step 6: Misc initializations: 0.0 seconds

Process-1988, Step 7: for loop: 0.007221 seconds

Process-1988All steps time: 0.149262 seconds

Process-1988, Find cars processing time: 0.149345 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010654 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000682 seconds

Process-1988, Step 3: Get HOG channels: 1e-05 seconds

Process-1988, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.138532 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.006502 seconds

Process-1988All steps time: 0.156391 seconds

Process-1988, Find cars processing time: 0.15648 seconds

Process-1988, Step 1: Divide with 255, processing time: 0.010495 seconds

Process-1988, Step 2: Resize if scale is not 1: 0.000866 seconds

Process-1988, Step 3: Get HOG channels: 8e-06 seconds

Process-1988, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1988, Step 5: Compute individual channel HOG features for the entire image: 0.137137 seconds

Process-1988, Step 6: Misc initializations: 1e-06 seconds

Process-1988, Step 7: for loop: 0.006518 seconds

Process-1988All steps time: 0.155034 seconds

Process-1988, Find cars processing time: 0.155126 seconds

The times for each task are: [0.575277, 0.558802, 0.494386, 0.551336, 0.379186, 0.333356, 0.3193

Minimum: 0.149345 Maximum: 0.575277 Average: 0.2934 seconds

Number of processes used: 1 window size 260

Length of task list: 21

Number of processes used: 1

Process-1989, Step 1: Divide with 255, processing time: 0.027161 seconds

Process-1989, Step 2: Resize if scale is not 1: 0.012268 seconds

Process-1989, Step 3: Get HOG channels: 0.000158 seconds

Process-1989, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.536262 seconds

Process-1989, Step 6: Misc initializations: 1e-06 seconds

Process-1989, Step 7: for loop: 0.03555 seconds

Process-1989All steps time: 0.611414 seconds

Process-1989, Find cars processing time: 0.611668 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.009875 seconds

Process-1989, Step 2: Resize if scale is not 1: 0.008768 seconds
Process-1989, Step 3: Get HOG channels: 1.1e-05 seconds
Process-1989, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.525404 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.035579 seconds

Process-1989All steps time: 0.579647 seconds

Process-1989, Find cars processing time: 0.579747 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.009562 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.004963 seconds
Process-1989, Step 3: Get HOG channels: 9e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.535979 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.040632 seconds

Process-1989All steps time: 0.591155 seconds

Process-1989, Find cars processing time: 0.591281 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.017846 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.004796 seconds
Process-1989, Step 3: Get HOG channels: 1e-05 seconds
Process-1989, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.421204 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.028402 seconds

Process-1989All steps time: 0.472269 seconds

Process-1989, Find cars processing time: 0.472388 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010144 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.004698 seconds
Process-1989, Step 3: Get HOG channels: 6e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.270363 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.01986 seconds

Process-1989All steps time: 0.305079 seconds

Process-1989, Find cars processing time: 0.305158 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.008194 seconds

Process-1989, Step 2: Resize if scale is not 1: 0.000688 seconds
Process-1989, Step 3: Get HOG channels: 5e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.278665 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.018594 seconds

Process-1989All steps time: 0.306154 seconds

Process-1989, Find cars processing time: 0.306212 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010365 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.000631 seconds
Process-1989, Step 3: Get HOG channels: 5e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.277011 seconds
Process-1989, Step 6: Misc initializations: 0.0 seconds
Process-1989, Step 7: for loop: 0.017771 seconds

Process-1989All steps time: 0.305789 seconds

Process-1989, Find cars processing time: 0.305857 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010714 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.001057 seconds
Process-1989, Step 3: Get HOG channels: 9e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.31557 seconds
Process-1989, Step 6: Misc initializations: 0.0 seconds
Process-1989, Step 7: for loop: 0.018149 seconds

Process-1989All steps time: 0.34550900000000007 seconds

Process-1989, Find cars processing time: 0.345594 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010937 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.000633 seconds
Process-1989, Step 3: Get HOG channels: 7e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.268856 seconds
Process-1989, Step 6: Misc initializations: 0.0 seconds
Process-1989, Step 7: for loop: 0.013364 seconds

Process-1989All steps time: 0.2938039999999995 seconds

Process-1989, Find cars processing time: 0.293886 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.029132 seconds

Process-1989, Step 2: Resize if scale is not 1: 0.000691 seconds
Process-1989, Step 3: Get HOG channels: 9e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.249853 seconds
Process-1989, Step 6: Misc initializations: 0.0 seconds
Process-1989, Step 7: for loop: 0.014814 seconds

Process-1989All steps time: 0.29450899999999997 seconds

Process-1989, Find cars processing time: 0.294612 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010222 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.000709 seconds
Process-1989, Step 3: Get HOG channels: 1.6e-05 seconds
Process-1989, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.241116 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.013862 seconds

Process-1989All steps time: 0.2659319999999995 seconds

Process-1989, Find cars processing time: 0.266027 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010514 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.000875 seconds
Process-1989, Step 3: Get HOG channels: 5e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.223907 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.013735 seconds

Process-1989All steps time: 0.249043 seconds

Process-1989, Find cars processing time: 0.249156 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.006332 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.000629 seconds
Process-1989, Step 3: Get HOG channels: 7e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.165474 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.008961 seconds

Process-1989All steps time: 0.18141100000000002 seconds

Process-1989, Find cars processing time: 0.181481 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010291 seconds

Process-1989, Step 2: Resize if scale is not 1: 0.000791 seconds
Process-1989, Step 3: Get HOG channels: 9e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.163753 seconds
Process-1989, Step 6: Misc initializations: 0.0 seconds
Process-1989, Step 7: for loop: 0.008348 seconds

Process-1989All steps time: 0.183201 seconds

Process-1989, Find cars processing time: 0.183287 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.009862 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.000793 seconds
Process-1989, Step 3: Get HOG channels: 1e-05 seconds
Process-1989, Step 4: Define blocks and steps as above: 2.5e-05 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.18007 seconds
Process-1989, Step 6: Misc initializations: 0.0 seconds
Process-1989, Step 7: for loop: 0.008227 seconds

Process-1989All steps time: 0.19898700000000002 seconds

Process-1989, Find cars processing time: 0.199077 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010939 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.000544 seconds
Process-1989, Step 3: Get HOG channels: 5e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.174902 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.009797 seconds

Process-1989All steps time: 0.196194 seconds

Process-1989, Find cars processing time: 0.196268 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010782 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.001407 seconds
Process-1989, Step 3: Get HOG channels: 8e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.130162 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.007304 seconds

Process-1989All steps time: 0.149673 seconds

Process-1989, Find cars processing time: 0.149755 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010163 seconds

Process-1989, Step 2: Resize if scale is not 1: 0.000634 seconds
Process-1989, Step 3: Get HOG channels: 7e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.148263 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.007321 seconds

Process-1989All steps time: 0.166396 seconds

Process-1989, Find cars processing time: 0.166476 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.009416 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.000536 seconds
Process-1989, Step 3: Get HOG channels: 6e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.130538 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.00731 seconds

Process-1989All steps time: 0.147813 seconds

Process-1989, Find cars processing time: 0.14789 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.009359 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.000676 seconds
Process-1989, Step 3: Get HOG channels: 7e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.126806 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.007322 seconds

Process-1989All steps time: 0.144418 seconds

Process-1989, Find cars processing time: 0.1444277 seconds

Process-1989, Step 1: Divide with 255, processing time: 0.010358 seconds
Process-1989, Step 2: Resize if scale is not 1: 0.0009 seconds
Process-1989, Step 3: Get HOG channels: 8e-06 seconds
Process-1989, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1989, Step 5: Compute individual channel HOG features for the entire image: 0.115114 seconds
Process-1989, Step 6: Misc initializations: 1e-06 seconds
Process-1989, Step 7: for loop: 0.006196 seconds

Process-1989All steps time: 0.132586 seconds

Process-1989, Find cars processing time: 0.132673 seconds

The times for each task are: [0.611668, 0.579747, 0.591281, 0.472388, 0.305158, 0.306212, 0.305833]

Minimum: 0.132673 Maximum: 0.611668 Average: 0.2916 seconds

Number of processes used: 1 window size 260

Length of task list: 21

Number of processes used: 1

Process-1990, Step 1: Divide with 255, processing time: 0.025955 seconds

Process-1990, Step 2: Resize if scale is not 1: 0.012661 seconds

Process-1990, Step 3: Get HOG channels: 7.3e-05 seconds

Process-1990, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.483468 seconds

Process-1990, Step 6: Misc initializations: 1e-06 seconds

Process-1990, Step 7: for loop: 0.035415 seconds

Process-1990All steps time: 0.557584 seconds

Process-1990, Find cars processing time: 0.557837 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.011043 seconds

Process-1990, Step 2: Resize if scale is not 1: 0.005854 seconds

Process-1990, Step 3: Get HOG channels: 1.5e-05 seconds

Process-1990, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.43487 seconds

Process-1990, Step 6: Misc initializations: 1e-06 seconds

Process-1990, Step 7: for loop: 0.044232 seconds

Process-1990All steps time: 0.4960259999999997 seconds

Process-1990, Find cars processing time: 0.496144 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.015364 seconds

Process-1990, Step 2: Resize if scale is not 1: 0.006159 seconds

Process-1990, Step 3: Get HOG channels: 9e-06 seconds

Process-1990, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.435234 seconds

Process-1990, Step 6: Misc initializations: 1e-06 seconds

Process-1990, Step 7: for loop: 0.034013 seconds

Process-1990All steps time: 0.490788 seconds

Process-1990, Find cars processing time: 0.49088 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.010889 seconds

Process-1990, Step 2: Resize if scale is not 1: 0.005575 seconds

Process-1990, Step 3: Get HOG channels: 9e-06 seconds

Process-1990, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.418082 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.029285 seconds

Process-1990All steps time: 0.463852 seconds

Process-1990, Find cars processing time: 0.463943 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.010517 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.001025 seconds
Process-1990, Step 3: Get HOG channels: 6e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.280493 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.018487 seconds

Process-1990All steps time: 0.3105359999999999 seconds

Process-1990, Find cars processing time: 0.310617 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.009493 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.00103 seconds
Process-1990, Step 3: Get HOG channels: 1.1e-05 seconds
Process-1990, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.275672 seconds
Process-1990, Step 6: Misc initializations: 0.0 seconds
Process-1990, Step 7: for loop: 0.017491 seconds

Process-1990All steps time: 0.3037089999999995 seconds

Process-1990, Find cars processing time: 0.30381 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.011269 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.000722 seconds
Process-1990, Step 3: Get HOG channels: 6e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.313607 seconds
Process-1990, Step 6: Misc initializations: 0.0 seconds
Process-1990, Step 7: for loop: 0.018158 seconds

Process-1990All steps time: 0.3437690000000005 seconds

Process-1990, Find cars processing time: 0.343844 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.010302 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.001058 seconds
Process-1990, Step 3: Get HOG channels: 9e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.310752 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.020724 seconds

Process-1990All steps time: 0.3428559999999994 seconds

Process-1990, Find cars processing time: 0.342934 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.01007 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.000766 seconds
Process-1990, Step 3: Get HOG channels: 8e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.206264 seconds
Process-1990, Step 6: Misc initializations: 0.0 seconds
Process-1990, Step 7: for loop: 0.011683 seconds

Process-1990All steps time: 0.228801 seconds

Process-1990, Find cars processing time: 0.228879 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.010686 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.00094 seconds
Process-1990, Step 3: Get HOG channels: 1e-05 seconds
Process-1990, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.184241 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.011811 seconds

Process-1990All steps time: 0.2076989999999997 seconds

Process-1990, Find cars processing time: 0.207783 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.00753 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.000645 seconds
Process-1990, Step 3: Get HOG channels: 6e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.178186 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.011753 seconds

Process-1990All steps time: 0.1981280000000003 seconds

Process-1990, Find cars processing time: 0.198185 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.009957 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.00092 seconds
Process-1990, Step 3: Get HOG channels: 1e-05 seconds
Process-1990, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.238297 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.013626 seconds

Process-1990All steps time: 0.262822 seconds

Process-1990, Find cars processing time: 0.262932 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.010951 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.000551 seconds
Process-1990, Step 3: Get HOG channels: 6e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.145024 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.007931 seconds

Process-1990All steps time: 0.1644719999999998 seconds

Process-1990, Find cars processing time: 0.164547 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.009049 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.000762 seconds
Process-1990, Step 3: Get HOG channels: 7e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.12832 seconds
Process-1990, Step 6: Misc initializations: 0.0 seconds
Process-1990, Step 7: for loop: 0.007243 seconds

Process-1990All steps time: 0.14539 seconds

Process-1990, Find cars processing time: 0.145455 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.010581 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.000579 seconds
Process-1990, Step 3: Get HOG channels: 6e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.183691 seconds
Process-1990, Step 6: Misc initializations: 0.0 seconds
Process-1990, Step 7: for loop: 0.009694 seconds

Process-1990All steps time: 0.204559 seconds

Process-1990, Find cars processing time: 0.204639 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.010768 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.000603 seconds
Process-1990, Step 3: Get HOG channels: 6e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.170238 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.008758 seconds

Process-1990All steps time: 0.190382 seconds

Process-1990, Find cars processing time: 0.190465 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.010333 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.001511 seconds
Process-1990, Step 3: Get HOG channels: 9e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.126306 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.007073 seconds

Process-1990All steps time: 0.145242 seconds

Process-1990, Find cars processing time: 0.145317 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.0103 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.000501 seconds
Process-1990, Step 3: Get HOG channels: 6e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.127499 seconds
Process-1990, Step 6: Misc initializations: 0.0 seconds
Process-1990, Step 7: for loop: 0.007277 seconds

Process-1990All steps time: 0.14559 seconds

Process-1990, Find cars processing time: 0.145666 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.010263 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.00063 seconds
Process-1990, Step 3: Get HOG channels: 7e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.131821 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.007206 seconds

Process-1990All steps time: 0.1499349999999998 seconds

Process-1990, Find cars processing time: 0.150014 seconds

Process-1990, Step 1: Divide with 255, processing time: 0.008889 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.00062 seconds
Process-1990, Step 3: Get HOG channels: 6e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 6e-06 seconds

```
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.128506 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.007271 seconds
```

```
Process-1990All steps time: 0.145299 seconds
```

```
Process-1990, Find cars processing time: 0.145379 seconds
```

```
Process-1990, Step 1: Divide with 255, processing time: 0.010523 seconds
Process-1990, Step 2: Resize if scale is not 1: 0.000711 seconds
Process-1990, Step 3: Get HOG channels: 6e-06 seconds
Process-1990, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1990, Step 5: Compute individual channel HOG features for the entire image: 0.145214 seconds
Process-1990, Step 6: Misc initializations: 1e-06 seconds
Process-1990, Step 7: for loop: 0.006689 seconds
```

```
Process-1990All steps time: 0.1631510000000002 seconds
```

```
Process-1990, Find cars processing time: 0.16325 seconds
```

```
The times for each task are: [0.557837, 0.496144, 0.49088, 0.463943, 0.310617, 0.30381, 0.343844]
```

```
Minimum: 0.145317 Maximum: 0.557837 Average: 0.2696 seconds
```

```
*****
Number of processes used: 1 window size 260
Length of task list: 21
Number of processes used: 1
```

```
Process-1991, Step 1: Divide with 255, processing time: 0.030179 seconds
Process-1991, Step 2: Resize if scale is not 1: 0.012047 seconds
Process-1991, Step 3: Get HOG channels: 7.3e-05 seconds
Process-1991, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.482666 seconds
Process-1991, Step 6: Misc initializations: 1e-06 seconds
Process-1991, Step 7: for loop: 0.035736 seconds
```

```
Process-1991All steps time: 0.560713 seconds
```

```
Process-1991, Find cars processing time: 0.561002 seconds
```

```
Process-1991, Step 1: Divide with 255, processing time: 0.010863 seconds
Process-1991, Step 2: Resize if scale is not 1: 0.003574 seconds
Process-1991, Step 3: Get HOG channels: 1.1e-05 seconds
Process-1991, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.434626 seconds
Process-1991, Step 6: Misc initializations: 1e-06 seconds
Process-1991, Step 7: for loop: 0.035162 seconds
```

Process-1991All steps time: 0.484247 seconds

Process-1991, Find cars processing time: 0.484334 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.016438 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.007653 seconds

Process-1991, Step 3: Get HOG channels: 1e-05 seconds

Process-1991, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.514326 seconds

Process-1991, Step 6: Misc initializations: 1e-06 seconds

Process-1991, Step 7: for loop: 0.032841 seconds

Process-1991All steps time: 0.571278 seconds

Process-1991, Find cars processing time: 0.571368 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010352 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.005221 seconds

Process-1991, Step 3: Get HOG channels: 1.6e-05 seconds

Process-1991, Step 4: Define blocks and steps as above: 1.5e-05 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.420611 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.028095 seconds

Process-1991All steps time: 0.46431 seconds

Process-1991, Find cars processing time: 0.464419 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010408 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.001398 seconds

Process-1991, Step 3: Get HOG channels: 1.1e-05 seconds

Process-1991, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.299776 seconds

Process-1991, Step 6: Misc initializations: 1e-06 seconds

Process-1991, Step 7: for loop: 0.018173 seconds

Process-1991All steps time: 0.32977799999999996 seconds

Process-1991, Find cars processing time: 0.32988 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010861 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000988 seconds

Process-1991, Step 3: Get HOG channels: 6e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.26615 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.018307 seconds

Process-1991All steps time: 0.296319 seconds

Process-1991, Find cars processing time: 0.296403 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.01079 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.00117 seconds

Process-1991, Step 3: Get HOG channels: 9e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.267431 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.017858 seconds

Process-1991All steps time: 0.297268 seconds

Process-1991, Find cars processing time: 0.297355 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.009417 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000669 seconds

Process-1991, Step 3: Get HOG channels: 6e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.27707 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.018533 seconds

Process-1991All steps time: 0.305701 seconds

Process-1991, Find cars processing time: 0.30577 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.009927 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000866 seconds

Process-1991, Step 3: Get HOG channels: 9e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.202604 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.012218 seconds

Process-1991All steps time: 0.225632 seconds

Process-1991, Find cars processing time: 0.225709 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010321 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000801 seconds

Process-1991, Step 3: Get HOG channels: 7e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.17756 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.011909 seconds

Process-1991All steps time: 0.200605 seconds

Process-1991, Find cars processing time: 0.200693 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010521 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000672 seconds

Process-1991, Step 3: Get HOG channels: 7e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.199246 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.012454 seconds

Process-1991All steps time: 0.222908 seconds

Process-1991, Find cars processing time: 0.22299 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010892 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000914 seconds

Process-1991, Step 3: Get HOG channels: 8e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.191576 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.012675 seconds

Process-1991All steps time: 0.216074 seconds

Process-1991, Find cars processing time: 0.216164 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.01084 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000892 seconds

Process-1991, Step 3: Get HOG channels: 9e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.132897 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.007357 seconds

Process-1991All steps time: 0.152005 seconds

Process-1991, Find cars processing time: 0.152086 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.0109 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.00084 seconds

Process-1991, Step 3: Get HOG channels: 9e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.184257 seconds

Process-1991, Step 6: Misc initializations: 1e-06 seconds

Process-1991, Step 7: for loop: 0.009889 seconds

Process-1991All steps time: 0.205905 seconds

Process-1991, Find cars processing time: 0.205997 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010294 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000808 seconds

Process-1991, Step 3: Get HOG channels: 8e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.165605 seconds

Process-1991, Step 6: Misc initializations: 1e-06 seconds

Process-1991, Step 7: for loop: 0.009577 seconds

Process-1991All steps time: 0.186302 seconds

Process-1991, Find cars processing time: 0.18637 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010557 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000577 seconds

Process-1991, Step 3: Get HOG channels: 6e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.166479 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.009457 seconds

Process-1991All steps time: 0.1870829999999997 seconds

Process-1991, Find cars processing time: 0.187151 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010445 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.001263 seconds

Process-1991, Step 3: Get HOG channels: 9e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.143157 seconds

Process-1991, Step 6: Misc initializations: 2e-06 seconds

Process-1991, Step 7: for loop: 0.007366 seconds

Process-1991All steps time: 0.1622500000000003 seconds

Process-1991, Find cars processing time: 0.162344 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.009165 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000753 seconds

Process-1991, Step 3: Get HOG channels: 8e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.12543 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.007091 seconds

Process-1991All steps time: 0.142455 seconds

Process-1991, Find cars processing time: 0.142517 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010796 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.00064 seconds

Process-1991, Step 3: Get HOG channels: 1e-05 seconds

Process-1991, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.127891 seconds

Process-1991, Step 6: Misc initializations: 1e-06 seconds

Process-1991, Step 7: for loop: 0.007115 seconds

Process-1991All steps time: 0.146462 seconds

Process-1991, Find cars processing time: 0.146544 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.00917 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.000721 seconds

Process-1991, Step 3: Get HOG channels: 7e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.128949 seconds

Process-1991, Step 6: Misc initializations: 1e-06 seconds

Process-1991, Step 7: for loop: 0.007218 seconds

Process-1991All steps time: 0.146075 seconds

Process-1991, Find cars processing time: 0.146149 seconds

Process-1991, Step 1: Divide with 255, processing time: 0.010898 seconds

Process-1991, Step 2: Resize if scale is not 1: 0.00098 seconds

Process-1991, Step 3: Get HOG channels: 9e-06 seconds

Process-1991, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1991, Step 5: Compute individual channel HOG features for the entire image: 0.115633 seconds

Process-1991, Step 6: Misc initializations: 0.0 seconds

Process-1991, Step 7: for loop: 0.006289 seconds

Process-1991All steps time: 0.133818 seconds

Process-1991, Find cars processing time: 0.133902 seconds

The times for each task are: [0.561002, 0.484334, 0.571368, 0.464419, 0.32988, 0.296403, 0.29735

Minimum: 0.133902 Maximum: 0.571368 Average: 0.2685 seconds

Number of processes used: 1 window size 260

Length of task list: 21

Number of processes used: 1

Process-1992, Step 1: Divide with 255, processing time: 0.038635 seconds
Process-1992, Step 2: Resize if scale is not 1: 0.011925 seconds
Process-1992, Step 3: Get HOG channels: 0.000111 seconds
Process-1992, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.481296 seconds
Process-1992, Step 6: Misc initializations: 0.0 seconds
Process-1992, Step 7: for loop: 0.034989 seconds

Process-1992All steps time: 0.566967 seconds

Process-1992, Find cars processing time: 0.567266 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.008874 seconds
Process-1992, Step 2: Resize if scale is not 1: 0.003712 seconds
Process-1992, Step 3: Get HOG channels: 1e-05 seconds
Process-1992, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.435228 seconds
Process-1992, Step 6: Misc initializations: 0.0 seconds
Process-1992, Step 7: for loop: 0.033783 seconds

Process-1992All steps time: 0.481617 seconds

Process-1992, Find cars processing time: 0.481697 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.015378 seconds
Process-1992, Step 2: Resize if scale is not 1: 0.005653 seconds
Process-1992, Step 3: Get HOG channels: 2.2e-05 seconds
Process-1992, Step 4: Define blocks and steps as above: 1.4e-05 seconds
Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.580085 seconds
Process-1992, Step 6: Misc initializations: 1e-06 seconds
Process-1992, Step 7: for loop: 0.035033 seconds

Process-1992All steps time: 0.6361859999999999 seconds

Process-1992, Find cars processing time: 0.636314 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010678 seconds
Process-1992, Step 2: Resize if scale is not 1: 0.003233 seconds
Process-1992, Step 3: Get HOG channels: 1.1e-05 seconds
Process-1992, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.454152 seconds
Process-1992, Step 6: Misc initializations: 1e-06 seconds
Process-1992, Step 7: for loop: 0.028525 seconds

Process-1992All steps time: 0.496612 seconds

Process-1992, Find cars processing time: 0.496701 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010533 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000846 seconds

Process-1992, Step 3: Get HOG channels: 6e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.270021 seconds

Process-1992, Step 6: Misc initializations: 1e-06 seconds

Process-1992, Step 7: for loop: 0.018292 seconds

Process-1992All steps time: 0.2997069999999995 seconds

Process-1992, Find cars processing time: 0.299784 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.01043 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.001055 seconds

Process-1992, Step 3: Get HOG channels: 9e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.342973 seconds

Process-1992, Step 6: Misc initializations: 0.0 seconds

Process-1992, Step 7: for loop: 0.018287 seconds

Process-1992All steps time: 0.372764 seconds

Process-1992, Find cars processing time: 0.372849 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010285 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.00078 seconds

Process-1992, Step 3: Get HOG channels: 6e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.299465 seconds

Process-1992, Step 6: Misc initializations: 1e-06 seconds

Process-1992, Step 7: for loop: 0.017765 seconds

Process-1992All steps time: 0.3283089999999999 seconds

Process-1992, Find cars processing time: 0.328383 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010823 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.00109 seconds

Process-1992, Step 3: Get HOG channels: 9e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.25978 seconds

Process-1992, Step 6: Misc initializations: 0.0 seconds

Process-1992, Step 7: for loop: 0.017755 seconds

Process-1992All steps time: 0.2894670000000003 seconds

Process-1992, Find cars processing time: 0.289568 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.018921 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000618 seconds

Process-1992, Step 3: Get HOG channels: 6e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.184288 seconds

Process-1992, Step 6: Misc initializations: 0.0 seconds

Process-1992, Step 7: for loop: 0.01191 seconds

Process-1992All steps time: 0.215749 seconds

Process-1992, Find cars processing time: 0.215819 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.011036 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000942 seconds

Process-1992, Step 3: Get HOG channels: 9e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.237072 seconds

Process-1992, Step 6: Misc initializations: 1e-06 seconds

Process-1992, Step 7: for loop: 0.012068 seconds

Process-1992All steps time: 0.2611380000000004 seconds

Process-1992, Find cars processing time: 0.261217 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.006114 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000479 seconds

Process-1992, Step 3: Get HOG channels: 5e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.187755 seconds

Process-1992, Step 6: Misc initializations: 0.0 seconds

Process-1992, Step 7: for loop: 0.012038 seconds

Process-1992All steps time: 0.206397 seconds

Process-1992, Find cars processing time: 0.206466 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.011093 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000921 seconds

Process-1992, Step 3: Get HOG channels: 9e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.201863 seconds

Process-1992, Step 6: Misc initializations: 1e-06 seconds

Process-1992, Step 7: for loop: 0.012243 seconds

Process-1992All steps time: 0.2261409999999998 seconds

Process-1992, Find cars processing time: 0.226226 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010216 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000844 seconds

Process-1992, Step 3: Get HOG channels: 8e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.136616 seconds

Process-1992, Step 6: Misc initializations: 0.0 seconds

Process-1992, Step 7: for loop: 0.007661 seconds

Process-1992All steps time: 0.155353 seconds

Process-1992, Find cars processing time: 0.155428 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010936 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000842 seconds

Process-1992, Step 3: Get HOG channels: 9e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.131016 seconds

Process-1992, Step 6: Misc initializations: 0.0 seconds

Process-1992, Step 7: for loop: 0.007237 seconds

Process-1992All steps time: 0.15005 seconds

Process-1992, Find cars processing time: 0.150131 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010694 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000693 seconds

Process-1992, Step 3: Get HOG channels: 5e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.134225 seconds

Process-1992, Step 6: Misc initializations: 1e-06 seconds

Process-1992, Step 7: for loop: 0.008665 seconds

Process-1992All steps time: 0.15429 seconds

Process-1992, Find cars processing time: 0.154368 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010285 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000831 seconds

Process-1992, Step 3: Get HOG channels: 8e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.184518 seconds

Process-1992, Step 6: Misc initializations: 1e-06 seconds

Process-1992, Step 7: for loop: 0.009731 seconds

Process-1992All steps time: 0.2053819999999998 seconds

Process-1992, Find cars processing time: 0.205464 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010957 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.001435 seconds

Process-1992, Step 3: Get HOG channels: 9e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.130503 seconds

Process-1992, Step 6: Misc initializations: 1e-06 seconds

Process-1992, Step 7: for loop: 0.007323 seconds

Process-1992All steps time: 0.150237 seconds

Process-1992, Find cars processing time: 0.150329 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010233 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.00073 seconds

Process-1992, Step 3: Get HOG channels: 8e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.126774 seconds

Process-1992, Step 6: Misc initializations: 0.0 seconds

Process-1992, Step 7: for loop: 0.007076 seconds

Process-1992All steps time: 0.14483 seconds

Process-1992, Find cars processing time: 0.144902 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.014273 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000788 seconds

Process-1992, Step 3: Get HOG channels: 8e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.140334 seconds

Process-1992, Step 6: Misc initializations: 1e-06 seconds

Process-1992, Step 7: for loop: 0.007497 seconds

Process-1992All steps time: 0.162912 seconds

Process-1992, Find cars processing time: 0.163011 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.011545 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000796 seconds

Process-1992, Step 3: Get HOG channels: 9e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.127293 seconds

Process-1992, Step 6: Misc initializations: 0.0 seconds

Process-1992, Step 7: for loop: 0.007171 seconds

Process-1992All steps time: 0.146824 seconds

Process-1992, Find cars processing time: 0.146909 seconds

Process-1992, Step 1: Divide with 255, processing time: 0.010817 seconds

Process-1992, Step 2: Resize if scale is not 1: 0.000963 seconds

Process-1992, Step 3: Get HOG channels: 8e-06 seconds

Process-1992, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1992, Step 5: Compute individual channel HOG features for the entire image: 0.112234 seconds

Process-1992, Step 6: Misc initializations: 0.0 seconds

Process-1992, Step 7: for loop: 0.006051 seconds

Process-1992All steps time: 0.130083 seconds

Process-1992, Find cars processing time: 0.13016 seconds

The times for each task are: [0.567266, 0.481697, 0.636314, 0.496701, 0.299784, 0.372849, 0.3283

Minimum: 0.13016 Maximum: 0.636314 Average: 0.2754 seconds

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

1 processes used for testing 5 window sizes

Processing times for each image [12.2095, 11.778, 12.4426, 11.7697, 11.7445, 11.757] with an ave

Time elapsed so far... 164.03

#####

Number of processes used: 2 window size 260

Length of task list: 21

Number of processes used: 2

Process-1993, Step 1: Divide with 255, processing time: 0.031588 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.012598 seconds

Process-1993, Step 3: Get HOG channels: 0.000114 seconds

Process-1993, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.524338 seconds

Process-1993, Step 6: Misc initializations: 1e-06 seconds

Process-1993, Step 7: for loop: 0.037522 seconds

Process-1993All steps time: 0.6061730000000001 seconds

Process-1993, Find cars processing time: 0.606432 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.029253 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.01212 seconds

Process-1994, Step 3: Get HOG channels: 0.000132 seconds

Process-1994, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.453269 seconds

Process-1994, Step 6: Misc initializations: 0.0 seconds

Process-1994, Step 7: for loop: 0.03885 seconds

Process-1994All steps time: 0.533631 seconds

Process-1994, Find cars processing time: 0.533834 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.00979 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.005445 seconds

Process-1993, Step 3: Get HOG channels: 7e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.43987 seconds

Process-1993, Step 6: Misc initializations: 1e-06 seconds

Process-1993, Step 7: for loop: 0.036082 seconds

Process-1993All steps time: 0.4912029999999995 seconds

Process-1993, Find cars processing time: 0.491276 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.009626 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.005371 seconds

Process-1994, Step 3: Get HOG channels: 8e-06 seconds

Process-1994, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.390574 seconds

Process-1994, Step 6: Misc initializations: 0.0 seconds

Process-1994, Step 7: for loop: 0.028926 seconds

Process-1994All steps time: 0.4345139999999996 seconds

Process-1994, Find cars processing time: 0.434587 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.010456 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.001153 seconds

Process-1993, Step 3: Get HOG channels: 8e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.270833 seconds

Process-1993, Step 6: Misc initializations: 0.0 seconds

Process-1993, Step 7: for loop: 0.017878 seconds

Process-1993All steps time: 0.300336 seconds

Process-1993, Find cars processing time: 0.3004 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.01033 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.005786 seconds

Process-1994, Step 3: Get HOG channels: 9e-06 seconds

Process-1994, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.269007 seconds

Process-1994, Step 6: Misc initializations: 1e-06 seconds

Process-1994, Step 7: for loop: 0.017715 seconds

Process-1994All steps time: 0.30285599999999996 seconds

Process-1994, Find cars processing time: 0.302946 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.009699 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.0006 seconds

Process-1993, Step 3: Get HOG channels: 5e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.263886 seconds

Process-1993, Step 6: Misc initializations: 0.0 seconds

Process-1993, Step 7: for loop: 0.018452 seconds

Process-1993All steps time: 0.292648 seconds

Process-1993, Find cars processing time: 0.292711 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.010412 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.000788 seconds

Process-1994, Step 3: Get HOG channels: 6e-06 seconds

Process-1994, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.32287 seconds

Process-1994, Step 6: Misc initializations: 1e-06 seconds

Process-1994, Step 7: for loop: 0.0183 seconds

Process-1994All steps time: 0.352384 seconds

Process-1994, Find cars processing time: 0.352453 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.010186 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.000842 seconds

Process-1993, Step 3: Get HOG channels: 9e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.280997 seconds

Process-1993, Step 6: Misc initializations: 1e-06 seconds

Process-1993, Step 7: for loop: 0.012838 seconds

Process-1993All steps time: 0.304881 seconds

Process-1993, Find cars processing time: 0.304953 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.010313 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.000869 seconds

Process-1994, Step 3: Get HOG channels: 8e-06 seconds

Process-1994, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.193059 seconds

Process-1994, Step 6: Misc initializations: 1e-06 seconds

Process-1994, Step 7: for loop: 0.012859 seconds

Process-1994All steps time: 0.217117 seconds

Process-1994, Find cars processing time: 0.217199 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.010392 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.000968 seconds

Process-1993, Step 3: Get HOG channels: 9e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.22758 seconds

Process-1993, Step 6: Misc initializations: 1e-06 seconds

Process-1993, Step 7: for loop: 0.012955 seconds

Process-1993All steps time: 0.251912 seconds

Process-1993, Find cars processing time: 0.251987 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.006084 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.000473 seconds

Process-1993, Step 3: Get HOG channels: 5e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.157879 seconds

Process-1993, Step 6: Misc initializations: 1e-06 seconds

Process-1993, Step 7: for loop: 0.008739 seconds

Process-1993All steps time: 0.1731869999999998 seconds

Process-1993, Find cars processing time: 0.173252 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.009478 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.000587 seconds

Process-1993, Step 3: Get HOG channels: 6e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.174983 seconds

Process-1993, Step 6: Misc initializations: 2e-06 seconds

Process-1993, Step 7: for loop: 0.009595 seconds

Process-1993All steps time: 0.194657 seconds

Process-1993, Find cars processing time: 0.19474 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.010424 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.000894 seconds

Process-1994, Step 3: Get HOG channels: 7e-06 seconds

Process-1994, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.202013 seconds

Process-1994, Step 6: Misc initializations: 1e-06 seconds

Process-1994, Step 7: for loop: 0.014151 seconds

Process-1994All steps time: 0.227498 seconds

Process-1994, Find cars processing time: 0.227573 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.01101 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.000914 seconds

Process-1993, Step 3: Get HOG channels: 8e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.17126 seconds

Process-1993, Step 6: Misc initializations: 1e-06 seconds

Process-1993, Step 7: for loop: 0.009674 seconds

Process-1993All steps time: 0.192877 seconds

Process-1993, Find cars processing time: 0.192972 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.01077 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.000888 seconds

Process-1994, Step 3: Get HOG channels: 8e-06 seconds

Process-1994, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.156219 seconds

Process-1994, Step 6: Misc initializations: 2e-06 seconds

Process-1994, Step 7: for loop: 0.008552 seconds

Process-1994All steps time: 0.176447 seconds

Process-1994, Find cars processing time: 0.176549 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.009554 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.001173 seconds

Process-1994, Step 3: Get HOG channels: 7e-06 seconds

Process-1994, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.106033 seconds

Process-1994, Step 6: Misc initializations: 1e-06 seconds

Process-1994, Step 7: for loop: 0.006239 seconds

Process-1994All steps time: 0.123014 seconds

Process-1994, Find cars processing time: 0.123098 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.010504 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.001334 seconds

Process-1993, Step 3: Get HOG channels: 8e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.128773 seconds

Process-1993, Step 6: Misc initializations: 1e-06 seconds

Process-1993, Step 7: for loop: 0.014197 seconds

Process-1993All steps time: 0.154824 seconds

Process-1993, Find cars processing time: 0.154908 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.009694 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.000682 seconds

Process-1994, Step 3: Get HOG channels: 5e-06 seconds

Process-1994, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.1039 seconds

Process-1994, Step 6: Misc initializations: 1e-06 seconds

Process-1994, Step 7: for loop: 0.006016 seconds

Process-1994All steps time: 0.120304 seconds

Process-1994, Find cars processing time: 0.120368 seconds

Process-1993, Step 1: Divide with 255, processing time: 0.010448 seconds

Process-1993, Step 2: Resize if scale is not 1: 0.000819 seconds

Process-1993, Step 3: Get HOG channels: 8e-06 seconds

Process-1993, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1993, Step 5: Compute individual channel HOG features for the entire image: 0.129328 seconds

Process-1993, Step 6: Misc initializations: 1e-06 seconds

Process-1993, Step 7: for loop: 0.007234 seconds

Process-1993All steps time: 0.1478479999999998 seconds

Process-1993, Find cars processing time: 0.147924 seconds

Process-1994, Step 1: Divide with 255, processing time: 0.010023 seconds

Process-1994, Step 2: Resize if scale is not 1: 0.000559 seconds

Process-1994, Step 3: Get HOG channels: 5e-06 seconds

Process-1994, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1994, Step 5: Compute individual channel HOG features for the entire image: 0.103609 seconds

Process-1994, Step 6: Misc initializations: 1e-06 seconds

Process-1994, Step 7: for loop: 0.006133 seconds

Process-1994All steps time: 0.1203350000000001 seconds

Process-1994, Find cars processing time: 0.120419 seconds

The times for each task are: [0.606432, 0.533834, 0.491276, 0.434587, 0.3004, 0.302946, 0.292711]

Minimum: 0.120368 Maximum: 0.606432 Average: 0.2724 seconds

Number of processes used: 2 window size 260

Length of task list: 21
Number of processes used: 2

Process-1995, Step 1: Divide with 255, processing time: 0.034177 seconds
Process-1995, Step 2: Resize if scale is not 1: 0.01116 seconds
Process-1995, Step 3: Get HOG channels: 0.000113 seconds
Process-1995, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.559259 seconds
Process-1995, Step 6: Misc initializations: 1e-06 seconds
Process-1995, Step 7: for loop: 0.040269 seconds

Process-1995All steps time: 0.644988 seconds

Process-1995, Find cars processing time: 0.645219 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.034632 seconds
Process-1996, Step 2: Resize if scale is not 1: 0.013266 seconds
Process-1996, Step 3: Get HOG channels: 0.000138 seconds
Process-1996, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.537147 seconds
Process-1996, Step 6: Misc initializations: 1e-06 seconds
Process-1996, Step 7: for loop: 0.037847 seconds

Process-1996All steps time: 0.623042 seconds

Process-1996, Find cars processing time: 0.623305 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.009602 seconds
Process-1995, Step 2: Resize if scale is not 1: 0.006355 seconds
Process-1995, Step 3: Get HOG channels: 1e-05 seconds
Process-1995, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.534075 seconds
Process-1995, Step 6: Misc initializations: 1e-06 seconds
Process-1995, Step 7: for loop: 0.035055 seconds

Process-1995All steps time: 0.5851059999999999 seconds

Process-1995, Find cars processing time: 0.585174 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.009896 seconds
Process-1996, Step 2: Resize if scale is not 1: 0.007129 seconds
Process-1996, Step 3: Get HOG channels: 9e-06 seconds
Process-1996, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.485005 seconds
Process-1996, Step 6: Misc initializations: 1e-06 seconds
Process-1996, Step 7: for loop: 0.030241 seconds

Process-1996All steps time: 0.532289 seconds

Process-1996, Find cars processing time: 0.532378 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.010671 seconds

Process-1995, Step 2: Resize if scale is not 1: 0.00111 seconds

Process-1995, Step 3: Get HOG channels: 8e-06 seconds

Process-1995, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.339378 seconds

Process-1995, Step 6: Misc initializations: 1e-06 seconds

Process-1995, Step 7: for loop: 0.018819 seconds

Process-1995All steps time: 0.36999699999999996 seconds

Process-1995, Find cars processing time: 0.370069 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.007383 seconds

Process-1996, Step 2: Resize if scale is not 1: 0.00139 seconds

Process-1996, Step 3: Get HOG channels: 1e-05 seconds

Process-1996, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.375879 seconds

Process-1996, Step 6: Misc initializations: 2e-06 seconds

Process-1996, Step 7: for loop: 0.036248 seconds

Process-1996All steps time: 0.420922 seconds

Process-1996, Find cars processing time: 0.421026 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.010701 seconds

Process-1995, Step 2: Resize if scale is not 1: 0.001064 seconds

Process-1995, Step 3: Get HOG channels: 8e-06 seconds

Process-1995, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.349902 seconds

Process-1995, Step 6: Misc initializations: 1e-06 seconds

Process-1995, Step 7: for loop: 0.019279 seconds

Process-1995All steps time: 0.3809639999999997 seconds

Process-1995, Find cars processing time: 0.381051 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.009672 seconds

Process-1996, Step 2: Resize if scale is not 1: 0.001019 seconds

Process-1996, Step 3: Get HOG channels: 8e-06 seconds

Process-1996, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.271371 seconds

Process-1996, Step 6: Misc initializations: 1e-06 seconds

Process-1996, Step 7: for loop: 0.019172 seconds

Process-1996All steps time: 0.3012519999999996 seconds

Process-1996, Find cars processing time: 0.301324 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.010516 seconds

Process-1995, Step 2: Resize if scale is not 1: 0.000868 seconds

Process-1995, Step 3: Get HOG channels: 8e-06 seconds

Process-1995, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.197423 seconds

Process-1995, Step 6: Misc initializations: 1e-06 seconds

Process-1995, Step 7: for loop: 0.012503 seconds

Process-1995All steps time: 0.22132599999999997 seconds

Process-1995, Find cars processing time: 0.221395 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.010296 seconds

Process-1996, Step 2: Resize if scale is not 1: 0.000877 seconds

Process-1996, Step 3: Get HOG channels: 8e-06 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.009862 seconds

Process-1996, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1995, Step 2: Resize if scale is not 1: 0.000827 seconds

Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.205605 seconds

Process-1995, Step 3: Get HOG channels: 8e-06 seconds

Process-1996, Step 6: Misc initializations: 0.0 seconds

Process-1995, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.199696 seconds

Process-1996, Step 7: for loop: 0.013018 seconds

Process-1995, Step 6: Misc initializations: 0.0 seconds

Process-1995, Step 7: for loop: 0.013083 seconds

Process-1996All steps time: 0.22981200000000002 seconds

Process-1995All steps time: 0.22348400000000002 seconds

Process-1996, Find cars processing time: 0.229882 seconds

Process-1995, Find cars processing time: 0.223548 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.010286 seconds

Process-1995, Step 2: Resize if scale is not 1: 0.000854 seconds

Process-1995, Step 3: Get HOG channels: 9e-06 seconds

Process-1995, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.137114 seconds

Process-1995, Step 6: Misc initializations: 0.0 seconds

Process-1995, Step 7: for loop: 0.007423 seconds

Process-1995All steps time: 0.15569400000000003 seconds

Process-1995, Find cars processing time: 0.155772 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.01027 seconds

Process-1995, Step 2: Resize if scale is not 1: 0.000783 seconds

Process-1995, Step 3: Get HOG channels: 8e-06 seconds

Process-1995, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.147841 seconds

Process-1995, Step 6: Misc initializations: 0.0 seconds

Process-1995, Step 7: for loop: 0.007918 seconds

Process-1995All steps time: 0.166828 seconds

Process-1995, Find cars processing time: 0.166896 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.007269 seconds

Process-1996, Step 2: Resize if scale is not 1: 0.000623 seconds

Process-1996, Step 3: Get HOG channels: 7e-06 seconds

Process-1996, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.282415 seconds

Process-1996, Step 6: Misc initializations: 1e-06 seconds

Process-1996, Step 7: for loop: 0.017072 seconds

Process-1996All steps time: 0.307394 seconds

Process-1996, Find cars processing time: 0.30748 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.011065 seconds

Process-1995, Step 2: Resize if scale is not 1: 0.000821 seconds

Process-1995, Step 3: Get HOG channels: 5e-06 seconds

Process-1995, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.130367 seconds

Process-1995, Step 6: Misc initializations: 0.0 seconds

Process-1995, Step 7: for loop: 0.007395 seconds

Process-1995All steps time: 0.149658 seconds

Process-1995, Find cars processing time: 0.149718 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.011451 seconds

Process-1996, Step 2: Resize if scale is not 1: 0.000872 seconds

Process-1996, Step 3: Get HOG channels: 9e-06 seconds

Process-1996, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.185051 seconds

Process-1996, Step 6: Misc initializations: 1e-06 seconds

Process-1996, Step 7: for loop: 0.00959 seconds

Process-1996All steps time: 0.2069839999999997 seconds

Process-1996, Find cars processing time: 0.20707 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.01036 seconds

Process-1996, Step 2: Resize if scale is not 1: 0.001367 seconds

Process-1996, Step 3: Get HOG channels: 7e-06 seconds

Process-1996, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.128046 seconds

Process-1996, Step 6: Misc initializations: 1e-06 seconds

Process-1996, Step 7: for loop: 0.007039 seconds

Process-1996All steps time: 0.1468269999999999 seconds

Process-1996, Find cars processing time: 0.146896 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.0102 seconds

Process-1995, Step 2: Resize if scale is not 1: 0.001316 seconds

Process-1995, Step 3: Get HOG channels: 8e-06 seconds

Process-1995, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.104506 seconds

Process-1995, Step 6: Misc initializations: 1e-06 seconds

Process-1995, Step 7: for loop: 0.006284 seconds

Process-1995All steps time: 0.122322 seconds

Process-1995, Find cars processing time: 0.122388 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.010452 seconds

Process-1996, Step 2: Resize if scale is not 1: 0.000755 seconds

Process-1996, Step 3: Get HOG channels: 8e-06 seconds

Process-1996, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.126307 seconds

Process-1996, Step 6: Misc initializations: 1e-06 seconds

Process-1996, Step 7: for loop: 0.007243 seconds

Process-1996All steps time: 0.144774 seconds

Process-1996, Find cars processing time: 0.144843 seconds

Process-1995, Step 1: Divide with 255, processing time: 0.009494 seconds

Process-1995, Step 2: Resize if scale is not 1: 0.000757 seconds

Process-1995, Step 3: Get HOG channels: 7e-06 seconds

Process-1995, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1995, Step 5: Compute individual channel HOG features for the entire image: 0.099251 seconds

Process-1995, Step 6: Misc initializations: 0.0 seconds

Process-1995, Step 7: for loop: 0.005697 seconds

Process-1995All steps time: 0.115213 seconds

Process-1995, Find cars processing time: 0.115273 seconds

Process-1996, Step 1: Divide with 255, processing time: 0.010619 seconds

Process-1996, Step 2: Resize if scale is not 1: 0.00067 seconds

Process-1996, Step 3: Get HOG channels: 6e-06 seconds

Process-1996, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1996, Step 5: Compute individual channel HOG features for the entire image: 0.112195 seconds

Process-1996, Step 6: Misc initializations: 1e-06 seconds

Process-1996, Step 7: for loop: 0.025116 seconds

Process-1996All steps time: 0.14861400000000002 seconds

Process-1996, Find cars processing time: 0.148701 seconds

The times for each task are: [0.645219, 0.623305, 0.585174, 0.532378, 0.370069, 0.421026, 0.3810

Minimum: 0.115273 Maximum: 0.645219 Average: 0.2952 seconds

Number of processes used: 2 window size 260

Length of task list: 21

Number of processes used: 2

Process-1997, Step 1: Divide with 255, processing time: 0.033507 seconds

Process-1997, Step 2: Resize if scale is not 1: 0.01317 seconds

Process-1997, Step 3: Get HOG channels: 8.6e-05 seconds

Process-1997, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.532355 seconds

Process-1997, Step 6: Misc initializations: 1e-06 seconds

Process-1997, Step 7: for loop: 0.035696 seconds

Process-1997All steps time: 0.614827 seconds

Process-1997, Find cars processing time: 0.615101 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.007886 seconds

Process-1997, Step 2: Resize if scale is not 1: 0.007548 seconds

Process-1997, Step 3: Get HOG channels: 1e-05 seconds

Process-1997, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.453167 seconds

Process-1997, Step 6: Misc initializations: 1e-06 seconds

Process-1997, Step 7: for loop: 0.037155 seconds

Process-1997All steps time: 0.505778 seconds

Process-1997, Find cars processing time: 0.505873 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.030653 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.010318 seconds
Process-1998, Step 3: Get HOG channels: 0.00012 seconds
Process-1998, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.555466 seconds
Process-1998, Step 6: Misc initializations: 1e-06 seconds
Process-1998, Step 7: for loop: 0.036712 seconds

Process-1998All steps time: 0.633279 seconds

Process-1998, Find cars processing time: 0.633557 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.008937 seconds
Process-1997, Step 2: Resize if scale is not 1: 0.005353 seconds
Process-1997, Step 3: Get HOG channels: 8e-06 seconds
Process-1997, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.422646 seconds
Process-1997, Step 6: Misc initializations: 0.0 seconds
Process-1997, Step 7: for loop: 0.03041 seconds

Process-1997All steps time: 0.467362 seconds

Process-1997, Find cars processing time: 0.467439 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.010521 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.000771 seconds
Process-1998, Step 3: Get HOG channels: 5e-06 seconds
Process-1998, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.287009 seconds
Process-1998, Step 6: Misc initializations: 1e-06 seconds
Process-1998, Step 7: for loop: 0.019416 seconds

Process-1998All steps time: 0.3177279999999995 seconds

Process-1998, Find cars processing time: 0.317797 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.01065 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.001091 seconds
Process-1998, Step 3: Get HOG channels: 9e-06 seconds
Process-1998, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.282177 seconds
Process-1998, Step 6: Misc initializations: 1e-06 seconds
Process-1998, Step 7: for loop: 0.019665 seconds

Process-1998All steps time: 0.313602 seconds

Process-1998, Find cars processing time: 0.313679 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.009679 seconds
Process-1997, Step 2: Resize if scale is not 1: 0.001569 seconds
Process-1997, Step 3: Get HOG channels: 8e-06 seconds
Process-1997, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.275805 seconds
Process-1997, Step 6: Misc initializations: 0.0 seconds
Process-1997, Step 7: for loop: 0.018674 seconds

Process-1997All steps time: 0.30574200000000007 seconds

Process-1997, Find cars processing time: 0.305809 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.018346 seconds
Process-1997, Step 2: Resize if scale is not 1: 0.000504 seconds
Process-1997, Step 3: Get HOG channels: 6e-06 seconds
Process-1997, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.195789 seconds
Process-1997, Step 6: Misc initializations: 0.0 seconds
Process-1997, Step 7: for loop: 0.012582 seconds

Process-1997All steps time: 0.227233 seconds

Process-1997, Find cars processing time: 0.227301 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.010844 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.001074 seconds
Process-1998, Step 3: Get HOG channels: 6e-06 seconds
Process-1998, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.264777 seconds
Process-1998, Step 6: Misc initializations: 0.0 seconds
Process-1998, Step 7: for loop: 0.019034 seconds

Process-1998All steps time: 0.295741 seconds

Process-1998, Find cars processing time: 0.295814 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.017428 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.000554 seconds
Process-1998, Step 3: Get HOG channels: 5e-06 seconds
Process-1998, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.187712 seconds
Process-1998, Step 6: Misc initializations: 1e-06 seconds
Process-1998, Step 7: for loop: 0.01279 seconds

Process-1998All steps time: 0.218495 seconds

Process-1998, Find cars processing time: 0.218568 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.010563 seconds
Process-1997, Step 2: Resize if scale is not 1: 0.00083 seconds
Process-1997, Step 3: Get HOG channels: 7e-06 seconds
Process-1997, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.191961 seconds
Process-1997, Step 6: Misc initializations: 0.0 seconds
Process-1997, Step 7: for loop: 0.013263 seconds

Process-1997All steps time: 0.216631 seconds

Process-1997, Find cars processing time: 0.216696 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.011899 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.000935 seconds
Process-1998, Step 3: Get HOG channels: 1e-05 seconds
Process-1998, Step 4: Define blocks and steps as above: 1.9e-05 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.188813 seconds
Process-1998, Step 6: Misc initializations: 1e-06 seconds
Process-1998, Step 7: for loop: 0.012563 seconds

Process-1998All steps time: 0.21424 seconds

Process-1998, Find cars processing time: 0.214325 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.010504 seconds
Process-1997, Step 2: Resize if scale is not 1: 0.000898 seconds
Process-1997, Step 3: Get HOG channels: 8e-06 seconds
Process-1997, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.14727 seconds
Process-1997, Step 6: Misc initializations: 1e-06 seconds
Process-1997, Step 7: for loop: 0.007806 seconds

Process-1997All steps time: 0.166495 seconds

Process-1997, Find cars processing time: 0.166566 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.009482 seconds
Process-1997, Step 2: Resize if scale is not 1: 0.000757 seconds
Process-1997, Step 3: Get HOG channels: 8e-06 seconds
Process-1997, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.134801 seconds
Process-1997, Step 6: Misc initializations: 1e-06 seconds
Process-1997, Step 7: for loop: 0.007767 seconds

Process-1997All steps time: 0.152825 seconds

Process-1997, Find cars processing time: 0.15289 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.010479 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.000833 seconds
Process-1998, Step 3: Get HOG channels: 8e-06 seconds
Process-1998, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.132014 seconds
Process-1998, Step 6: Misc initializations: 1e-06 seconds
Process-1998, Step 7: for loop: 0.007418 seconds

Process-1998All steps time: 0.150761 seconds

Process-1998, Find cars processing time: 0.15084 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.009727 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.001284 seconds
Process-1998, Step 3: Get HOG channels: 8e-06 seconds
Process-1998, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.096746 seconds
Process-1998, Step 6: Misc initializations: 1e-06 seconds
Process-1998, Step 7: for loop: 0.005957 seconds

Process-1998All steps time: 0.113731 seconds

Process-1998, Find cars processing time: 0.113792 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.009371 seconds
Process-1997, Step 2: Resize if scale is not 1: 0.000527 seconds
Process-1997, Step 3: Get HOG channels: 6e-06 seconds
Process-1997, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.149752 seconds
Process-1997, Step 6: Misc initializations: 1e-06 seconds
Process-1997, Step 7: for loop: 0.008233 seconds

Process-1997All steps time: 0.167897 seconds

Process-1997, Find cars processing time: 0.167987 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.010314 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.000431 seconds
Process-1998, Step 3: Get HOG channels: 5e-06 seconds
Process-1998, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.100229 seconds
Process-1998, Step 6: Misc initializations: 1e-06 seconds
Process-1998, Step 7: for loop: 0.005475 seconds

Process-1998All steps time: 0.116461 seconds

Process-1998, Find cars processing time: 0.116516 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.010368 seconds
Process-1997, Step 2: Resize if scale is not 1: 0.001396 seconds
Process-1997, Step 3: Get HOG channels: 9e-06 seconds
Process-1997, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.11665 seconds
Process-1997, Step 6: Misc initializations: 1e-06 seconds
Process-1997, Step 7: for loop: 0.005574 seconds

Process-1997All steps time: 0.134008 seconds

Process-1997, Find cars processing time: 0.134089 seconds

Process-1997, Step 1: Divide with 255, processing time: 0.010623 seconds
Process-1997, Step 2: Resize if scale is not 1: 0.000733 seconds
Process-1997, Step 3: Get HOG channels: 2.1e-05 seconds
Process-1997, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-1997, Step 5: Compute individual channel HOG features for the entire image: 0.088664 seconds
Process-1997, Step 6: Misc initializations: 0.0 seconds
Process-1997, Step 7: for loop: 0.004765 seconds

Process-1997All steps time: 0.10481300000000002 seconds

Process-1997, Find cars processing time: 0.104892 seconds

Process-1998, Step 1: Divide with 255, processing time: 0.010826 seconds
Process-1998, Step 2: Resize if scale is not 1: 0.000496 seconds
Process-1998, Step 3: Get HOG channels: 6e-06 seconds
Process-1998, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1998, Step 5: Compute individual channel HOG features for the entire image: 0.126628 seconds
Process-1998, Step 6: Misc initializations: 0.0 seconds
Process-1998, Step 7: for loop: 0.007228 seconds

Process-1998All steps time: 0.14519 seconds

Process-1998, Find cars processing time: 0.145256 seconds

The times for each task are: [0.615101, 0.505873, 0.633557, 0.467439, 0.317797, 0.313679, 0.305811]

Minimum: 0.104892 Maximum: 0.633557 Average: 0.2659 seconds

Number of processes used: 2 window size 260
Length of task list: 21
Number of processes used: 2

Process-1999, Step 1: Divide with 255, processing time: 0.033184 seconds
Process-1999, Step 2: Resize if scale is not 1: 0.012711 seconds
Process-1999, Step 3: Get HOG channels: 8.6e-05 seconds

Process-1999, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.497136 seconds
Process-1999, Step 6: Misc initializations: 1e-06 seconds
Process-1999, Step 7: for loop: 0.035934 seconds

Process-1999All steps time: 0.579065 seconds

Process-1999, Find cars processing time: 0.579348 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.03304 seconds
Process-2000, Step 2: Resize if scale is not 1: 0.012636 seconds
Process-2000, Step 3: Get HOG channels: 0.000118 seconds
Process-2000, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.484947 seconds
Process-2000, Step 6: Misc initializations: 1e-06 seconds
Process-2000, Step 7: for loop: 0.034161 seconds

Process-2000All steps time: 0.5649160000000001 seconds

Process-2000, Find cars processing time: 0.56518 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.010244 seconds
Process-1999, Step 2: Resize if scale is not 1: 0.005778 seconds
Process-1999, Step 3: Get HOG channels: 1.2e-05 seconds
Process-1999, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.495393 seconds
Process-1999, Step 6: Misc initializations: 1e-06 seconds
Process-1999, Step 7: for loop: 0.03445 seconds

Process-1999All steps time: 0.5458879999999999 seconds

Process-1999, Find cars processing time: 0.545977 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.007711 seconds
Process-1999, Step 2: Resize if scale is not 1: 0.000758 seconds
Process-1999, Step 3: Get HOG channels: 6e-06 seconds
Process-1999, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.279011 seconds
Process-1999, Step 6: Misc initializations: 1e-06 seconds
Process-1999, Step 7: for loop: 0.018991 seconds

Process-1999All steps time: 0.306484 seconds

Process-1999, Find cars processing time: 0.30655 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.010057 seconds
Process-2000, Step 2: Resize if scale is not 1: 0.005153 seconds
Process-2000, Step 3: Get HOG channels: 1e-05 seconds

Process-2000, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.436143 seconds

Process-2000, Step 6: Misc initializations: 1e-06 seconds

Process-2000, Step 7: for loop: 0.038519 seconds

Process-2000All steps time: 0.489893 seconds

Process-2000, Find cars processing time: 0.489977 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.008577 seconds

Process-1999, Step 2: Resize if scale is not 1: 0.000944 seconds

Process-1999, Step 3: Get HOG channels: 8e-06 seconds

Process-1999, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.265754 seconds

Process-1999, Step 6: Misc initializations: 1e-06 seconds

Process-1999, Step 7: for loop: 0.01855 seconds

Process-1999All steps time: 0.2938429999999997 seconds

Process-1999, Find cars processing time: 0.293912 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.01054 seconds

Process-2000, Step 2: Resize if scale is not 1: 0.00125 seconds

Process-2000, Step 3: Get HOG channels: 5e-06 seconds

Process-2000, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.286546 seconds

Process-2000, Step 6: Misc initializations: 1e-06 seconds

Process-2000, Step 7: for loop: 0.018502 seconds

Process-2000All steps time: 0.316849 seconds

Process-2000, Find cars processing time: 0.316915 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.008518 seconds

Process-2000, Step 2: Resize if scale is not 1: 0.000724 seconds

Process-2000, Step 3: Get HOG channels: 7e-06 seconds

Process-2000, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.181825 seconds

Process-2000, Step 6: Misc initializations: 0.0 seconds

Process-2000, Step 7: for loop: 0.012426 seconds

Process-2000All steps time: 0.2035079999999997 seconds

Process-2000, Find cars processing time: 0.203579 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.010574 seconds

Process-1999, Step 2: Resize if scale is not 1: 0.001075 seconds

Process-1999, Step 3: Get HOG channels: 8e-06 seconds

Process-1999, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.274734 seconds

Process-1999, Step 6: Misc initializations: 1e-06 seconds

Process-1999, Step 7: for loop: 0.017558 seconds

Process-1999All steps time: 0.30395799999999995 seconds

Process-1999, Find cars processing time: 0.304025 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.010655 seconds

Process-2000, Step 2: Resize if scale is not 1: 0.00099 seconds

Process-2000, Step 3: Get HOG channels: 1e-05 seconds

Process-2000, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.198011 seconds

Process-2000, Step 6: Misc initializations: 1e-06 seconds

Process-2000, Step 7: for loop: 0.012687 seconds

Process-2000All steps time: 0.222364 seconds

Process-2000, Find cars processing time: 0.222454 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.010537 seconds

Process-1999, Step 2: Resize if scale is not 1: 0.000908 seconds

Process-1999, Step 3: Get HOG channels: 7e-06 seconds

Process-1999, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.228377 seconds

Process-1999, Step 6: Misc initializations: 1e-06 seconds

Process-1999, Step 7: for loop: 0.012715 seconds

Process-1999All steps time: 0.252552 seconds

Process-1999, Find cars processing time: 0.252613 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.007877 seconds

Process-2000, Step 2: Resize if scale is not 1: 0.000707 seconds

Process-2000, Step 3: Get HOG channels: 7e-06 seconds

Process-2000, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.184597 seconds

Process-2000, Step 6: Misc initializations: 1e-06 seconds

Process-2000, Step 7: for loop: 0.013781 seconds

Process-2000All steps time: 0.206977 seconds

Process-2000, Find cars processing time: 0.207035 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.010063 seconds

Process-2000, Step 2: Resize if scale is not 1: 0.000684 seconds

Process-2000, Step 3: Get HOG channels: 7e-06 seconds

Process-2000, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.176234 seconds

Process-2000, Step 6: Misc initializations: 0.0 seconds

Process-2000, Step 7: for loop: 0.009384 seconds

Process-2000All steps time: 0.196379 seconds

Process-2000, Find cars processing time: 0.19646 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.01053 seconds

Process-1999, Step 2: Resize if scale is not 1: 0.000523 seconds

Process-1999, Step 3: Get HOG channels: 5e-06 seconds

Process-1999, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.152625 seconds

Process-1999, Step 6: Misc initializations: 0.0 seconds

Process-1999, Step 7: for loop: 0.008156 seconds

Process-1999All steps time: 0.171844 seconds

Process-1999, Find cars processing time: 0.171912 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.009721 seconds

Process-1999, Step 2: Resize if scale is not 1: 0.000626 seconds

Process-1999, Step 3: Get HOG channels: 7e-06 seconds

Process-1999, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.180344 seconds

Process-1999, Step 6: Misc initializations: 1e-06 seconds

Process-1999, Step 7: for loop: 0.009876 seconds

Process-1999All steps time: 0.200581 seconds

Process-1999, Find cars processing time: 0.200663 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.010439 seconds

Process-2000, Step 2: Resize if scale is not 1: 0.000516 seconds

Process-2000, Step 3: Get HOG channels: 6e-06 seconds

Process-2000, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.179396 seconds

Process-2000, Step 6: Misc initializations: 1e-06 seconds

Process-2000, Step 7: for loop: 0.008053 seconds

Process-2000All steps time: 0.198417 seconds

Process-2000, Find cars processing time: 0.198486 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.010283 seconds

Process-1999, Step 2: Resize if scale is not 1: 0.001355 seconds

Process-1999, Step 3: Get HOG channels: 8e-06 seconds

Process-1999, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.098606 seconds
Process-1999, Step 6: Misc initializations: 1e-06 seconds
Process-1999, Step 7: for loop: 0.0055 seconds

Process-1999All steps time: 0.115761 seconds

Process-1999, Find cars processing time: 0.115829 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.008304 seconds
Process-1999, Step 2: Resize if scale is not 1: 0.000655 seconds
Process-1999, Step 3: Get HOG channels: 8e-06 seconds
Process-1999, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.102177 seconds
Process-1999, Step 6: Misc initializations: 0.0 seconds
Process-1999, Step 7: for loop: 0.009323 seconds

Process-1999All steps time: 0.120475 seconds

Process-1999, Find cars processing time: 0.120538 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.006807 seconds
Process-2000, Step 2: Resize if scale is not 1: 0.001064 seconds
Process-2000, Step 3: Get HOG channels: 6e-06 seconds
Process-2000, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.10457 seconds
Process-2000, Step 6: Misc initializations: 0.0 seconds
Process-2000, Step 7: for loop: 0.005973 seconds

Process-2000All steps time: 0.118426 seconds

Process-2000, Find cars processing time: 0.118513 seconds

Process-1999, Step 1: Divide with 255, processing time: 0.010679 seconds
Process-1999, Step 2: Resize if scale is not 1: 0.000796 seconds
Process-1999, Step 3: Get HOG channels: 9e-06 seconds
Process-1999, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-1999, Step 5: Compute individual channel HOG features for the entire image: 0.123008 seconds
Process-1999, Step 6: Misc initializations: 0.0 seconds
Process-1999, Step 7: for loop: 0.005537 seconds

Process-1999All steps time: 0.140038 seconds

Process-1999, Find cars processing time: 0.140117 seconds

Process-2000, Step 1: Divide with 255, processing time: 0.009112 seconds
Process-2000, Step 2: Resize if scale is not 1: 0.000746 seconds
Process-2000, Step 3: Get HOG channels: 7e-06 seconds

Process-2000, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2000, Step 5: Compute individual channel HOG features for the entire image: 0.087771 seconds

Process-2000, Step 6: Misc initializations: 1e-06 seconds

Process-2000, Step 7: for loop: 0.004481 seconds

Process-2000All steps time: 0.10212600000000001 seconds

Process-2000, Find cars processing time: 0.102182 seconds

The times for each task are: [0.579348, 0.56518, 0.545977, 0.30655, 0.489977, 0.293912, 0.316915]

Minimum: 0.102182 Maximum: 0.579348 Average: 0.2692 seconds

Number of processes used: 2 window size 260

Length of task list: 21

Number of processes used: 2

Process-2001, Step 1: Divide with 255, processing time: 0.028331 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.01155 seconds

Process-2001, Step 3: Get HOG channels: 0.000101 seconds

Process-2001, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.469722 seconds

Process-2001, Step 6: Misc initializations: 0.0 seconds

Process-2001, Step 7: for loop: 0.034163 seconds

Process-2001All steps time: 0.543879 seconds

Process-2001, Find cars processing time: 0.544126 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.032602 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.012687 seconds

Process-2002, Step 3: Get HOG channels: 9.4e-05 seconds

Process-2002, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.491375 seconds

Process-2002, Step 6: Misc initializations: 1e-06 seconds

Process-2002, Step 7: for loop: 0.035748 seconds

Process-2002All steps time: 0.5725180000000001 seconds

Process-2002, Find cars processing time: 0.5728 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.010156 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.004231 seconds

Process-2001, Step 3: Get HOG channels: 1.1e-05 seconds

Process-2001, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.447044 seconds

Process-2001, Step 6: Misc initializations: 1e-06 seconds

Process-2001, Step 7: for loop: 0.034016 seconds

Process-2001All steps time: 0.4954709999999994 seconds

Process-2001, Find cars processing time: 0.495547 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.010736 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.009717 seconds

Process-2002, Step 3: Get HOG channels: 1.2e-05 seconds

Process-2002, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.46976 seconds

Process-2002, Step 6: Misc initializations: 1e-06 seconds

Process-2002, Step 7: for loop: 0.029572 seconds

Process-2002All steps time: 0.519806 seconds

Process-2002, Find cars processing time: 0.519902 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.011043 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.001105 seconds

Process-2001, Step 3: Get HOG channels: 8e-06 seconds

Process-2001, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.27445 seconds

Process-2001, Step 6: Misc initializations: 1e-06 seconds

Process-2001, Step 7: for loop: 0.019466 seconds

Process-2001All steps time: 0.306081 seconds

Process-2001, Find cars processing time: 0.306164 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.010575 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.000792 seconds

Process-2001, Step 3: Get HOG channels: 7e-06 seconds

Process-2001, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.398524 seconds

Process-2001, Step 6: Misc initializations: 1e-06 seconds

Process-2001, Step 7: for loop: 0.021139 seconds

Process-2001All steps time: 0.4310469999999996 seconds

Process-2001, Find cars processing time: 0.431136 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.010205 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.001644 seconds

Process-2002, Step 3: Get HOG channels: 8e-06 seconds

Process-2002, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.286483 seconds

Process-2002, Step 6: Misc initializations: 1e-06 seconds

Process-2002, Step 7: for loop: 0.023041 seconds

Process-2002All steps time: 0.3213899999999995 seconds

Process-2002, Find cars processing time: 0.321455 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.011139 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.000961 seconds

Process-2002, Step 3: Get HOG channels: 9e-06 seconds

Process-2002, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.18872 seconds

Process-2002, Step 6: Misc initializations: 1e-06 seconds

Process-2002, Step 7: for loop: 0.012414 seconds

Process-2002All steps time: 0.213252 seconds

Process-2002, Find cars processing time: 0.213336 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.01077 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.000712 seconds

Process-2001, Step 3: Get HOG channels: 7e-06 seconds

Process-2001, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.314022 seconds

Process-2001, Step 6: Misc initializations: 1e-06 seconds

Process-2001, Step 7: for loop: 0.035364 seconds

Process-2001All steps time: 0.360883 seconds

Process-2001, Find cars processing time: 0.360968 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.009576 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.000882 seconds

Process-2002, Step 3: Get HOG channels: 8e-06 seconds

Process-2002, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.197964 seconds

Process-2002, Step 6: Misc initializations: 0.0 seconds

Process-2002, Step 7: for loop: 0.012299 seconds

Process-2002All steps time: 0.2207370000000002 seconds

Process-2002, Find cars processing time: 0.220806 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.009853 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.00088 seconds

Process-2001, Step 3: Get HOG channels: 9e-06 seconds

Process-2001, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.223509 seconds

Process-2001, Step 6: Misc initializations: 1e-06 seconds

Process-2001, Step 7: for loop: 0.013704 seconds

Process-2001All steps time: 0.24796500000000002 seconds

Process-2001, Find cars processing time: 0.248044 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.010623 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.00093 seconds

Process-2002, Step 3: Get HOG channels: 8e-06 seconds

Process-2002, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.179214 seconds

Process-2002, Step 6: Misc initializations: 0.0 seconds

Process-2002, Step 7: for loop: 0.011944 seconds

Process-2002All steps time: 0.20272700000000002 seconds

Process-2002, Find cars processing time: 0.202802 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.009513 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.000867 seconds

Process-2001, Step 3: Get HOG channels: 8e-06 seconds

Process-2001, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.149367 seconds

Process-2001, Step 6: Misc initializations: 1e-06 seconds

Process-2001, Step 7: for loop: 0.008013 seconds

Process-2001All steps time: 0.1677769999999998 seconds

Process-2001, Find cars processing time: 0.167846 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.008942 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.000782 seconds

Process-2001, Step 3: Get HOG channels: 8e-06 seconds

Process-2001, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.16833 seconds

Process-2001, Step 6: Misc initializations: 0.0 seconds

Process-2001, Step 7: for loop: 0.008929 seconds

Process-2001All steps time: 0.186999 seconds

Process-2001, Find cars processing time: 0.18707 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.01058 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.000877 seconds

Process-2002, Step 3: Get HOG channels: 8e-06 seconds

Process-2002, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.13437 seconds

Process-2002, Step 6: Misc initializations: 1e-06 seconds

Process-2002, Step 7: for loop: 0.00746 seconds

Process-2002All steps time: 0.153303 seconds

Process-2002, Find cars processing time: 0.153365 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.006943 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.000809 seconds

Process-2002, Step 3: Get HOG channels: 6e-06 seconds

Process-2002, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.107125 seconds

Process-2002, Step 6: Misc initializations: 1e-06 seconds

Process-2002, Step 7: for loop: 0.006061 seconds

Process-2002All steps time: 0.12095199999999999 seconds

Process-2002, Find cars processing time: 0.121023 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.009677 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.000487 seconds

Process-2001, Step 3: Get HOG channels: 5e-06 seconds

Process-2001, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.159359 seconds

Process-2001, Step 6: Misc initializations: 0.0 seconds

Process-2001, Step 7: for loop: 0.013872 seconds

Process-2001All steps time: 0.1834049999999998 seconds

Process-2001, Find cars processing time: 0.183476 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.010351 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.000762 seconds

Process-2002, Step 3: Get HOG channels: 8e-06 seconds

Process-2002, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.095971 seconds

Process-2002, Step 6: Misc initializations: 0.0 seconds

Process-2002, Step 7: for loop: 0.005836 seconds

Process-2002All steps time: 0.112936 seconds

Process-2002, Find cars processing time: 0.113006 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.010581 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.001477 seconds

Process-2001, Step 3: Get HOG channels: 9e-06 seconds

Process-2001, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.131981 seconds

Process-2001, Step 6: Misc initializations: 0.0 seconds

Process-2001, Step 7: for loop: 0.006381 seconds

Process-2001All steps time: 0.150437 seconds

Process-2001, Find cars processing time: 0.150511 seconds

Process-2002, Step 1: Divide with 255, processing time: 0.007995 seconds

Process-2002, Step 2: Resize if scale is not 1: 0.000581 seconds

Process-2002, Step 3: Get HOG channels: 5e-06 seconds

Process-2002, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2002, Step 5: Compute individual channel HOG features for the entire image: 0.107432 seconds

Process-2002, Step 6: Misc initializations: 1e-06 seconds

Process-2002, Step 7: for loop: 0.006429 seconds

Process-2002All steps time: 0.122449 seconds

Process-2002, Find cars processing time: 0.122519 seconds

Process-2001, Step 1: Divide with 255, processing time: 0.010763 seconds

Process-2001, Step 2: Resize if scale is not 1: 0.001087 seconds

Process-2001, Step 3: Get HOG channels: 1e-05 seconds

Process-2001, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2001, Step 5: Compute individual channel HOG features for the entire image: 0.108337 seconds

Process-2001, Step 6: Misc initializations: 1e-06 seconds

Process-2001, Step 7: for loop: 0.00508 seconds

Process-2001All steps time: 0.125288 seconds

Process-2001, Find cars processing time: 0.125408 seconds

The times for each task are: [0.544126, 0.5728, 0.495547, 0.519902, 0.306164, 0.431136, 0.321455]

Minimum: 0.113006 Maximum: 0.5728 Average: 0.2743 seconds

Number of processes used: 2 window size 260

Length of task list: 21

Number of processes used: 2

Process-2003, Step 1: Divide with 255, processing time: 0.032529 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.012198 seconds

Process-2003, Step 3: Get HOG channels: 0.000105 seconds

Process-2003, Step 4: Define blocks and steps as above: 1.5e-05 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.523829 seconds

Process-2003, Step 6: Misc initializations: 0.0 seconds

Process-2003, Step 7: for loop: 0.038322 seconds

Process-2003All steps time: 0.6069979999999999 seconds

Process-2003, Find cars processing time: 0.607296 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.032858 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.011894 seconds

Process-2004, Step 3: Get HOG channels: 0.000102 seconds

Process-2004, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.510686 seconds

Process-2004, Step 6: Misc initializations: 1e-06 seconds

Process-2004, Step 7: for loop: 0.03597 seconds

Process-2004All steps time: 0.59152 seconds

Process-2004, Find cars processing time: 0.591791 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.008528 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.004604 seconds

Process-2003, Step 3: Get HOG channels: 1.2e-05 seconds

Process-2003, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.518959 seconds

Process-2003, Step 6: Misc initializations: 0.0 seconds

Process-2003, Step 7: for loop: 0.035313 seconds

Process-2003All steps time: 0.567426 seconds

Process-2003, Find cars processing time: 0.567512 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.009415 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.001286 seconds

Process-2003, Step 3: Get HOG channels: 1.4e-05 seconds

Process-2003, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.327215 seconds

Process-2003, Step 6: Misc initializations: 1e-06 seconds

Process-2003, Step 7: for loop: 0.019747 seconds

Process-2003All steps time: 0.357691 seconds

Process-2003, Find cars processing time: 0.357795 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.010284 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.004271 seconds

Process-2004, Step 3: Get HOG channels: 1e-05 seconds

Process-2004, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.44418 seconds

Process-2004, Step 6: Misc initializations: 1e-06 seconds

Process-2004, Step 7: for loop: 0.02884 seconds

Process-2004All steps time: 0.487596 seconds

Process-2004, Find cars processing time: 0.487677 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.010888 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.001763 seconds

Process-2004, Step 3: Get HOG channels: 7e-06 seconds

Process-2004, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.335749 seconds

Process-2004, Step 6: Misc initializations: 1e-06 seconds

Process-2004, Step 7: for loop: 0.019344 seconds

Process-2004All steps time: 0.3677589999999995 seconds

Process-2004, Find cars processing time: 0.367858 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.010351 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.001059 seconds

Process-2003, Step 3: Get HOG channels: 1e-05 seconds

Process-2003, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.314476 seconds

Process-2003, Step 6: Misc initializations: 1e-06 seconds

Process-2003, Step 7: for loop: 0.021807 seconds

Process-2003All steps time: 0.347713 seconds

Process-2003, Find cars processing time: 0.347793 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.007727 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.000686 seconds

Process-2003, Step 3: Get HOG channels: 9e-06 seconds

Process-2003, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.245382 seconds

Process-2003, Step 6: Misc initializations: 1e-06 seconds

Process-2003, Step 7: for loop: 0.013798 seconds

Process-2003All steps time: 0.2676099999999996 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.010789 seconds

Process-2003, Find cars processing time: 0.267716 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.00088 seconds

Process-2004, Step 3: Get HOG channels: 6e-06 seconds

Process-2004, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.271271 seconds

Process-2004, Step 6: Misc initializations: 1e-06 seconds

Process-2004, Step 7: for loop: 0.0197 seconds

Process-2004All steps time: 0.302654 seconds

Process-2004, Find cars processing time: 0.302727 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.009997 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.000823 seconds

Process-2003, Step 3: Get HOG channels: 8e-06 seconds

Process-2003, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.208959 seconds

Process-2003, Step 6: Misc initializations: 1e-06 seconds

Process-2003, Step 7: for loop: 0.015358 seconds

Process-2003All steps time: 0.23515300000000003 seconds

Process-2003, Find cars processing time: 0.235265 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.024266 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.000456 seconds

Process-2004, Step 3: Get HOG channels: 4e-06 seconds

Process-2004, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.200753 seconds

Process-2004, Step 6: Misc initializations: 1e-06 seconds

Process-2004, Step 7: for loop: 0.013473 seconds

Process-2004All steps time: 0.238958 seconds

Process-2004, Find cars processing time: 0.239012 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.010335 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.000838 seconds

Process-2004, Step 3: Get HOG channels: 8e-06 seconds

Process-2004, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.144805 seconds

Process-2004, Step 6: Misc initializations: 1e-06 seconds

Process-2004, Step 7: for loop: 0.007896 seconds

Process-2004All steps time: 0.1638899999999998 seconds

Process-2004, Find cars processing time: 0.163957 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.009153 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.00082 seconds

Process-2003, Step 3: Get HOG channels: 7e-06 seconds

Process-2003, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.24721 seconds

Process-2003, Step 6: Misc initializations: 1e-06 seconds

Process-2003, Step 7: for loop: 0.015859 seconds

Process-2003All steps time: 0.273057 seconds

Process-2003, Find cars processing time: 0.273152 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.010737 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.000772 seconds

Process-2004, Step 3: Get HOG channels: 8e-06 seconds

Process-2004, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.180152 seconds

Process-2004, Step 6: Misc initializations: 2e-06 seconds

Process-2004, Step 7: for loop: 0.010094 seconds

Process-2004All steps time: 0.201772 seconds

Process-2004, Find cars processing time: 0.201868 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.010626 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.001009 seconds

Process-2003, Step 3: Get HOG channels: 1.9e-05 seconds

Process-2003, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.136577 seconds

Process-2003, Step 6: Misc initializations: 1e-06 seconds

Process-2003, Step 7: for loop: 0.007584 seconds

Process-2003All steps time: 0.1558240000000002 seconds

Process-2003, Find cars processing time: 0.155924 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.011739 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.000694 seconds

Process-2003, Step 3: Get HOG channels: 5e-06 seconds

Process-2003, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.100194 seconds

Process-2003, Step 6: Misc initializations: 0.0 seconds

Process-2003, Step 7: for loop: 0.006378 seconds

Process-2003All steps time: 0.119015 seconds

Process-2003, Find cars processing time: 0.119072 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.010325 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.000588 seconds

Process-2004, Step 3: Get HOG channels: 8e-06 seconds

Process-2004, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.165209 seconds

Process-2004, Step 6: Misc initializations: 0.0 seconds

Process-2004, Step 7: for loop: 0.008145 seconds

Process-2004All steps time: 0.1842809999999997 seconds

Process-2004, Find cars processing time: 0.184345 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.011076 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.00093 seconds

Process-2004, Step 3: Get HOG channels: 6e-06 seconds

Process-2004, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.10436 seconds

Process-2004, Step 6: Misc initializations: 0.0 seconds

Process-2004, Step 7: for loop: 0.006282 seconds

Process-2004All steps time: 0.12266099999999999 seconds

Process-2004, Find cars processing time: 0.122731 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.016666 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.000702 seconds

Process-2003, Step 3: Get HOG channels: 6e-06 seconds

Process-2003, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.129765 seconds

Process-2003, Step 6: Misc initializations: 0.0 seconds

Process-2003, Step 7: for loop: 0.007184 seconds

Process-2003All steps time: 0.154329 seconds

Process-2003, Find cars processing time: 0.154393 seconds

Process-2004, Step 1: Divide with 255, processing time: 0.018737 seconds

Process-2004, Step 2: Resize if scale is not 1: 0.00071 seconds

Process-2004, Step 3: Get HOG channels: 2.2e-05 seconds

Process-2004, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2004, Step 5: Compute individual channel HOG features for the entire image: 0.097989 seconds

Process-2004, Step 6: Misc initializations: 0.0 seconds

Process-2004, Step 7: for loop: 0.005933 seconds

Process-2004All steps time: 0.123397 seconds

Process-2004, Find cars processing time: 0.123456 seconds

Process-2003, Step 1: Divide with 255, processing time: 0.009737 seconds

Process-2003, Step 2: Resize if scale is not 1: 0.000745 seconds

Process-2003, Step 3: Get HOG channels: 5e-06 seconds

Process-2003, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2003, Step 5: Compute individual channel HOG features for the entire image: 0.11268 seconds

Process-2003, Step 6: Misc initializations: 1e-06 seconds

Process-2003, Step 7: for loop: 0.005994 seconds

Process-2003All steps time: 0.129168 seconds

Process-2003, Find cars processing time: 0.129235 seconds

The times for each task are: [0.607296, 0.591791, 0.567512, 0.357795, 0.487677, 0.367858, 0.3477

Minimum: 0.119072 Maximum: 0.607296 Average: 0.2857 seconds

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

2 processes used for testing 5 window sizes

Processing times for each image [6.7674, 6.5616, 6.2889, 6.423, 6.8113, 6.702] with an average o

Time elapsed so far... 203.5842

#####

Number of processes used: 3 window size 260

Length of task list: 21

Number of processes used: 3

Process-2005, Step 1: Divide with 255, processing time: 0.035084 seconds

Process-2005, Step 2: Resize if scale is not 1: 0.012403 seconds

Process-2005, Step 3: Get HOG channels: 9.6e-05 seconds

Process-2005, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-2005, Step 5: Compute individual channel HOG features for the entire image: 0.489255 sec

Process-2005, Step 6: Misc initializations: 1e-06 seconds

Process-2005, Step 7: for loop: 0.036293 seconds

Process-2005All steps time: 0.5731440000000001 seconds

Process-2005, Find cars processing time: 0.573428 seconds

Process-2006, Step 1: Divide with 255, processing time: 0.029256 seconds

Process-2006, Step 2: Resize if scale is not 1: 0.011434 seconds

Process-2006, Step 3: Get HOG channels: 0.000125 seconds

Process-2006, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2006, Step 5: Compute individual channel HOG features for the entire image: 0.462585 sec

Process-2006, Step 6: Misc initializations: 1e-06 seconds

Process-2006, Step 7: for loop: 0.038077 seconds

Process-2006All steps time: 0.5414860000000001 seconds

Process-2006, Find cars processing time: 0.541732 seconds

Process-2007, Step 1: Divide with 255, processing time: 0.036857 seconds

Process-2007, Step 2: Resize if scale is not 1: 0.013958 seconds

Process-2007, Step 3: Get HOG channels: 0.000217 seconds

Process-2007, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-2007, Step 5: Compute individual channel HOG features for the entire image: 0.458943 sec

Process-2007, Step 6: Misc initializations: 0.0 seconds
Process-2007, Step 7: for loop: 0.036949 seconds

Process-2007All steps time: 0.546935 seconds

Process-2007, Find cars processing time: 0.547241 seconds

Process-2005, Step 1: Divide with 255, processing time: 0.006602 seconds
Process-2005, Step 2: Resize if scale is not 1: 0.003124 seconds
Process-2005, Step 3: Get HOG channels: 8e-06 seconds
Process-2005, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2005, Step 5: Compute individual channel HOG features for the entire image: 0.560376 seconds
Process-2005, Step 6: Misc initializations: 1e-06 seconds
Process-2005, Step 7: for loop: 0.033756 seconds

Process-2005All steps time: 0.603876 seconds

Process-2005, Find cars processing time: 0.603948 seconds

Process-2006, Step 1: Divide with 255, processing time: 0.008319 seconds
Process-2006, Step 2: Resize if scale is not 1: 0.001026 seconds
Process-2006, Step 3: Get HOG channels: 8e-06 seconds
Process-2006, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2006, Step 5: Compute individual channel HOG features for the entire image: 0.351995 seconds
Process-2006, Step 6: Misc initializations: 1e-06 seconds
Process-2006, Step 7: for loop: 0.019673 seconds

Process-2006All steps time: 0.381032 seconds

Process-2006, Find cars processing time: 0.38112 seconds

Process-2005, Step 1: Divide with 255, processing time: 0.008567 seconds
Process-2006, Step 1: Divide with 255, processing time: 0.008126 seconds
Process-2005, Step 2: Resize if scale is not 1: 0.001495 seconds
Process-2006, Step 2: Resize if scale is not 1: 0.000903 seconds
Process-2005, Step 3: Get HOG channels: 8e-06 seconds
Process-2006, Step 3: Get HOG channels: 6e-06 seconds
Process-2005, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2006, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2005, Step 5: Compute individual channel HOG features for the entire image: 0.345236 seconds
Process-2006, Step 5: Compute individual channel HOG features for the entire image: 0.265144 seconds
Process-2005, Step 6: Misc initializations: 1e-06 seconds
Process-2006, Step 6: Misc initializations: 1e-06 seconds
Process-2005, Step 7: for loop: 0.020699 seconds
Process-2006, Step 7: for loop: 0.019707 seconds

Process-2005All steps time: 0.376013 seconds

Process-2006All steps time: 0.29389399999999993 seconds

Process-2005, Find cars processing time: 0.376092 seconds
Process-2006, Find cars processing time: 0.293948 seconds

Process-2007, Step 1: Divide with 255, processing time: 0.010876 seconds
Process-2007, Step 2: Resize if scale is not 1: 0.000935 seconds
Process-2007, Step 3: Get HOG channels: 5e-06 seconds
Process-2007, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2007, Step 5: Compute individual channel HOG features for the entire image: 0.328696 seconds
Process-2007, Step 6: Misc initializations: 1e-06 seconds
Process-2007, Step 7: for loop: 0.019649 seconds

Process-2007All steps time: 0.3601679999999993 seconds

Process-2007, Find cars processing time: 0.360238 seconds

Process-2005, Step 1: Divide with 255, processing time: 0.008695 seconds
Process-2005, Step 2: Resize if scale is not 1: 0.000641 seconds
Process-2005, Step 3: Get HOG channels: 6e-06 seconds
Process-2005, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2005, Step 5: Compute individual channel HOG features for the entire image: 0.225797 seconds
Process-2005, Step 6: Misc initializations: 0.0 seconds
Process-2005, Step 7: for loop: 0.012768 seconds

Process-2005All steps time: 0.247914 seconds

Process-2005, Find cars processing time: 0.247983 seconds

Process-2007, Step 1: Divide with 255, processing time: 0.010735 seconds
Process-2007, Step 2: Resize if scale is not 1: 0.000963 seconds
Process-2007, Step 3: Get HOG channels: 7e-06 seconds
Process-2007, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2007, Step 5: Compute individual channel HOG features for the entire image: 0.198214 seconds
Process-2007, Step 6: Misc initializations: 1e-06 seconds
Process-2007, Step 7: for loop: 0.01305 seconds

Process-2007All steps time: 0.222979 seconds

Process-2007, Find cars processing time: 0.223069 seconds

Process-2006, Step 1: Divide with 255, processing time: 0.010307 seconds
Process-2006, Step 2: Resize if scale is not 1: 0.000503 seconds
Process-2006, Step 3: Get HOG channels: 5e-06 seconds
Process-2006, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2006, Step 5: Compute individual channel HOG features for the entire image: 0.201657 seconds

Process-2006, Step 6: Misc initializations: 1e-06 seconds
Process-2006, Step 7: for loop: 0.014468 seconds

Process-2006All steps time: 0.226946 seconds

Process-2006, Find cars processing time: 0.227034 seconds

Process-2005, Step 1: Divide with 255, processing time: 0.010419 seconds
Process-2005, Step 2: Resize if scale is not 1: 0.000596 seconds
Process-2005, Step 3: Get HOG channels: 5e-06 seconds
Process-2005, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2005, Step 5: Compute individual channel HOG features for the entire image: 0.224665 seconds
Process-2005, Step 6: Misc initializations: 1e-06 seconds
Process-2005, Step 7: for loop: 0.014862 seconds

Process-2005All steps time: 0.250554 seconds

Process-2005, Find cars processing time: 0.250642 seconds

Process-2007, Step 1: Divide with 255, processing time: 0.007707 seconds
Process-2007, Step 2: Resize if scale is not 1: 0.000701 seconds
Process-2007, Step 3: Get HOG channels: 1.9e-05 seconds
Process-2007, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2007, Step 5: Compute individual channel HOG features for the entire image: 0.143881 seconds
Process-2007, Step 6: Misc initializations: 0.0 seconds
Process-2007, Step 7: for loop: 0.008276 seconds

Process-2007All steps time: 0.16059 seconds

Process-2007, Find cars processing time: 0.160676 seconds

Process-2006, Step 1: Divide with 255, processing time: 0.010681 seconds
Process-2006, Step 2: Resize if scale is not 1: 0.000972 seconds
Process-2006, Step 3: Get HOG channels: 1.7e-05 seconds
Process-2006, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2006, Step 5: Compute individual channel HOG features for the entire image: 0.140809 seconds
Process-2006, Step 6: Misc initializations: 0.0 seconds
Process-2006, Step 7: for loop: 0.00921 seconds

Process-2006All steps time: 0.1616969999999998 seconds

Process-2006, Find cars processing time: 0.161828 seconds

Process-2005, Step 1: Divide with 255, processing time: 0.010923 seconds
Process-2005, Step 2: Resize if scale is not 1: 0.001139 seconds
Process-2005, Step 3: Get HOG channels: 1e-05 seconds
Process-2005, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2005, Step 5: Compute individual channel HOG features for the entire image: 0.178427 seconds

Process-2005, Step 6: Misc initializations: 1e-06 seconds
Process-2005, Step 7: for loop: 0.011013 seconds

Process-2005All steps time: 0.201522 seconds

Process-2005, Find cars processing time: 0.201617 seconds

Process-2006, Step 1: Divide with 255, processing time: 0.011481 seconds
Process-2006, Step 2: Resize if scale is not 1: 0.000865 seconds
Process-2006, Step 3: Get HOG channels: 6e-06 seconds
Process-2006, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2006, Step 5: Compute individual channel HOG features for the entire image: 0.133777 seconds
Process-2006, Step 6: Misc initializations: 1e-06 seconds
Process-2006, Step 7: for loop: 0.00746 seconds

Process-2006All steps time: 0.153597 seconds

Process-2006, Find cars processing time: 0.153677 seconds

Process-2007, Step 1: Divide with 255, processing time: 0.010656 seconds
Process-2007, Step 2: Resize if scale is not 1: 0.000863 seconds
Process-2007, Step 3: Get HOG channels: 9e-06 seconds
Process-2007, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2007, Step 5: Compute individual channel HOG features for the entire image: 0.163345 seconds
Process-2007, Step 6: Misc initializations: 0.0 seconds
Process-2007, Step 7: for loop: 0.008717 seconds

Process-2007All steps time: 0.1835979999999998 seconds

Process-2007, Find cars processing time: 0.183674 seconds

Process-2005, Step 1: Divide with 255, processing time: 0.018878 seconds
Process-2005, Step 2: Resize if scale is not 1: 0.000867 seconds
Process-2006, Step 1: Divide with 255, processing time: 0.009315 seconds
Process-2005, Step 3: Get HOG channels: 5e-06 seconds
Process-2006, Step 2: Resize if scale is not 1: 0.000512 seconds
Process-2005, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2006, Step 3: Get HOG channels: 5e-06 seconds
Process-2005, Step 5: Compute individual channel HOG features for the entire image: 0.129703 seconds
Process-2005, Step 6: Misc initializations: 1e-06 seconds
Process-2006, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2006, Step 5: Compute individual channel HOG features for the entire image: 0.123581 seconds
Process-2005, Step 7: for loop: 0.007143 seconds
Process-2006, Step 6: Misc initializations: 1e-06 seconds

Process-2006, Step 7: for loop: 0.006236 seconds

Process-2005All steps time: 0.1566020000000002 seconds

Process-2005, Find cars processing time: 0.156658 seconds
Process-2006All steps time: 0.139654 seconds

Process-2006, Find cars processing time: 0.13971 seconds

Process-2005, Step 1: Divide with 255, processing time: 0.010656 seconds
Process-2005, Step 2: Resize if scale is not 1: 0.001045 seconds
Process-2005, Step 3: Get HOG channels: 8e-06 seconds
Process-2005, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2005, Step 5: Compute individual channel HOG features for the entire image: 0.116421 seconds
Process-2005, Step 6: Misc initializations: 0.0 seconds
Process-2005, Step 7: for loop: 0.007655 seconds

Process-2005All steps time: 0.135794 seconds

Process-2005, Find cars processing time: 0.13588 seconds

Process-2007, Step 1: Divide with 255, processing time: 0.035599 seconds
Process-2007, Step 2: Resize if scale is not 1: 0.002013 seconds
Process-2007, Step 3: Get HOG channels: 7e-06 seconds
Process-2007, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2007, Step 5: Compute individual channel HOG features for the entire image: 0.152709 seconds
Process-2007, Step 6: Misc initializations: 1e-06 seconds
Process-2007, Step 7: for loop: 0.007526 seconds

Process-2007All steps time: 0.19786 seconds

Process-2007, Find cars processing time: 0.197929 seconds

The times for each task are: [0.573428, 0.541732, 0.547241, 0.603948, 0.38112, 0.376092, 0.29394

Minimum: 0.13588 Maximum: 0.603948 Average: 0.2913 seconds

Number of processes used: 3 window size 260
Length of task list: 21
Number of processes used: 3

Process-2009, Step 1: Divide with 255, processing time: 0.052713 seconds
Process-2009, Step 2: Resize if scale is not 1: 0.01323 seconds
Process-2009, Step 3: Get HOG channels: 0.000128 seconds
Process-2009, Step 4: Define blocks and steps as above: 1.4e-05 seconds
Process-2009, Step 5: Compute individual channel HOG features for the entire image: 0.62633 seconds
Process-2009, Step 6: Misc initializations: 1e-06 seconds
Process-2009, Step 7: for loop: 0.049447 seconds

Process-2009All steps time: 0.741863 seconds

Process-2009, Find cars processing time: 0.742456 seconds

Process-2008, Step 1: Divide with 255, processing time: 0.033417 seconds

Process-2008, Step 2: Resize if scale is not 1: 0.015241 seconds

Process-2008, Step 3: Get HOG channels: 8.6e-05 seconds

Process-2008, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-2008, Step 5: Compute individual channel HOG features for the entire image: 0.594595 seconds

Process-2008, Step 6: Misc initializations: 1e-06 seconds

Process-2008, Step 7: for loop: 0.044703 seconds

Process-2008All steps time: 0.6880560000000001 seconds

Process-2008, Find cars processing time: 0.688395 seconds

Process-2010, Step 1: Divide with 255, processing time: 0.027066 seconds

Process-2010, Step 2: Resize if scale is not 1: 0.013866 seconds

Process-2010, Step 3: Get HOG channels: 6.3e-05 seconds

Process-2010, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2010, Step 5: Compute individual channel HOG features for the entire image: 0.530909 seconds

Process-2010, Step 6: Misc initializations: 1e-06 seconds

Process-2010, Step 7: for loop: 0.036214 seconds

Process-2010All steps time: 0.6081259999999999 seconds

Process-2010, Find cars processing time: 0.608345 seconds

Process-2009, Step 1: Divide with 255, processing time: 0.008444 seconds

Process-2009, Step 2: Resize if scale is not 1: 0.00527 seconds

Process-2009, Step 3: Get HOG channels: 1.6e-05 seconds

Process-2009, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-2009, Step 5: Compute individual channel HOG features for the entire image: 0.551068 seconds

Process-2009, Step 6: Misc initializations: 1e-06 seconds

Process-2009, Step 7: for loop: 0.042709 seconds

Process-2009All steps time: 0.607522 seconds

Process-2009, Find cars processing time: 0.607641 seconds

Process-2010, Step 1: Divide with 255, processing time: 0.010103 seconds

Process-2010, Step 2: Resize if scale is not 1: 0.000932 seconds

Process-2010, Step 3: Get HOG channels: 8e-06 seconds

Process-2010, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2010, Step 5: Compute individual channel HOG features for the entire image: 0.366128 seconds

Process-2010, Step 6: Misc initializations: 0.0 seconds

Process-2010, Step 7: for loop: 0.019828 seconds

Process-2010All steps time: 0.397006 seconds

Process-2010, Find cars processing time: 0.397096 seconds

Process-2008, Step 1: Divide with 255, processing time: 0.010494 seconds

Process-2008, Step 2: Resize if scale is not 1: 0.001107 seconds

Process-2008, Step 3: Get HOG channels: 8e-06 seconds

Process-2008, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2008, Step 5: Compute individual channel HOG features for the entire image: 0.336471 seconds

Process-2008, Step 6: Misc initializations: 1e-06 seconds

Process-2008, Step 7: for loop: 0.022194 seconds

Process-2008All steps time: 0.370283 seconds

Process-2008, Find cars processing time: 0.370404 seconds

Process-2009, Step 1: Divide with 255, processing time: 0.017848 seconds

Process-2009, Step 2: Resize if scale is not 1: 0.001157 seconds

Process-2009, Step 3: Get HOG channels: 6e-06 seconds

Process-2009, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2009, Step 5: Compute individual channel HOG features for the entire image: 0.399832 seconds

Process-2009, Step 6: Misc initializations: 2e-06 seconds

Process-2009, Step 7: for loop: 0.02527 seconds

Process-2009All steps time: 0.4441210000000004 seconds

Process-2009, Find cars processing time: 0.444202 seconds

Process-2010, Step 1: Divide with 255, processing time: 0.016185 seconds

Process-2010, Step 2: Resize if scale is not 1: 0.000821 seconds

Process-2010, Step 3: Get HOG channels: 6e-06 seconds

Process-2010, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2010, Step 5: Compute individual channel HOG features for the entire image: 0.283848 seconds

Process-2010, Step 6: Misc initializations: 1e-06 seconds

Process-2010, Step 7: for loop: 0.018618 seconds

Process-2010All steps time: 0.319484 seconds

Process-2010, Find cars processing time: 0.319554 seconds

Process-2008, Step 1: Divide with 255, processing time: 0.022135 seconds

Process-2008, Step 2: Resize if scale is not 1: 0.000656 seconds

Process-2008, Step 3: Get HOG channels: 5e-06 seconds

Process-2008, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2008, Step 5: Compute individual channel HOG features for the entire image: 0.19631 seconds

Process-2008, Step 6: Misc initializations: 0.0 seconds

Process-2008, Step 7: for loop: 0.012172 seconds

Process-2008All steps time: 0.23128200000000002 seconds

Process-2008, Find cars processing time: 0.231371 seconds

Process-2010, Step 1: Divide with 255, processing time: 0.010205 seconds

Process-2010, Step 2: Resize if scale is not 1: 0.00101 seconds

Process-2010, Step 3: Get HOG channels: 9e-06 seconds

Process-2010, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2010, Step 5: Compute individual channel HOG features for the entire image: 0.253421 seconds

Process-2010, Step 6: Misc initializations: 1e-06 seconds

Process-2010, Step 7: for loop: 0.013522 seconds

Process-2010All steps time: 0.2781749999999995 seconds

Process-2010, Find cars processing time: 0.278247 seconds

Process-2009, Step 1: Divide with 255, processing time: 0.010295 seconds

Process-2009, Step 2: Resize if scale is not 1: 0.000841 seconds

Process-2009, Step 3: Get HOG channels: 6e-06 seconds

Process-2009, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2009, Step 5: Compute individual channel HOG features for the entire image: 0.247742 seconds

Process-2009, Step 6: Misc initializations: 1e-06 seconds

Process-2009, Step 7: for loop: 0.016223 seconds

Process-2009All steps time: 0.2751139999999997 seconds

Process-2009, Find cars processing time: 0.275186 seconds

Process-2008, Step 1: Divide with 255, processing time: 0.010354 seconds

Process-2008, Step 2: Resize if scale is not 1: 0.000905 seconds

Process-2008, Step 3: Get HOG channels: 5e-06 seconds

Process-2008, Step 4: Define blocks and steps as above: 2.1e-05 seconds

Process-2008, Step 5: Compute individual channel HOG features for the entire image: 0.198053 seconds

Process-2008, Step 6: Misc initializations: 1e-06 seconds

Process-2008, Step 7: for loop: 0.014678 seconds

Process-2008All steps time: 0.224017 seconds

Process-2008, Find cars processing time: 0.224104 seconds

Process-2010, Step 1: Divide with 255, processing time: 0.010318 seconds

Process-2010, Step 2: Resize if scale is not 1: 0.000873 seconds

Process-2010, Step 3: Get HOG channels: 8e-06 seconds

Process-2010, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2010, Step 5: Compute individual channel HOG features for the entire image: 0.130581 seconds

Process-2010, Step 6: Misc initializations: 0.0 seconds

Process-2010, Step 7: for loop: 0.007635 seconds

Process-2010All steps time: 0.149423 seconds

Process-2010, Find cars processing time: 0.149484 seconds

Process-2008, Step 1: Divide with 255, processing time: 0.02402 seconds

Process-2008, Step 2: Resize if scale is not 1: 0.001302 seconds

Process-2008, Step 3: Get HOG channels: 5e-06 seconds

Process-2008, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2008, Step 5: Compute individual channel HOG features for the entire image: 0.133047 seconds

Process-2008, Step 6: Misc initializations: 1e-06 seconds

Process-2008, Step 7: for loop: 0.010595 seconds

Process-2008All steps time: 0.1689739999999999 seconds

Process-2008, Find cars processing time: 0.169039 seconds

Process-2009, Step 1: Divide with 255, processing time: 0.011632 seconds

Process-2009, Step 2: Resize if scale is not 1: 0.000846 seconds

Process-2009, Step 3: Get HOG channels: 8e-06 seconds

Process-2009, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2009, Step 5: Compute individual channel HOG features for the entire image: 0.172495 seconds

Process-2009, Step 6: Misc initializations: 0.0 seconds

Process-2009, Step 7: for loop: 0.009906 seconds

Process-2009All steps time: 0.194894 seconds

Process-2009, Find cars processing time: 0.194969 seconds

Process-2008, Step 1: Divide with 255, processing time: 0.011379 seconds

Process-2008, Step 2: Resize if scale is not 1: 0.001879 seconds

Process-2008, Step 3: Get HOG channels: 1.7e-05 seconds

Process-2008, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-2008, Step 5: Compute individual channel HOG features for the entire image: 0.149867 seconds

Process-2008, Step 6: Misc initializations: 1e-06 seconds

Process-2008, Step 7: for loop: 0.00762 seconds

Process-2008All steps time: 0.1707759999999998 seconds

Process-2008, Find cars processing time: 0.170891 seconds

Process-2009, Step 1: Divide with 255, processing time: 0.011025 seconds

Process-2009, Step 2: Resize if scale is not 1: 0.001064 seconds

Process-2009, Step 3: Get HOG channels: 7e-06 seconds

Process-2009, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2009, Step 5: Compute individual channel HOG features for the entire image: 0.142384 seconds

Process-2010, Step 1: Divide with 255, processing time: 0.010559 seconds

Process-2009, Step 6: Misc initializations: 1e-06 seconds

Process-2010, Step 2: Resize if scale is not 1: 0.00085 seconds

Process-2009, Step 7: for loop: 0.007677 seconds

Process-2010, Step 3: Get HOG channels: 8e-06 seconds

Process-2010, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2009All steps time: 0.162164 seconds

Process-2010, Step 5: Compute individual channel HOG features for the entire image: 0.155858 seconds

Process-2010, Step 6: Misc initializations: 1e-06 seconds

Process-2009, Find cars processing time: 0.16227 seconds

Process-2010, Step 7: for loop: 0.009026 seconds

Process-2010All steps time: 0.17631 seconds

Process-2010, Find cars processing time: 0.176387 seconds

Process-2008, Step 1: Divide with 255, processing time: 0.010929 seconds

Process-2008, Step 2: Resize if scale is not 1: 0.00093 seconds

Process-2008, Step 3: Get HOG channels: 1e-05 seconds

Process-2008, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2008, Step 5: Compute individual channel HOG features for the entire image: 0.128776 seconds

Process-2008, Step 6: Misc initializations: 0.0 seconds

Process-2008, Step 7: for loop: 0.006993 seconds

Process-2010, Step 1: Divide with 255, processing time: 0.009592 seconds

Process-2008All steps time: 0.147647 seconds

Process-2010, Step 2: Resize if scale is not 1: 0.000604 seconds

Process-2010, Step 3: Get HOG channels: 5e-06 seconds

Process-2008, Find cars processing time: 0.147728 seconds

Process-2010, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2010, Step 5: Compute individual channel HOG features for the entire image: 0.087833 seconds

Process-2010, Step 6: Misc initializations: 0.0 seconds

Process-2010, Step 7: for loop: 0.005153 seconds

Process-2010All steps time: 0.1031919999999999 seconds

Process-2010, Find cars processing time: 0.103245 seconds

Process-2009, Step 1: Divide with 255, processing time: 0.011001 seconds

Process-2009, Step 2: Resize if scale is not 1: 0.000819 seconds

Process-2009, Step 3: Get HOG channels: 1e-05 seconds

Process-2009, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2009, Step 5: Compute individual channel HOG features for the entire image: 0.129429 seconds

Process-2009, Step 6: Misc initializations: 1e-06 seconds

Process-2009, Step 7: for loop: 0.007226 seconds

Process-2009All steps time: 0.148496 seconds

Process-2009, Find cars processing time: 0.148573 seconds

The times for each task are: [0.742456, 0.688395, 0.608345, 0.607641, 0.397096, 0.370404, 0.4442

Minimum: 0.103245 Maximum: 0.742456 Average: 0.3147 seconds

Number of processes used: 3 window size 260

Length of task list: 21

Number of processes used: 3

Process-2012, Step 1: Divide with 255, processing time: 0.036486 seconds

Process-2012, Step 2: Resize if scale is not 1: 0.010879 seconds

Process-2012, Step 3: Get HOG channels: 8.9e-05 seconds

Process-2012, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-2012, Step 5: Compute individual channel HOG features for the entire image: 0.466614 sec

Process-2012, Step 6: Misc initializations: 1e-06 seconds

Process-2012, Step 7: for loop: 0.038264 seconds

Process-2012All steps time: 0.552345 seconds

Process-2012, Find cars processing time: 0.552585 seconds

Process-2011, Step 1: Divide with 255, processing time: 0.031291 seconds

Process-2011, Step 2: Resize if scale is not 1: 0.013481 seconds

Process-2011, Step 3: Get HOG channels: 0.000115 seconds

Process-2011, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-2011, Step 5: Compute individual channel HOG features for the entire image: 0.582674 sec

Process-2011, Step 6: Misc initializations: 1e-06 seconds

Process-2011, Step 7: for loop: 0.038925 seconds

Process-2011All steps time: 0.666501 seconds

Process-2011, Find cars processing time: 0.666799 seconds

Process-2011, Step 1: Divide with 255, processing time: 0.006572 seconds

Process-2011, Step 2: Resize if scale is not 1: 0.00071 seconds

Process-2011, Step 3: Get HOG channels: 6e-06 seconds

Process-2011, Step 4: Define blocks and steps as above: 2e-05 seconds

Process-2011, Step 5: Compute individual channel HOG features for the entire image: 0.357352 sec

Process-2011, Step 6: Misc initializations: 1e-06 seconds

Process-2011, Step 7: for loop: 0.020945 seconds

Process-2011All steps time: 0.3856059999999995 seconds

Process-2011, Find cars processing time: 0.385668 seconds

Process-2013, Step 1: Divide with 255, processing time: 0.027663 seconds
Process-2013, Step 2: Resize if scale is not 1: 0.011038 seconds
Process-2013, Step 3: Get HOG channels: 7.1e-05 seconds
Process-2013, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2013, Step 5: Compute individual channel HOG features for the entire image: 0.517588 seconds
Process-2013, Step 6: Misc initializations: 1e-06 seconds
Process-2013, Step 7: for loop: 0.035969 seconds

Process-2013All steps time: 0.5923380000000001 seconds

Process-2013, Find cars processing time: 0.592539 seconds

Process-2012, Step 1: Divide with 255, processing time: 0.010095 seconds
Process-2012, Step 2: Resize if scale is not 1: 0.004183 seconds
Process-2012, Step 3: Get HOG channels: 8e-06 seconds
Process-2012, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2012, Step 5: Compute individual channel HOG features for the entire image: 0.436781 seconds
Process-2012, Step 6: Misc initializations: 1e-06 seconds
Process-2012, Step 7: for loop: 0.037584 seconds

Process-2012All steps time: 0.4886599999999999 seconds

Process-2012, Find cars processing time: 0.488732 seconds

Process-2011, Step 1: Divide with 255, processing time: 0.023843 seconds
Process-2011, Step 2: Resize if scale is not 1: 0.000994 seconds
Process-2011, Step 6: Misc initializations: 1e-06 seconds
Process-2011, Step 3: Get HOG channels: 6e-06 seconds
Process-2011, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2011, Step 5: Compute individual channel HOG features for the entire image: 0.361212 seconds
Process-2011, Step 7: for loop: 0.021298 seconds

Process-2011All steps time: 0.4073599999999994 seconds

Process-2011, Find cars processing time: 0.407429 seconds

Process-2013, Step 1: Divide with 255, processing time: 0.011257 seconds
Process-2013, Step 2: Resize if scale is not 1: 0.001188 seconds
Process-2013, Step 3: Get HOG channels: 1e-05 seconds
Process-2013, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2013, Step 5: Compute individual channel HOG features for the entire image: 0.365226 seconds
Process-2013, Step 6: Misc initializations: 0.0 seconds
Process-2013, Step 7: for loop: 0.023994 seconds

Process-2013All steps time: 0.401685 seconds

Process-2013, Find cars processing time: 0.401775 seconds

Process-2012, Step 1: Divide with 255, processing time: 0.015011 seconds
Process-2012, Step 2: Resize if scale is not 1: 0.001109 seconds
Process-2012, Step 3: Get HOG channels: 6e-06 seconds
Process-2012, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2012, Step 5: Compute individual channel HOG features for the entire image: 0.380879 seconds
Process-2012, Step 6: Misc initializations: 1e-06 seconds
Process-2011, Step 1: Divide with 255, processing time: 0.008111 seconds
Process-2012, Step 7: for loop: 0.026767 seconds
Process-2011, Step 2: Resize if scale is not 1: 0.00073 seconds
Process-2011, Step 3: Get HOG channels: 7e-06 seconds

Process-2011, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2012All steps time: 0.4237789999999996 seconds
Process-2011, Step 5: Compute individual channel HOG features for the entire image: 0.236872 seconds

Process-2011, Step 6: Misc initializations: 1e-06 seconds
Process-2012, Find cars processing time: 0.423854 seconds
Process-2011, Step 7: for loop: 0.023784 seconds

Process-2011All steps time: 0.269511 seconds

Process-2011, Find cars processing time: 0.269588 seconds

Process-2011, Step 1: Divide with 255, processing time: 0.009431 seconds
Process-2011, Step 2: Resize if scale is not 1: 0.000823 seconds
Process-2011, Step 3: Get HOG channels: 8e-06 seconds
Process-2011, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2011, Step 5: Compute individual channel HOG features for the entire image: 0.270615 seconds
Process-2011, Step 6: Misc initializations: 2e-06 seconds
Process-2011, Step 7: for loop: 0.017175 seconds

Process-2011All steps time: 0.2980609999999996 seconds

Process-2011, Find cars processing time: 0.298138 seconds

Process-2013, Step 1: Divide with 255, processing time: 0.014062 seconds
Process-2013, Step 2: Resize if scale is not 1: 0.000669 seconds
Process-2013, Step 3: Get HOG channels: 6e-06 seconds
Process-2013, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2013, Step 5: Compute individual channel HOG features for the entire image: 0.238181 seconds
Process-2013, Step 6: Misc initializations: 0.0 seconds
Process-2013, Step 7: for loop: 0.012836 seconds

Process-2013All steps time: 0.26576 seconds

Process-2013, Find cars processing time: 0.265827 seconds

Process-2012, Step 1: Divide with 255, processing time: 0.009377 seconds
Process-2012, Step 2: Resize if scale is not 1: 0.00052 seconds
Process-2012, Step 3: Get HOG channels: 5e-06 seconds
Process-2012, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2012, Step 5: Compute individual channel HOG features for the entire image: 0.248267 seconds
Process-2012, Step 6: Misc initializations: 1e-06 seconds
Process-2012, Step 7: for loop: 0.012676 seconds

Process-2012All steps time: 0.270851 seconds

Process-2012, Find cars processing time: 0.270909 seconds

Process-2011, Step 1: Divide with 255, processing time: 0.021445 seconds
Process-2011, Step 2: Resize if scale is not 1: 0.000604 seconds
Process-2011, Step 3: Get HOG channels: 5e-06 seconds
Process-2011, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2011, Step 5: Compute individual channel HOG features for the entire image: 0.181177 seconds
Process-2011, Step 6: Misc initializations: 1e-06 seconds
Process-2011, Step 7: for loop: 0.009702 seconds

Process-2011All steps time: 0.212939 seconds

Process-2011, Find cars processing time: 0.213002 seconds

Process-2013, Step 1: Divide with 255, processing time: 0.019658 seconds
Process-2013, Step 2: Resize if scale is not 1: 0.000545 seconds
Process-2013, Step 3: Get HOG channels: 6e-06 seconds
Process-2013, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2013, Step 5: Compute individual channel HOG features for the entire image: 0.170487 seconds
Process-2013, Step 6: Misc initializations: 1e-06 seconds
Process-2013, Step 7: for loop: 0.00952 seconds

Process-2013All steps time: 0.2002229999999998 seconds

Process-2013, Find cars processing time: 0.200288 seconds

Process-2011, Step 1: Divide with 255, processing time: 0.00825 seconds
Process-2011, Step 2: Resize if scale is not 1: 0.00099 seconds
Process-2011, Step 3: Get HOG channels: 1e-05 seconds
Process-2011, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2011, Step 5: Compute individual channel HOG features for the entire image: 0.189619 seconds
Process-2011, Step 6: Misc initializations: 0.0 seconds
Process-2011, Step 7: for loop: 0.010561 seconds

Process-2011All steps time: 0.209439 seconds

Process-2011, Find cars processing time: 0.209522 seconds

Process-2012, Step 1: Divide with 255, processing time: 0.011438 seconds
Process-2012, Step 2: Resize if scale is not 1: 0.000556 seconds
Process-2012, Step 3: Get HOG channels: 6e-06 seconds
Process-2012, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2012, Step 5: Compute individual channel HOG features for the entire image: 0.173074 seconds
Process-2012, Step 6: Misc initializations: 1e-06 seconds
Process-2012, Step 7: for loop: 0.009682 seconds

Process-2012All steps time: 0.194763 seconds

Process-2012, Find cars processing time: 0.194826 seconds

Process-2011, Step 1: Divide with 255, processing time: 0.010879 seconds
Process-2011, Step 2: Resize if scale is not 1: 0.00201 seconds
Process-2011, Step 3: Get HOG channels: 9e-06 seconds
Process-2011, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2011, Step 5: Compute individual channel HOG features for the entire image: 0.168407 seconds
Process-2011, Step 6: Misc initializations: 2e-06 seconds
Process-2011, Step 7: for loop: 0.00762 seconds

Process-2011All steps time: 0.188936 seconds

Process-2011, Find cars processing time: 0.189025 seconds

Process-2013, Step 1: Divide with 255, processing time: 0.010186 seconds
Process-2013, Step 2: Resize if scale is not 1: 0.000761 seconds
Process-2013, Step 3: Get HOG channels: 6e-06 seconds
Process-2013, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2013, Step 5: Compute individual channel HOG features for the entire image: 0.12882 seconds
Process-2013, Step 6: Misc initializations: 1e-06 seconds
Process-2013, Step 7: for loop: 0.007333 seconds

Process-2013All steps time: 0.147113 seconds

Process-2013, Find cars processing time: 0.14717 seconds

Process-2012, Step 1: Divide with 255, processing time: 0.00961 seconds
Process-2012, Step 2: Resize if scale is not 1: 0.001096 seconds
Process-2012, Step 3: Get HOG channels: 7e-06 seconds
Process-2012, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2012, Step 5: Compute individual channel HOG features for the entire image: 0.131085 seconds
Process-2012, Step 6: Misc initializations: 1e-06 seconds
Process-2012, Step 7: for loop: 0.007342 seconds

Process-2012All steps time: 0.149148 seconds

Process-2012, Find cars processing time: 0.149214 seconds

Process-2013, Step 1: Divide with 255, processing time: 0.019583 seconds
Process-2013, Step 2: Resize if scale is not 1: 0.000624 seconds
Process-2013, Step 3: Get HOG channels: 6e-06 seconds
Process-2013, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2013, Step 5: Compute individual channel HOG features for the entire image: 0.111865 seconds
Process-2013, Step 6: Misc initializations: 1e-06 seconds
Process-2013, Step 7: for loop: 0.006124 seconds

Process-2013All steps time: 0.13821 seconds

Process-2013, Find cars processing time: 0.138285 seconds

Process-2011, Step 1: Divide with 255, processing time: 0.010937 seconds
Process-2011, Step 2: Resize if scale is not 1: 0.000824 seconds
Process-2011, Step 3: Get HOG channels: 9e-06 seconds
Process-2011, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2011, Step 5: Compute individual channel HOG features for the entire image: 0.130532 seconds
Process-2011, Step 6: Misc initializations: 1e-06 seconds
Process-2011, Step 7: for loop: 0.007244 seconds

Process-2011All steps time: 0.14955500000000002 seconds

Process-2011, Find cars processing time: 0.149637 seconds

The times for each task are: [0.552585, 0.666799, 0.385668, 0.592539, 0.488732, 0.407429, 0.401771]

Minimum: 0.138285 Maximum: 0.666799 Average: 0.315 seconds

Number of processes used: 3 window size 260
Length of task list: 21
Number of processes used: 3

Process-2014, Step 1: Divide with 255, processing time: 0.033565 seconds
Process-2014, Step 2: Resize if scale is not 1: 0.01242 seconds
Process-2014, Step 3: Get HOG channels: 9.2e-05 seconds
Process-2014, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2014, Step 5: Compute individual channel HOG features for the entire image: 0.62334 seconds
Process-2014, Step 6: Misc initializations: 1e-06 seconds
Process-2014, Step 7: for loop: 0.060816 seconds

Process-2014All steps time: 0.730244 seconds

Process-2014, Find cars processing time: 0.730563 seconds

Process-2015, Step 1: Divide with 255, processing time: 0.038077 seconds
Process-2015, Step 2: Resize if scale is not 1: 0.014231 seconds

Process-2015, Step 3: Get HOG channels: 5.7e-05 seconds
Process-2015, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2015, Step 5: Compute individual channel HOG features for the entire image: 0.504549 seconds
Process-2015, Step 6: Misc initializations: 1e-06 seconds
Process-2015, Step 7: for loop: 0.035843 seconds

Process-2015All steps time: 0.592768 seconds

Process-2015, Find cars processing time: 0.592984 seconds

Process-2016, Step 1: Divide with 255, processing time: 0.026889 seconds
Process-2016, Step 2: Resize if scale is not 1: 0.012431 seconds
Process-2016, Step 3: Get HOG channels: 0.000101 seconds
Process-2016, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2016, Step 5: Compute individual channel HOG features for the entire image: 0.521614 seconds
Process-2016, Step 6: Misc initializations: 1e-06 seconds
Process-2016, Step 7: for loop: 0.037958 seconds

Process-2016All steps time: 0.5990040000000001 seconds

Process-2016, Find cars processing time: 0.599258 seconds

Process-2015, Step 1: Divide with 255, processing time: 0.009065 seconds
Process-2015, Step 2: Resize if scale is not 1: 0.000978 seconds
Process-2015, Step 3: Get HOG channels: 7e-06 seconds
Process-2015, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2015, Step 5: Compute individual channel HOG features for the entire image: 0.280011 seconds
Process-2015, Step 6: Misc initializations: 1e-06 seconds
Process-2015, Step 7: for loop: 0.019482 seconds

Process-2015All steps time: 0.30955099999999997 seconds

Process-2015, Find cars processing time: 0.309617 seconds

Process-2014, Step 1: Divide with 255, processing time: 0.010811 seconds
Process-2014, Step 2: Resize if scale is not 1: 0.011165 seconds
Process-2014, Step 3: Get HOG channels: 1.2e-05 seconds
Process-2014, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-2014, Step 5: Compute individual channel HOG features for the entire image: 0.520156 seconds
Process-2014, Step 6: Misc initializations: 1e-06 seconds
Process-2014, Step 7: for loop: 0.038384 seconds

Process-2014All steps time: 0.58054 seconds

Process-2014, Find cars processing time: 0.58064 seconds

Process-2015, Step 1: Divide with 255, processing time: 0.01195 seconds
Process-2015, Step 2: Resize if scale is not 1: 0.000998 seconds

Process-2015, Step 3: Get HOG channels: 5e-06 seconds
Process-2015, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2015, Step 5: Compute individual channel HOG features for the entire image: 0.281255 seconds
Process-2015, Step 6: Misc initializations: 0.0 seconds
Process-2015, Step 7: for loop: 0.019734 seconds

Process-2016, Step 1: Divide with 255, processing time: 0.01174 seconds
Process-2015All steps time: 0.313947 seconds
Process-2016, Step 2: Resize if scale is not 1: 0.00107 seconds

Process-2016, Step 3: Get HOG channels: 7e-06 seconds
Process-2016, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2015, Find cars processing time: 0.314027 seconds
Process-2016, Step 5: Compute individual channel HOG features for the entire image: 0.282079 seconds

Process-2016, Step 6: Misc initializations: 0.0 seconds
Process-2016, Step 7: for loop: 0.027494 seconds

Process-2016All steps time: 0.32239700000000004 seconds

Process-2016, Find cars processing time: 0.322457 seconds

Process-2014, Step 1: Divide with 255, processing time: 0.010565 seconds
Process-2014, Step 2: Resize if scale is not 1: 0.001143 seconds
Process-2014, Step 3: Get HOG channels: 7e-06 seconds
Process-2014, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2014, Step 5: Compute individual channel HOG features for the entire image: 0.405517 seconds
Process-2014, Step 6: Misc initializations: 2e-06 seconds
Process-2014, Step 7: for loop: 0.030238 seconds

Process-2014All steps time: 0.447479 seconds

Process-2014, Find cars processing time: 0.447574 seconds

Process-2015, Step 1: Divide with 255, processing time: 0.008648 seconds
Process-2015, Step 2: Resize if scale is not 1: 0.000602 seconds
Process-2015, Step 3: Get HOG channels: 6e-06 seconds
Process-2015, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2015, Step 5: Compute individual channel HOG features for the entire image: 0.257127 seconds
Process-2015, Step 6: Misc initializations: 1e-06 seconds
Process-2015, Step 7: for loop: 0.016079 seconds

Process-2015All steps time: 0.2824689999999997 seconds

Process-2015, Find cars processing time: 0.282534 seconds

Process-2016, Step 1: Divide with 255, processing time: 0.010314 seconds
Process-2016, Step 2: Resize if scale is not 1: 0.000821 seconds

Process-2016, Step 3: Get HOG channels: 5e-06 seconds
Process-2016, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2016, Step 5: Compute individual channel HOG features for the entire image: 0.210067 seconds
Process-2016, Step 6: Misc initializations: 1e-06 seconds
Process-2016, Step 7: for loop: 0.012419 seconds

Process-2016All steps time: 0.233632 seconds

Process-2016, Find cars processing time: 0.233694 seconds

Process-2014, Step 1: Divide with 255, processing time: 0.010177 seconds
Process-2014, Step 2: Resize if scale is not 1: 0.000749 seconds
Process-2014, Step 3: Get HOG channels: 7e-06 seconds
Process-2014, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2014, Step 5: Compute individual channel HOG features for the entire image: 0.25389 seconds
Process-2014, Step 6: Misc initializations: 1e-06 seconds
Process-2014, Step 7: for loop: 0.01629 seconds

Process-2014All steps time: 0.2811199999999999 seconds

Process-2014, Find cars processing time: 0.281194 seconds

Process-2015, Step 1: Divide with 255, processing time: 0.010626 seconds
Process-2015, Step 2: Resize if scale is not 1: 0.00071 seconds
Process-2015, Step 3: Get HOG channels: 6e-06 seconds
Process-2015, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2015, Step 5: Compute individual channel HOG features for the entire image: 0.218705 seconds
Process-2015, Step 6: Misc initializations: 0.0 seconds
Process-2015, Step 7: for loop: 0.012052 seconds

Process-2015All steps time: 0.2421060000000002 seconds

Process-2015, Find cars processing time: 0.242175 seconds

Process-2014, Step 1: Divide with 255, processing time: 0.00922 seconds
Process-2014, Step 2: Resize if scale is not 1: 0.000721 seconds
Process-2014, Step 3: Get HOG channels: 8e-06 seconds
Process-2014, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2014, Step 5: Compute individual channel HOG features for the entire image: 0.177298 seconds
Process-2014, Step 6: Misc initializations: 1e-06 seconds
Process-2014, Step 7: for loop: 0.009819 seconds

Process-2014All steps time: 0.197075 seconds

Process-2014, Find cars processing time: 0.197146 seconds

Process-2016, Step 1: Divide with 255, processing time: 0.01031 seconds
Process-2016, Step 2: Resize if scale is not 1: 0.00092 seconds

Process-2016, Step 3: Get HOG channels: 8e-06 seconds
Process-2016, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2016, Step 5: Compute individual channel HOG features for the entire image: 0.14074 seconds
Process-2016, Step 6: Misc initializations: 0.0 seconds
Process-2016, Step 7: for loop: 0.007465 seconds

Process-2016All steps time: 0.159451 seconds

Process-2016, Find cars processing time: 0.159522 seconds

Process-2016, Step 1: Divide with 255, processing time: 0.010258 seconds
Process-2016, Step 2: Resize if scale is not 1: 0.001072 seconds
Process-2016, Step 3: Get HOG channels: 6e-06 seconds
Process-2016, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2016, Step 5: Compute individual channel HOG features for the entire image: 0.136877 seconds
Process-2016, Step 6: Misc initializations: 1e-06 seconds
Process-2016, Step 7: for loop: 0.007548 seconds

Process-2016All steps time: 0.155768 seconds

Process-2016, Find cars processing time: 0.155836 seconds

Process-2015, Step 1: Divide with 255, processing time: 0.006881 seconds
Process-2015, Step 2: Resize if scale is not 1: 0.000692 seconds
Process-2015, Step 3: Get HOG channels: 7e-06 seconds
Process-2015, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2015, Step 5: Compute individual channel HOG features for the entire image: 0.145455 seconds
Process-2015, Step 6: Misc initializations: 0.0 seconds
Process-2015, Step 7: for loop: 0.00815 seconds

Process-2015All steps time: 0.161191 seconds

Process-2015, Find cars processing time: 0.161258 seconds

Process-2016, Step 1: Divide with 255, processing time: 0.009934 seconds
Process-2016, Step 2: Resize if scale is not 1: 0.000743 seconds
Process-2016, Step 3: Get HOG channels: 7e-06 seconds
Process-2016, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2016, Step 5: Compute individual channel HOG features for the entire image: 0.125462 seconds
Process-2016, Step 6: Misc initializations: 0.0 seconds
Process-2016, Step 7: for loop: 0.007242 seconds

Process-2016All steps time: 0.143395 seconds

Process-2016, Find cars processing time: 0.143456 seconds

Process-2014, Step 1: Divide with 255, processing time: 0.010925 seconds
Process-2014, Step 2: Resize if scale is not 1: 0.000885 seconds

Process-2014, Step 3: Get HOG channels: 9e-06 seconds
Process-2014, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2014, Step 5: Compute individual channel HOG features for the entire image: 0.193211 seconds
Process-2014, Step 6: Misc initializations: 1e-06 seconds
Process-2014, Step 7: for loop: 0.00991 seconds

Process-2014All steps time: 0.21495 seconds

Process-2014, Find cars processing time: 0.215035 seconds

Process-2015, Step 1: Divide with 255, processing time: 0.009654 seconds
Process-2015, Step 2: Resize if scale is not 1: 0.000968 seconds
Process-2015, Step 3: Get HOG channels: 6e-06 seconds
Process-2015, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2015, Step 5: Compute individual channel HOG features for the entire image: 0.10828 seconds
Process-2015, Step 6: Misc initializations: 1e-06 seconds
Process-2015, Step 7: for loop: 0.005928 seconds

Process-2015All steps time: 0.12484300000000001 seconds

Process-2015, Find cars processing time: 0.124912 seconds

Process-2016, Step 1: Divide with 255, processing time: 0.009939 seconds
Process-2016, Step 2: Resize if scale is not 1: 0.000777 seconds
Process-2016, Step 3: Get HOG channels: 8e-06 seconds
Process-2016, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2016, Step 5: Compute individual channel HOG features for the entire image: 0.145573 seconds
Process-2016, Step 6: Misc initializations: 0.0 seconds
Process-2016, Step 7: for loop: 0.006937 seconds

Process-2016All steps time: 0.163242 seconds

Process-2016, Find cars processing time: 0.163309 seconds

Process-2014, Step 1: Divide with 255, processing time: 0.011055 seconds
Process-2014, Step 2: Resize if scale is not 1: 0.001055 seconds
Process-2014, Step 3: Get HOG channels: 9e-06 seconds
Process-2014, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2014, Step 5: Compute individual channel HOG features for the entire image: 0.132925 seconds
Process-2014, Step 6: Misc initializations: 1e-06 seconds
Process-2014, Step 7: for loop: 0.0065 seconds

Process-2014All steps time: 0.151554 seconds

Process-2014, Find cars processing time: 0.151646 seconds

The times for each task are: [0.730563, 0.592984, 0.599258, 0.309617, 0.58064, 0.314027, 0.32245]

Minimum: 0.124912 Maximum: 0.730563 Average: 0.3052 seconds

Number of processes used: 3 window size 260
Length of task list: 21
Number of processes used: 3

Process-2017, Step 1: Divide with 255, processing time: 0.031196 seconds
Process-2017, Step 2: Resize if scale is not 1: 0.01229 seconds
Process-2017, Step 3: Get HOG channels: 0.000205 seconds
Process-2017, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2017, Step 5: Compute individual channel HOG features for the entire image: 0.565122 seconds
Process-2017, Step 6: Misc initializations: 1e-06 seconds
Process-2017, Step 7: for loop: 0.036225 seconds

Process-2017All steps time: 0.645049 seconds

Process-2017, Find cars processing time: 0.645356 seconds

Process-2018, Step 1: Divide with 255, processing time: 0.024623 seconds
Process-2018, Step 2: Resize if scale is not 1: 0.013458 seconds
Process-2018, Step 3: Get HOG channels: 8.3e-05 seconds
Process-2018, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-2018, Step 5: Compute individual channel HOG features for the entire image: 0.56062 seconds
Process-2018, Step 6: Misc initializations: 1e-06 seconds
Process-2018, Step 7: for loop: 0.050939 seconds

Process-2018All steps time: 0.649737 seconds

Process-2018, Find cars processing time: 0.649953 seconds

Process-2019, Step 1: Divide with 255, processing time: 0.025315 seconds
Process-2019, Step 2: Resize if scale is not 1: 0.011024 seconds
Process-2019, Step 3: Get HOG channels: 0.000123 seconds
Process-2019, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-2019, Step 5: Compute individual channel HOG features for the entire image: 0.50719 seconds
Process-2019, Step 6: Misc initializations: 1e-06 seconds
Process-2019, Step 7: for loop: 0.037071 seconds

Process-2019All steps time: 0.5807370000000001 seconds

Process-2019, Find cars processing time: 0.580974 seconds

Process-2018, Step 1: Divide with 255, processing time: 0.009468 seconds
Process-2018, Step 2: Resize if scale is not 1: 0.000888 seconds
Process-2018, Step 3: Get HOG channels: 6e-06 seconds
Process-2018, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2018, Step 5: Compute individual channel HOG features for the entire image: 0.365264 seconds

Process-2018, Step 6: Misc initializations: 1e-06 seconds
Process-2018, Step 7: for loop: 0.019638 seconds

Process-2018All steps time: 0.39527199999999996 seconds

Process-2018, Find cars processing time: 0.395345 seconds

Process-2017, Step 1: Divide with 255, processing time: 0.010963 seconds
Process-2017, Step 2: Resize if scale is not 1: 0.003868 seconds
Process-2017, Step 3: Get HOG channels: 9e-06 seconds
Process-2017, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2017, Step 5: Compute individual channel HOG features for the entire image: 0.505697 seconds
Process-2017, Step 6: Misc initializations: 1e-06 seconds
Process-2017, Step 7: for loop: 0.039961 seconds

Process-2017All steps time: 0.560508 seconds

Process-2017, Find cars processing time: 0.560593 seconds

Process-2019, Step 1: Divide with 255, processing time: 0.010344 seconds
Process-2019, Step 2: Resize if scale is not 1: 0.001146 seconds
Process-2019, Step 3: Get HOG channels: 8e-06 seconds
Process-2019, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2019, Step 5: Compute individual channel HOG features for the entire image: 0.334638 seconds
Process-2019, Step 6: Misc initializations: 1e-06 seconds
Process-2019, Step 7: for loop: 0.018693 seconds

Process-2019All steps time: 0.364838 seconds

Process-2019, Find cars processing time: 0.364911 seconds

Process-2019, Step 1: Divide with 255, processing time: 0.00915 seconds
Process-2019, Step 2: Resize if scale is not 1: 0.000903 seconds
Process-2019, Step 3: Get HOG channels: 8e-06 seconds
Process-2019, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2019, Step 5: Compute individual channel HOG features for the entire image: 0.218244 seconds
Process-2019, Step 6: Misc initializations: 1e-06 seconds
Process-2019, Step 7: for loop: 0.013836 seconds

Process-2019All steps time: 0.2421489999999998 seconds

Process-2019, Find cars processing time: 0.242224 seconds

Process-2018, Step 1: Divide with 255, processing time: 0.009762 seconds
Process-2018, Step 2: Resize if scale is not 1: 0.000705 seconds
Process-2018, Step 3: Get HOG channels: 6e-06 seconds
Process-2018, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2018, Step 5: Compute individual channel HOG features for the entire image: 0.301281 seconds

Process-2018, Step 6: Misc initializations: 1e-06 seconds
Process-2018, Step 7: for loop: 0.02075 seconds

Process-2018All steps time: 0.332511 seconds

Process-2018, Find cars processing time: 0.332589 seconds

Process-2017, Step 1: Divide with 255, processing time: 0.00963 seconds
Process-2017, Step 2: Resize if scale is not 1: 0.001413 seconds
Process-2017, Step 3: Get HOG channels: 8e-06 seconds
Process-2017, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2017, Step 5: Compute individual channel HOG features for the entire image: 0.417214 seconds
Process-2017, Step 6: Misc initializations: 1e-06 seconds
Process-2017, Step 7: for loop: 0.0205 seconds

Process-2017All steps time: 0.448773 seconds

Process-2017, Find cars processing time: 0.44886 seconds

Process-2018, Step 1: Divide with 255, processing time: 0.009022 seconds
Process-2018, Step 2: Resize if scale is not 1: 0.001114 seconds
Process-2018, Step 3: Get HOG channels: 1.3e-05 seconds
Process-2018, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2018, Step 5: Compute individual channel HOG features for the entire image: 0.206778 seconds
Process-2018, Step 6: Misc initializations: 0.0 seconds
Process-2018, Step 7: for loop: 0.012896 seconds

Process-2018All steps time: 0.2298329999999998 seconds

Process-2018, Find cars processing time: 0.229926 seconds

Process-2017, Step 1: Divide with 255, processing time: 0.009509 seconds
Process-2017, Step 2: Resize if scale is not 1: 0.000798 seconds
Process-2017, Step 3: Get HOG channels: 8e-06 seconds
Process-2017, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2017, Step 5: Compute individual channel HOG features for the entire image: 0.24721 seconds
Process-2017, Step 6: Misc initializations: 0.0 seconds
Process-2017, Step 7: for loop: 0.01368 seconds

Process-2017All steps time: 0.2712130000000004 seconds

Process-2017, Find cars processing time: 0.271276 seconds

Process-2019, Step 1: Divide with 255, processing time: 0.010539 seconds
Process-2019, Step 2: Resize if scale is not 1: 0.000849 seconds
Process-2019, Step 3: Get HOG channels: 8e-06 seconds
Process-2019, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2019, Step 5: Compute individual channel HOG features for the entire image: 0.186145 seconds

Process-2019, Step 6: Misc initializations: 1e-06 seconds
Process-2019, Step 7: for loop: 0.013004 seconds

Process-2019All steps time: 0.210554 seconds

Process-2019, Find cars processing time: 0.210653 seconds

Process-2017, Step 1: Divide with 255, processing time: 0.008317 seconds
Process-2017, Step 2: Resize if scale is not 1: 0.000642 seconds
Process-2017, Step 3: Get HOG channels: 7e-06 seconds
Process-2017, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2017, Step 5: Compute individual channel HOG features for the entire image: 0.183288 seconds
Process-2017, Step 6: Misc initializations: 1e-06 seconds
Process-2017, Step 7: for loop: 0.009633 seconds

Process-2017All steps time: 0.20189400000000002 seconds

Process-2017, Find cars processing time: 0.20196 seconds

Process-2018, Step 1: Divide with 255, processing time: 0.006041 seconds
Process-2018, Step 2: Resize if scale is not 1: 0.000627 seconds
Process-2018, Step 3: Get HOG channels: 6e-06 seconds
Process-2018, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2018, Step 5: Compute individual channel HOG features for the entire image: 0.140128 seconds
Process-2018, Step 6: Misc initializations: 1e-06 seconds
Process-2018, Step 7: for loop: 0.00819 seconds

Process-2018All steps time: 0.155 seconds

Process-2018, Find cars processing time: 0.155055 seconds

Process-2019, Step 1: Divide with 255, processing time: 0.007284 seconds
Process-2019, Step 2: Resize if scale is not 1: 0.000672 seconds
Process-2019, Step 3: Get HOG channels: 6e-06 seconds
Process-2019, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2019, Step 5: Compute individual channel HOG features for the entire image: 0.137364 seconds
Process-2019, Step 6: Misc initializations: 1e-06 seconds
Process-2019, Step 7: for loop: 0.008088 seconds

Process-2019All steps time: 0.15342200000000003 seconds

Process-2019, Find cars processing time: 0.15348 seconds

Process-2017, Step 1: Divide with 255, processing time: 0.010743 seconds
Process-2017, Step 2: Resize if scale is not 1: 0.00085 seconds
Process-2017, Step 3: Get HOG channels: 7e-06 seconds
Process-2017, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2017, Step 5: Compute individual channel HOG features for the entire image: 0.186438 seconds

Process-2017, Step 6: Misc initializations: 1e-06 seconds
Process-2017, Step 7: for loop: 0.009988 seconds

Process-2017All steps time: 0.208035 seconds

Process-2017, Find cars processing time: 0.20811 seconds

Process-2017, Step 1: Divide with 255, processing time: 0.011039 seconds
Process-2017, Step 2: Resize if scale is not 1: 0.001016 seconds
Process-2017, Step 3: Get HOG channels: 6e-06 seconds
Process-2017, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2017, Step 5: Compute individual channel HOG features for the entire image: 0.127035 seconds
Process-2017, Step 6: Misc initializations: 0.0 seconds
Process-2017, Step 7: for loop: 0.006366 seconds

Process-2017All steps time: 0.14546900000000001 seconds

Process-2017, Find cars processing time: 0.145541 seconds

Process-2018, Step 1: Divide with 255, processing time: 0.00857 seconds
Process-2018, Step 2: Resize if scale is not 1: 0.001067 seconds
Process-2018, Step 3: Get HOG channels: 7e-06 seconds
Process-2018, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2018, Step 5: Compute individual channel HOG features for the entire image: 0.160184 seconds
Process-2018, Step 6: Misc initializations: 1e-06 seconds
Process-2018, Step 7: for loop: 0.007321 seconds

Process-2018All steps time: 0.1771569999999998 seconds

Process-2018, Find cars processing time: 0.177234 seconds

Process-2019, Step 1: Divide with 255, processing time: 0.010885 seconds
Process-2019, Step 2: Resize if scale is not 1: 0.001242 seconds
Process-2019, Step 3: Get HOG channels: 8e-06 seconds
Process-2019, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2019, Step 5: Compute individual channel HOG features for the entire image: 0.126621 seconds
Process-2019, Step 6: Misc initializations: 1e-06 seconds
Process-2019, Step 7: for loop: 0.007307 seconds

Process-2019All steps time: 0.1460710000000003 seconds

Process-2019, Find cars processing time: 0.146139 seconds

Process-2017, Step 1: Divide with 255, processing time: 0.010847 seconds
Process-2017, Step 2: Resize if scale is not 1: 0.001101 seconds
Process-2017, Step 3: Get HOG channels: 1.3e-05 seconds
Process-2017, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-2017, Step 5: Compute individual channel HOG features for the entire image: 0.126316 seconds

Process-2017, Step 6: Misc initializations: 0.0 seconds
Process-2017, Step 7: for loop: 0.007146 seconds

Process-2017All steps time: 0.145436 seconds

Process-2017, Find cars processing time: 0.145535 seconds

Process-2018, Step 1: Divide with 255, processing time: 0.010586 seconds
Process-2018, Step 2: Resize if scale is not 1: 0.00062 seconds
Process-2018, Step 3: Get HOG channels: 6e-06 seconds
Process-2018, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2018, Step 5: Compute individual channel HOG features for the entire image: 0.112729 seconds
Process-2018, Step 6: Misc initializations: 1e-06 seconds
Process-2018, Step 7: for loop: 0.006039 seconds

Process-2018All steps time: 0.12999 seconds

Process-2018, Find cars processing time: 0.130063 seconds

The times for each task are: [0.645356, 0.649953, 0.580974, 0.395345, 0.560593, 0.364911, 0.242221]

Minimum: 0.130063 Maximum: 0.649953 Average: 0.3046 seconds

Number of processes used: 3 window size 260
Length of task list: 21
Number of processes used: 3

Process-2020, Step 1: Divide with 255, processing time: 0.027701 seconds
Process-2020, Step 2: Resize if scale is not 1: 0.010546 seconds
Process-2020, Step 3: Get HOG channels: 8.6e-05 seconds
Process-2020, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-2020, Step 5: Compute individual channel HOG features for the entire image: 0.533191 seconds
Process-2020, Step 6: Misc initializations: 1e-06 seconds
Process-2020, Step 7: for loop: 0.039659 seconds

Process-2020All steps time: 0.611196 seconds

Process-2020, Find cars processing time: 0.611419 seconds

Process-2021, Step 1: Divide with 255, processing time: 0.029236 seconds
Process-2021, Step 2: Resize if scale is not 1: 0.012412 seconds
Process-2021, Step 3: Get HOG channels: 6e-05 seconds
Process-2021, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-2021, Step 5: Compute individual channel HOG features for the entire image: 0.54298 seconds
Process-2021, Step 6: Misc initializations: 1e-06 seconds
Process-2021, Step 7: for loop: 0.036866 seconds

Process-2021All steps time: 0.621568 seconds

Process-2021, Find cars processing time: 0.62181 seconds

Process-2022, Step 1: Divide with 255, processing time: 0.025174 seconds

Process-2022, Step 2: Resize if scale is not 1: 0.010267 seconds

Process-2022, Step 3: Get HOG channels: 7.3e-05 seconds

Process-2022, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2022, Step 5: Compute individual channel HOG features for the entire image: 0.457966 seconds

Process-2022, Step 6: Misc initializations: 1e-06 seconds

Process-2022, Step 7: for loop: 0.036747 seconds

Process-2022All steps time: 0.530235 seconds

Process-2022, Find cars processing time: 0.530443 seconds

Process-2020, Step 1: Divide with 255, processing time: 0.009016 seconds

Process-2020, Step 2: Resize if scale is not 1: 0.005085 seconds

Process-2020, Step 3: Get HOG channels: 8e-06 seconds

Process-2020, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2020, Step 5: Compute individual channel HOG features for the entire image: 0.430237 seconds

Process-2020, Step 6: Misc initializations: 1e-06 seconds

Process-2020, Step 7: for loop: 0.029721 seconds

Process-2020All steps time: 0.4740759999999994 seconds

Process-2020, Find cars processing time: 0.474147 seconds

Process-2021, Step 1: Divide with 255, processing time: 0.010256 seconds

Process-2021, Step 2: Resize if scale is not 1: 0.001186 seconds

Process-2021, Step 3: Get HOG channels: 8e-06 seconds

Process-2021, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2021, Step 5: Compute individual channel HOG features for the entire image: 0.289844 seconds

Process-2021, Step 6: Misc initializations: 1e-06 seconds

Process-2021, Step 7: for loop: 0.019348 seconds

Process-2021All steps time: 0.3206519999999994 seconds

Process-2021, Find cars processing time: 0.32073 seconds

Process-2022, Step 1: Divide with 255, processing time: 0.008976 seconds

Process-2022, Step 2: Resize if scale is not 1: 0.001292 seconds

Process-2022, Step 3: Get HOG channels: 1.1e-05 seconds

Process-2022, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2022, Step 5: Compute individual channel HOG features for the entire image: 0.35418 seconds

Process-2022, Step 6: Misc initializations: 2e-06 seconds

Process-2022, Step 7: for loop: 0.019755 seconds

Process-2022All steps time: 0.3842250000000004 seconds

Process-2022, Find cars processing time: 0.384307 seconds

Process-2020, Step 1: Divide with 255, processing time: 0.010608 seconds

Process-2020, Step 2: Resize if scale is not 1: 0.001348 seconds

Process-2020, Step 3: Get HOG channels: 5e-06 seconds

Process-2020, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2020, Step 5: Compute individual channel HOG features for the entire image: 0.317005 seconds

Process-2020, Step 6: Misc initializations: 1e-06 seconds

Process-2020, Step 7: for loop: 0.024428 seconds

Process-2020All steps time: 0.3534009999999997 seconds

Process-2020, Find cars processing time: 0.353491 seconds

Process-2022, Step 1: Divide with 255, processing time: 0.010664 seconds

Process-2021, Step 1: Divide with 255, processing time: 0.010245 seconds

Process-2021, Step 2: Resize if scale is not 1: 0.000623 seconds

Process-2022, Step 2: Resize if scale is not 1: 0.000863 seconds

Process-2021, Step 3: Get HOG channels: 5e-06 seconds

Process-2022, Step 3: Get HOG channels: 9e-06 seconds

Process-2022, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2021, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2022, Step 5: Compute individual channel HOG features for the entire image: 0.302719 seconds

Process-2021, Step 5: Compute individual channel HOG features for the entire image: 0.301018 seconds

Process-2022, Step 6: Misc initializations: 1e-06 seconds

Process-2021, Step 6: Misc initializations: 1e-06 seconds

Process-2022, Step 7: for loop: 0.015142 seconds

Process-2021, Step 7: for loop: 0.021173 seconds

Process-2022All steps time: 0.329405 seconds

Process-2021All steps time: 0.33307 seconds

Process-2022, Find cars processing time: 0.329476 seconds

Process-2021, Find cars processing time: 0.333127 seconds

Process-2022, Step 1: Divide with 255, processing time: 0.010536 seconds

Process-2022, Step 2: Resize if scale is not 1: 0.000835 seconds

Process-2022, Step 3: Get HOG channels: 7e-06 seconds

Process-2022, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2022, Step 5: Compute individual channel HOG features for the entire image: 0.256737 seconds

Process-2022, Step 6: Misc initializations: 1e-06 seconds

Process-2022, Step 7: for loop: 0.015631 seconds

Process-2022All steps time: 0.283755 seconds

Process-2022, Find cars processing time: 0.283836 seconds

Process-2020, Step 1: Divide with 255, processing time: 0.011139 seconds

Process-2020, Step 2: Resize if scale is not 1: 0.001016 seconds

Process-2020, Step 3: Get HOG channels: 8e-06 seconds

Process-2020, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2020, Step 5: Compute individual channel HOG features for the entire image: 0.253466 seconds

Process-2020, Step 6: Misc initializations: 1e-06 seconds

Process-2020, Step 7: for loop: 0.015797 seconds

Process-2020All steps time: 0.281436 seconds

Process-2020, Find cars processing time: 0.28151 seconds

Process-2021, Step 1: Divide with 255, processing time: 0.006711 seconds

Process-2021, Step 2: Resize if scale is not 1: 0.000723 seconds

Process-2021, Step 3: Get HOG channels: 7e-06 seconds

Process-2021, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2021, Step 5: Compute individual channel HOG features for the entire image: 0.280118 seconds

Process-2021, Step 6: Misc initializations: 1e-06 seconds

Process-2021, Step 7: for loop: 0.015991 seconds

Process-2021All steps time: 0.3035579999999994 seconds

Process-2021, Find cars processing time: 0.303632 seconds

Process-2022, Step 1: Divide with 255, processing time: 0.010167 seconds

Process-2022, Step 2: Resize if scale is not 1: 0.000845 seconds

Process-2022, Step 3: Get HOG channels: 7e-06 seconds

Process-2022, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2022, Step 5: Compute individual channel HOG features for the entire image: 0.170349 seconds

Process-2022, Step 6: Misc initializations: 1e-06 seconds

Process-2022, Step 7: for loop: 0.009788 seconds

Process-2022All steps time: 0.191164 seconds

Process-2022, Find cars processing time: 0.191238 seconds

Process-2020, Step 1: Divide with 255, processing time: 0.010275 seconds

Process-2020, Step 2: Resize if scale is not 1: 0.000526 seconds

Process-2020, Step 3: Get HOG channels: 6e-06 seconds

Process-2020, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2020, Step 5: Compute individual channel HOG features for the entire image: 0.192365 seconds

Process-2020, Step 6: Misc initializations: 1e-06 seconds

Process-2020, Step 7: for loop: 0.009727 seconds

Process-2020All steps time: 0.212905 seconds

Process-2020, Find cars processing time: 0.212969 seconds

Process-2020, Step 1: Divide with 255, processing time: 0.009624 seconds

Process-2022, Step 1: Divide with 255, processing time: 0.009496 seconds

Process-2020, Step 2: Resize if scale is not 1: 0.001191 seconds

Process-2022, Step 2: Resize if scale is not 1: 0.000513 seconds

Process-2020, Step 3: Get HOG channels: 8e-06 seconds

Process-2022, Step 3: Get HOG channels: 5e-06 seconds

Process-2020, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2022, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2020, Step 5: Compute individual channel HOG features for the entire image: 0.106758 seconds

Process-2022, Step 5: Compute individual channel HOG features for the entire image: 0.169095 seconds

Process-2020, Step 6: Misc initializations: 0.0 seconds

Process-2022, Step 6: Misc initializations: 1e-06 seconds

Process-2020, Step 7: for loop: 0.005924 seconds

Process-2022, Step 7: for loop: 0.009475 seconds

Process-2020All steps time: 0.123512 seconds

Process-2022All steps time: 0.188591 seconds

Process-2020, Find cars processing time: 0.123572 seconds

Process-2022, Find cars processing time: 0.188651 seconds

Process-2021, Step 1: Divide with 255, processing time: 0.01084 seconds

Process-2021, Step 2: Resize if scale is not 1: 0.001007 seconds

Process-2021, Step 3: Get HOG channels: 9e-06 seconds

Process-2021, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2021, Step 5: Compute individual channel HOG features for the entire image: 0.181312 seconds

Process-2021, Step 6: Misc initializations: 1e-06 seconds

Process-2021, Step 7: for loop: 0.009525 seconds

Process-2021All steps time: 0.202703 seconds

Process-2021, Find cars processing time: 0.202778 seconds

Process-2022, Step 1: Divide with 255, processing time: 0.010444 seconds

Process-2022, Step 2: Resize if scale is not 1: 0.001271 seconds

Process-2022, Step 3: Get HOG channels: 9e-06 seconds

Process-2022, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2022, Step 5: Compute individual channel HOG features for the entire image: 0.129055 seconds

Process-2022, Step 6: Misc initializations: 0.0 seconds

Process-2022, Step 7: for loop: 0.023713 seconds

Process-2022All steps time: 0.164501 seconds

Process-2022, Find cars processing time: 0.164591 seconds

Process-2021, Step 1: Divide with 255, processing time: 0.00977 seconds

Process-2021, Step 2: Resize if scale is not 1: 0.001057 seconds

Process-2021, Step 3: Get HOG channels: 7e-06 seconds

Process-2021, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2021, Step 5: Compute individual channel HOG features for the entire image: 0.152431 seconds

Process-2021, Step 6: Misc initializations: 1e-06 seconds

Process-2021, Step 7: for loop: 0.007216 seconds

Process-2021All steps time: 0.170488 seconds

Process-2021, Find cars processing time: 0.170552 seconds

Process-2020, Step 1: Divide with 255, processing time: 0.010271 seconds

Process-2020, Step 2: Resize if scale is not 1: 0.000768 seconds

Process-2020, Step 3: Get HOG channels: 8e-06 seconds

Process-2020, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2020, Step 5: Compute individual channel HOG features for the entire image: 0.098868 seconds

Process-2020, Step 6: Misc initializations: 0.0 seconds

Process-2020, Step 7: for loop: 0.005714 seconds

Process-2020All steps time: 0.1156369999999999 seconds

Process-2020, Find cars processing time: 0.115715 seconds

Process-2022, Step 1: Divide with 255, processing time: 0.010233 seconds

Process-2022, Step 2: Resize if scale is not 1: 0.00089 seconds

Process-2022, Step 3: Get HOG channels: 8e-06 seconds

Process-2022, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2022, Step 5: Compute individual channel HOG features for the entire image: 0.084615 seconds

Process-2022, Step 6: Misc initializations: 1e-06 seconds

Process-2022, Step 7: for loop: 0.004611 seconds

Process-2022All steps time: 0.100367 seconds

Process-2022, Find cars processing time: 0.100426 seconds

The times for each task are: [0.611419, 0.62181, 0.530443, 0.474147, 0.32073, 0.384307, 0.353491]

Minimum: 0.100426 Maximum: 0.62181 Average: 0.2999 seconds

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

3 processes used for testing 5 window sizes

Processing times for each image [4.7973, 5.6143, 5.0966, 5.0994, 5.2143, 4.5707] with an average

Process-2024, Step 5: Compute individual channel HOG features for the entire image: 0.30158 seconds
Process-2024, Step 6: Misc initializations: 1e-06 seconds
Process-2024, Step 7: for loop: 0.020271 seconds

Process-2024All steps time: 0.3326769999999994 seconds

Process-2024, Find cars processing time: 0.332749 seconds

Process-2023, Step 1: Divide with 255, processing time: 0.010116 seconds
Process-2023, Step 2: Resize if scale is not 1: 0.001287 seconds
Process-2023, Step 3: Get HOG channels: 1e-05 seconds
Process-2023, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2023, Step 5: Compute individual channel HOG features for the entire image: 0.333234 seconds
Process-2023, Step 6: Misc initializations: 1e-06 seconds
Process-2023, Step 7: for loop: 0.019088 seconds

Process-2023All steps time: 0.3637449999999993 seconds

Process-2023, Find cars processing time: 0.363838 seconds

Process-2026, Step 1: Divide with 255, processing time: 0.027467 seconds
Process-2026, Step 2: Resize if scale is not 1: 0.011279 seconds
Process-2026, Step 3: Get HOG channels: 6.8e-05 seconds
Process-2026, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2026, Step 5: Compute individual channel HOG features for the entire image: 0.458968 seconds
Process-2026, Step 6: Misc initializations: 1e-06 seconds
Process-2026, Step 7: for loop: 0.032351 seconds

Process-2026All steps time: 0.530142 seconds

Process-2026, Find cars processing time: 0.530362 seconds

Process-2025, Step 1: Divide with 255, processing time: 0.009685 seconds
Process-2025, Step 2: Resize if scale is not 1: 0.000825 seconds
Process-2025, Step 3: Get HOG channels: 5e-06 seconds
Process-2025, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2025, Step 5: Compute individual channel HOG features for the entire image: 0.295877 seconds
Process-2025, Step 6: Misc initializations: 1e-06 seconds
Process-2025, Step 7: for loop: 0.019019 seconds

Process-2025All steps time: 0.3254169999999996 seconds

Process-2025, Find cars processing time: 0.325498 seconds

Process-2024, Step 1: Divide with 255, processing time: 0.009084 seconds
Process-2024, Step 2: Resize if scale is not 1: 0.000962 seconds
Process-2024, Step 3: Get HOG channels: 7e-06 seconds
Process-2024, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2024, Step 5: Compute individual channel HOG features for the entire image: 0.274031 seconds
Process-2024, Step 6: Misc initializations: 5e-06 seconds
Process-2024, Step 7: for loop: 0.020375 seconds

Process-2024All steps time: 0.30447199999999996 seconds

Process-2024, Find cars processing time: 0.304536 seconds

Process-2023, Step 1: Divide with 255, processing time: 0.008588 seconds
Process-2023, Step 2: Resize if scale is not 1: 0.000798 seconds
Process-2023, Step 3: Get HOG channels: 7e-06 seconds
Process-2023, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2023, Step 5: Compute individual channel HOG features for the entire image: 0.199919 seconds
Process-2023, Step 6: Misc initializations: 1e-06 seconds
Process-2023, Step 7: for loop: 0.012652 seconds

Process-2023All steps time: 0.221972 seconds

Process-2023, Find cars processing time: 0.222029 seconds

Process-2025, Step 1: Divide with 255, processing time: 0.008583 seconds
Process-2025, Step 2: Resize if scale is not 1: 0.000747 seconds
Process-2025, Step 3: Get HOG channels: 6e-06 seconds
Process-2025, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2025, Step 5: Compute individual channel HOG features for the entire image: 0.192158 seconds
Process-2025, Step 6: Misc initializations: 0.0 seconds
Process-2025, Step 7: for loop: 0.012897 seconds

Process-2025All steps time: 0.2143979999999998 seconds

Process-2025, Find cars processing time: 0.214457 seconds

Process-2023, Step 1: Divide with 255, processing time: 0.005934 seconds
Process-2023, Step 2: Resize if scale is not 1: 0.000638 seconds
Process-2023, Step 3: Get HOG channels: 6e-06 seconds
Process-2023, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2023, Step 5: Compute individual channel HOG features for the entire image: 0.140846 seconds
Process-2023, Step 6: Misc initializations: 1e-06 seconds
Process-2023, Step 7: for loop: 0.00783 seconds

Process-2023All steps time: 0.15526 seconds

Process-2023, Find cars processing time: 0.155316 seconds

Process-2024, Step 1: Divide with 255, processing time: 0.010732 seconds
Process-2024, Step 2: Resize if scale is not 1: 0.000521 seconds
Process-2024, Step 3: Get HOG channels: 5e-06 seconds
Process-2024, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2024, Step 5: Compute individual channel HOG features for the entire image: 0.246953 seconds
Process-2024, Step 6: Misc initializations: 1e-06 seconds
Process-2024, Step 7: for loop: 0.013949 seconds

Process-2024All steps time: 0.2721659999999996 seconds

Process-2024, Find cars processing time: 0.272233 seconds

Process-2025, Step 1: Divide with 255, processing time: 0.010485 seconds
Process-2025, Step 2: Resize if scale is not 1: 0.000892 seconds
Process-2025, Step 3: Get HOG channels: 7e-06 seconds
Process-2025, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2025, Step 5: Compute individual channel HOG features for the entire image: 0.146113 seconds
Process-2025, Step 6: Misc initializations: 1e-06 seconds
Process-2025, Step 7: for loop: 0.007955 seconds

Process-2025All steps time: 0.16546 seconds

Process-2023, Step 1: Divide with 255, processing time: 0.008885 seconds

Process-2025, Find cars processing time: 0.165524 seconds

Process-2023, Step 2: Resize if scale is not 1: 0.000741 seconds

Process-2023, Step 3: Get HOG channels: 6e-06 seconds
Process-2023, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2023, Step 5: Compute individual channel HOG features for the entire image: 0.146472 seconds
Process-2023, Step 6: Misc initializations: 0.0 seconds
Process-2023, Step 7: for loop: 0.008254 seconds

Process-2023All steps time: 0.164364 seconds

Process-2023, Find cars processing time: 0.164418 seconds

Process-2026, Step 1: Divide with 255, processing time: 0.009928 seconds

Process-2026, Step 2: Resize if scale is not 1: 0.001044 seconds

Process-2026, Step 3: Get HOG channels: 5e-06 seconds
Process-2026, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2026, Step 5: Compute individual channel HOG features for the entire image: 0.231761 seconds
Process-2026, Step 6: Misc initializations: 1e-06 seconds
Process-2026, Step 7: for loop: 0.014722 seconds

Process-2026All steps time: 0.257466 seconds

Process-2026, Find cars processing time: 0.257545 seconds

Process-2024, Step 1: Divide with 255, processing time: 0.006751 seconds

Process-2024, Step 2: Resize if scale is not 1: 0.000632 seconds

Process-2024, Step 3: Get HOG channels: 6e-06 seconds

Process-2024, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2024, Step 5: Compute individual channel HOG features for the entire image: 0.148441 seconds
Process-2024, Step 6: Misc initializations: 1e-06 seconds
Process-2024, Step 7: for loop: 0.008874 seconds

Process-2024All steps time: 0.16471 seconds

Process-2024, Find cars processing time: 0.164758 seconds

Process-2026, Step 1: Divide with 255, processing time: 0.010839 seconds
Process-2026, Step 2: Resize if scale is not 1: 0.000964 seconds
Process-2026, Step 3: Get HOG channels: 7e-06 seconds
Process-2026, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2026, Step 5: Compute individual channel HOG features for the entire image: 0.131845 seconds
Process-2026, Step 6: Misc initializations: 1e-06 seconds
Process-2026, Step 7: for loop: 0.007167 seconds

Process-2026All steps time: 0.150829 seconds

Process-2026, Find cars processing time: 0.150901 seconds

Process-2023, Step 1: Divide with 255, processing time: 0.010581 seconds
Process-2023, Step 2: Resize if scale is not 1: 0.001251 seconds
Process-2023, Step 3: Get HOG channels: 1.3e-05 seconds
Process-2023, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2023, Step 5: Compute individual channel HOG features for the entire image: 0.109633 seconds
Process-2023, Step 6: Misc initializations: 0.0 seconds
Process-2023, Step 7: for loop: 0.020602 seconds

Process-2023All steps time: 0.142089 seconds

Process-2023, Find cars processing time: 0.142179 seconds

Process-2024, Step 1: Divide with 255, processing time: 0.010503 seconds
Process-2024, Step 2: Resize if scale is not 1: 0.001193 seconds
Process-2024, Step 3: Get HOG channels: 8e-06 seconds
Process-2024, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2024, Step 5: Compute individual channel HOG features for the entire image: 0.103861 seconds
Process-2024, Step 6: Misc initializations: 0.0 seconds
Process-2024, Step 7: for loop: 0.005916 seconds

Process-2024All steps time: 0.121488 seconds

Process-2024, Find cars processing time: 0.121547 seconds

Process-2026, Step 1: Divide with 255, processing time: 0.009962 seconds
Process-2026, Step 2: Resize if scale is not 1: 0.001903 seconds
Process-2026, Step 3: Get HOG channels: 7e-06 seconds
Process-2026, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2026, Step 5: Compute individual channel HOG features for the entire image: 0.110606 seconds
Process-2026, Step 6: Misc initializations: 1e-06 seconds
Process-2026, Step 7: for loop: 0.005868 seconds

Process-2026All steps time: 0.128355 seconds

Process-2026, Find cars processing time: 0.128418 seconds

Process-2025, Step 1: Divide with 255, processing time: 0.010471 seconds
Process-2025, Step 2: Resize if scale is not 1: 0.000792 seconds
Process-2025, Step 3: Get HOG channels: 6e-06 seconds
Process-2025, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2025, Step 5: Compute individual channel HOG features for the entire image: 0.135252 seconds
Process-2025, Step 6: Misc initializations: 0.0 seconds
Process-2025, Step 7: for loop: 0.005716 seconds

Process-2025All steps time: 0.15224400000000002 seconds

Process-2025, Find cars processing time: 0.152308 seconds

The times for each task are: [0.558372, 0.62378, 0.562586, 0.332749, 0.363838, 0.530362, 0.32549]

Minimum: 0.121547 Maximum: 0.62378 Average: 0.2816 seconds

Number of processes used: 4 window size 260
Length of task list: 21
Number of processes used: 4

Process-2027, Step 1: Divide with 255, processing time: 0.029764 seconds
Process-2027, Step 2: Resize if scale is not 1: 0.010957 seconds
Process-2027, Step 3: Get HOG channels: 9.8e-05 seconds
Process-2027, Step 4: Define blocks and steps as above: 1.6e-05 seconds
Process-2027, Step 5: Compute individual channel HOG features for the entire image: 0.475018 seconds
Process-2027, Step 6: Misc initializations: 1e-06 seconds
Process-2027, Step 7: for loop: 0.037114 seconds

Process-2027All steps time: 0.552968 seconds

Process-2027, Find cars processing time: 0.55324 seconds

Process-2028, Step 1: Divide with 255, processing time: 0.028871 seconds
Process-2028, Step 2: Resize if scale is not 1: 0.013442 seconds
Process-2028, Step 3: Get HOG channels: 6.2e-05 seconds
Process-2028, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2028, Step 5: Compute individual channel HOG features for the entire image: 0.574111 seconds
Process-2028, Step 6: Misc initializations: 1e-06 seconds
Process-2028, Step 7: for loop: 0.03616 seconds

Process-2028All steps time: 0.652657 seconds

Process-2028, Find cars processing time: 0.652923 seconds

Process-2029, Step 1: Divide with 255, processing time: 0.024971 seconds

Process-2029, Step 2: Resize if scale is not 1: 0.011372 seconds

Process-2029, Step 3: Get HOG channels: 6.6e-05 seconds

Process-2029, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2029, Step 5: Compute individual channel HOG features for the entire image: 0.529992 seconds

Process-2029, Step 6: Misc initializations: 1e-06 seconds

Process-2029, Step 7: for loop: 0.038045 seconds

Process-2029All steps time: 0.6044550000000001 seconds

Process-2029, Find cars processing time: 0.604653 seconds

Process-2027, Step 1: Divide with 255, processing time: 0.01032 seconds

Process-2027, Step 2: Resize if scale is not 1: 0.000826 seconds

Process-2027, Step 3: Get HOG channels: 7e-06 seconds

Process-2027, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2027, Step 5: Compute individual channel HOG features for the entire image: 0.367429 seconds

Process-2027, Step 6: Misc initializations: 1e-06 seconds

Process-2027, Step 7: for loop: 0.019391 seconds

Process-2027All steps time: 0.3979799999999994 seconds

Process-2027, Find cars processing time: 0.398061 seconds

Process-2030, Step 1: Divide with 255, processing time: 0.025788 seconds

Process-2030, Step 2: Resize if scale is not 1: 0.018001 seconds

Process-2030, Step 3: Get HOG channels: 2.2e-05 seconds

Process-2030, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2030, Step 5: Compute individual channel HOG features for the entire image: 0.450257 seconds

Process-2030, Step 6: Misc initializations: 1e-06 seconds

Process-2030, Step 7: for loop: 0.032068 seconds

Process-2030All steps time: 0.526143 seconds

Process-2030, Find cars processing time: 0.526293 seconds

Process-2029, Step 1: Divide with 255, processing time: 0.010473 seconds

Process-2029, Step 2: Resize if scale is not 1: 0.001161 seconds

Process-2029, Step 3: Get HOG channels: 9e-06 seconds

Process-2029, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2029, Step 5: Compute individual channel HOG features for the entire image: 0.331655 seconds

Process-2029, Step 6: Misc initializations: 2e-06 seconds

Process-2029, Step 7: for loop: 0.02045 seconds

Process-2029All steps time: 0.363758 seconds

Process-2029, Find cars processing time: 0.363846 seconds

Process-2028, Step 1: Divide with 255, processing time: 0.010975 seconds

Process-2028, Step 2: Resize if scale is not 1: 0.001169 seconds

Process-2028, Step 3: Get HOG channels: 8e-06 seconds

Process-2028, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2028, Step 5: Compute individual channel HOG features for the entire image: 0.298999 seconds

Process-2028, Step 6: Misc initializations: 1e-06 seconds

Process-2028, Step 7: for loop: 0.020913 seconds

Process-2028All steps time: 0.332073 seconds

Process-2028, Find cars processing time: 0.33214 seconds

Process-2030, Step 1: Divide with 255, processing time: 0.00996 seconds

Process-2030, Step 2: Resize if scale is not 1: 0.001539 seconds

Process-2030, Step 3: Get HOG channels: 7e-06 seconds

Process-2030, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2030, Step 5: Compute individual channel HOG features for the entire image: 0.195287 seconds

Process-2030, Step 6: Misc initializations: 0.0 seconds

Process-2030, Step 7: for loop: 0.012984 seconds

Process-2030All steps time: 0.2197849999999998 seconds

Process-2030, Find cars processing time: 0.219847 seconds

Process-2027, Step 1: Divide with 255, processing time: 0.011029 seconds

Process-2029, Step 1: Divide with 255, processing time: 0.009541 seconds

Process-2027, Step 2: Resize if scale is not 1: 0.001072 seconds

Process-2029, Step 2: Resize if scale is not 1: 0.000992 seconds

Process-2027, Step 3: Get HOG channels: 7e-06 seconds

Process-2029, Step 3: Get HOG channels: 5e-06 seconds

Process-2027, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2029, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2027, Step 5: Compute individual channel HOG features for the entire image: 0.284466 seconds

Process-2029, Step 5: Compute individual channel HOG features for the entire image: 0.192068 seconds

Process-2027, Step 6: Misc initializations: 3e-06 seconds

Process-2029, Step 6: Misc initializations: 1e-06 seconds

Process-2027, Step 7: for loop: 0.021159 seconds

Process-2029, Step 7: for loop: 0.013654 seconds

Process-2027All steps time: 0.3177429999999994 seconds

Process-2029All steps time: 0.2162649999999999 seconds

```
Process-2027, Find cars processing time: 0.317862 seconds
Process-2029, Find cars processing time: 0.216333 seconds
```

```
Process-2027, Step 1: Divide with 255, processing time: 0.007455 seconds
Process-2027, Step 2: Resize if scale is not 1: 0.001153 seconds
Process-2027, Step 3: Get HOG channels: 8e-06 seconds
Process-2027, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2027, Step 5: Compute individual channel HOG features for the entire image: 0.13313 seconds
Process-2027, Step 6: Misc initializations: 0.0 seconds
Process-2027, Step 7: for loop: 0.007406 seconds
```

```
Process-2027All steps time: 0.1491579999999999 seconds
```

```
Process-2027, Find cars processing time: 0.149223 seconds
```

```
Process-2028, Step 1: Divide with 255, processing time: 0.010511 seconds
Process-2028, Step 2: Resize if scale is not 1: 0.000582 seconds
Process-2028, Step 3: Get HOG channels: 5e-06 seconds
Process-2028, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2028, Step 5: Compute individual channel HOG features for the entire image: 0.238729 seconds
Process-2028, Step 6: Misc initializations: 1e-06 seconds
Process-2028, Step 7: for loop: 0.012956 seconds
```

```
Process-2028All steps time: 0.262789 seconds
```

```
Process-2028, Find cars processing time: 0.262853 seconds
```

```
Process-2027, Step 1: Divide with 255, processing time: 0.005993 seconds
Process-2027, Step 2: Resize if scale is not 1: 0.000465 seconds
Process-2027, Step 3: Get HOG channels: 5e-06 seconds
Process-2027, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2027, Step 5: Compute individual channel HOG features for the entire image: 0.130969 seconds
Process-2027, Step 6: Misc initializations: 1e-06 seconds
Process-2027, Step 7: for loop: 0.0085 seconds
```

```
Process-2027All steps time: 0.145938 seconds
```

```
Process-2027, Find cars processing time: 0.146001 seconds
```

```
Process-2029, Step 1: Divide with 255, processing time: 0.010424 seconds
Process-2029, Step 2: Resize if scale is not 1: 0.000825 seconds
Process-2029, Step 3: Get HOG channels: 8e-06 seconds
Process-2029, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2029, Step 5: Compute individual channel HOG features for the entire image: 0.138355 seconds
Process-2029, Step 6: Misc initializations: 0.0 seconds
Process-2029, Step 7: for loop: 0.008089 seconds
```

Process-2029All steps time: 0.15770800000000001 seconds

Process-2029, Find cars processing time: 0.15777 seconds

Process-2028, Step 1: Divide with 255, processing time: 0.009266 seconds

Process-2028, Step 2: Resize if scale is not 1: 0.000855 seconds

Process-2028, Step 3: Get HOG channels: 7e-06 seconds

Process-2028, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2028, Step 5: Compute individual channel HOG features for the entire image: 0.136938 seconds

Process-2028, Step 6: Misc initializations: 0.0 seconds

Process-2028, Step 7: for loop: 0.007957 seconds

Process-2028All steps time: 0.155029 seconds

Process-2028, Find cars processing time: 0.15512 seconds

Process-2030, Step 1: Divide with 255, processing time: 0.010152 seconds

Process-2030, Step 2: Resize if scale is not 1: 0.000726 seconds

Process-2030, Step 3: Get HOG channels: 7e-06 seconds

Process-2030, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2030, Step 5: Compute individual channel HOG features for the entire image: 0.272783 seconds

Process-2030, Step 6: Misc initializations: 1e-06 seconds

Process-2030, Step 7: for loop: 0.013817 seconds

Process-2030All steps time: 0.297491 seconds

Process-2030, Find cars processing time: 0.297564 seconds

Process-2027, Step 1: Divide with 255, processing time: 0.011313 seconds

Process-2027, Step 2: Resize if scale is not 1: 0.001232 seconds

Process-2027, Step 3: Get HOG channels: 8e-06 seconds

Process-2027, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2027, Step 5: Compute individual channel HOG features for the entire image: 0.105213 seconds

Process-2027, Step 6: Misc initializations: 0.0 seconds

Process-2027, Step 7: for loop: 0.006482 seconds

Process-2027All steps time: 0.124256 seconds

Process-2027, Find cars processing time: 0.124329 seconds

Process-2028, Step 1: Divide with 255, processing time: 0.022749 seconds

Process-2028, Step 2: Resize if scale is not 1: 0.000838 seconds

Process-2028, Step 3: Get HOG channels: 4e-06 seconds

Process-2028, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2028, Step 5: Compute individual channel HOG features for the entire image: 0.103296 seconds

Process-2028, Step 6: Misc initializations: 0.0 seconds

Process-2028, Step 7: for loop: 0.009237 seconds

Process-2028All steps time: 0.136128 seconds

Process-2028, Find cars processing time: 0.136185 seconds

Process-2030, Step 1: Divide with 255, processing time: 0.013433 seconds

Process-2030, Step 2: Resize if scale is not 1: 0.000787 seconds

Process-2030, Step 3: Get HOG channels: 5e-06 seconds

Process-2030, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2030, Step 5: Compute individual channel HOG features for the entire image: 0.120856 seconds

Process-2030, Step 6: Misc initializations: 0.0 seconds

Process-2030, Step 7: for loop: 0.00599 seconds

Process-2030All steps time: 0.141075 seconds

Process-2030, Find cars processing time: 0.141139 seconds

Process-2029, Step 1: Divide with 255, processing time: 0.01435 seconds

Process-2029, Step 2: Resize if scale is not 1: 0.000643 seconds

Process-2029, Step 3: Get HOG channels: 4e-06 seconds

Process-2029, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2029, Step 5: Compute individual channel HOG features for the entire image: 0.101488 seconds

Process-2029, Step 6: Misc initializations: 1e-06 seconds

Process-2029, Step 7: for loop: 0.006904 seconds

Process-2029All steps time: 0.123394 seconds

Process-2029, Find cars processing time: 0.123439 seconds

Process-2027, Step 1: Divide with 255, processing time: 0.035226 seconds

Process-2027, Step 2: Resize if scale is not 1: 0.001128 seconds

Process-2027, Step 3: Get HOG channels: 5e-06 seconds

Process-2027, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2027, Step 5: Compute individual channel HOG features for the entire image: 0.084628 seconds

Process-2027, Step 6: Misc initializations: 0.0 seconds

Process-2027, Step 7: for loop: 0.004641 seconds

Process-2027All steps time: 0.125633 seconds

Process-2027, Find cars processing time: 0.125676 seconds

The times for each task are: [0.55324, 0.652923, 0.604653, 0.398061, 0.526293, 0.363846, 0.33214

Minimum: 0.123439 Maximum: 0.652923 Average: 0.2859 seconds

Number of processes used: 4 window size 260

Length of task list: 21

Number of processes used: 4

Process-2031, Step 1: Divide with 255, processing time: 0.026097 seconds
Process-2031, Step 2: Resize if scale is not 1: 0.011728 seconds
Process-2031, Step 3: Get HOG channels: 8.3e-05 seconds
Process-2031, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-2031, Step 5: Compute individual channel HOG features for the entire image: 0.488709 seconds
Process-2031, Step 6: Misc initializations: 1e-06 seconds
Process-2031, Step 7: for loop: 0.038053 seconds

Process-2031All steps time: 0.564682 seconds

Process-2031, Find cars processing time: 0.564921 seconds

Process-2032, Step 1: Divide with 255, processing time: 0.029784 seconds
Process-2032, Step 2: Resize if scale is not 1: 0.012152 seconds
Process-2032, Step 3: Get HOG channels: 0.000177 seconds
Process-2032, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-2032, Step 5: Compute individual channel HOG features for the entire image: 0.519389 seconds
Process-2032, Step 6: Misc initializations: 1e-06 seconds
Process-2032, Step 7: for loop: 0.03633 seconds

Process-2032All steps time: 0.5978439999999999 seconds

Process-2032, Find cars processing time: 0.59814 seconds

Process-2031, Step 1: Divide with 255, processing time: 0.010285 seconds
Process-2031, Step 2: Resize if scale is not 1: 0.001037 seconds
Process-2031, Step 3: Get HOG channels: 8e-06 seconds
Process-2031, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2031, Step 5: Compute individual channel HOG features for the entire image: 0.286805 seconds
Process-2031, Step 6: Misc initializations: 1e-06 seconds
Process-2031, Step 7: for loop: 0.02 seconds

Process-2031All steps time: 0.3181449999999996 seconds

Process-2031, Find cars processing time: 0.318212 seconds

Process-2033, Step 1: Divide with 255, processing time: 0.042592 seconds
Process-2033, Step 2: Resize if scale is not 1: 0.013398 seconds
Process-2033, Step 3: Get HOG channels: 0.000101 seconds
Process-2033, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-2033, Step 5: Compute individual channel HOG features for the entire image: 0.529188 seconds
Process-2033, Step 6: Misc initializations: 1e-06 seconds
Process-2033, Step 7: for loop: 0.037021 seconds

Process-2033All steps time: 0.622314 seconds

Process-2033, Find cars processing time: 0.622652 seconds

Process-2032, Step 1: Divide with 255, processing time: 0.008838 seconds

Process-2032, Step 2: Resize if scale is not 1: 0.000803 seconds

Process-2032, Step 3: Get HOG channels: 5e-06 seconds

Process-2032, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2032, Step 5: Compute individual channel HOG features for the entire image: 0.30077 seconds

Process-2032, Step 6: Misc initializations: 1e-06 seconds

Process-2032, Step 7: for loop: 0.020703 seconds

Process-2032All steps time: 0.3311259999999999 seconds

Process-2032, Find cars processing time: 0.331183 seconds

Process-2034, Step 1: Divide with 255, processing time: 0.055045 seconds

Process-2034, Step 2: Resize if scale is not 1: 0.016701 seconds

Process-2034, Step 3: Get HOG channels: 5.8e-05 seconds

Process-2034, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2034, Step 5: Compute individual channel HOG features for the entire image: 0.509702 seconds

Process-2034, Step 6: Misc initializations: 2e-06 seconds

Process-2034, Step 7: for loop: 0.042079 seconds

Process-2034All steps time: 0.623597 seconds

Process-2034, Find cars processing time: 0.623865 seconds

Process-2033, Step 1: Divide with 255, processing time: 0.010553 seconds

Process-2033, Step 2: Resize if scale is not 1: 0.001197 seconds

Process-2033, Step 3: Get HOG channels: 8e-06 seconds

Process-2033, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2033, Step 5: Compute individual channel HOG features for the entire image: 0.289214 seconds

Process-2033, Step 6: Misc initializations: 0.0 seconds

Process-2033, Step 7: for loop: 0.02089 seconds

Process-2033All steps time: 0.3218700000000004 seconds

Process-2033, Find cars processing time: 0.321941 seconds

Process-2034, Step 1: Divide with 255, processing time: 0.010668 seconds

Process-2034, Step 2: Resize if scale is not 1: 0.001652 seconds

Process-2034, Step 3: Get HOG channels: 8e-06 seconds

Process-2034, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2034, Step 5: Compute individual channel HOG features for the entire image: 0.262565 seconds

Process-2034, Step 6: Misc initializations: 1e-06 seconds

Process-2034, Step 7: for loop: 0.016402 seconds

Process-2034All steps time: 0.291304 seconds

Process-2034, Find cars processing time: 0.291384 seconds

Process-2032, Step 1: Divide with 255, processing time: 0.013549 seconds

Process-2032, Step 2: Resize if scale is not 1: 0.000497 seconds

Process-2032, Step 3: Get HOG channels: 4e-06 seconds

Process-2032, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2032, Step 5: Compute individual channel HOG features for the entire image: 0.198467 seconds

Process-2032, Step 6: Misc initializations: 0.0 seconds

Process-2032, Step 7: for loop: 0.013324 seconds

Process-2032All steps time: 0.22584600000000002 seconds

Process-2032, Find cars processing time: 0.225905 seconds

Process-2031, Step 1: Divide with 255, processing time: 0.017217 seconds

Process-2031, Step 2: Resize if scale is not 1: 0.000974 seconds

Process-2031, Step 3: Get HOG channels: 5e-06 seconds

Process-2031, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2031, Step 5: Compute individual channel HOG features for the entire image: 0.329652 seconds

Process-2031, Step 6: Misc initializations: 1e-06 seconds

Process-2031, Step 7: for loop: 0.019986 seconds

Process-2031All steps time: 0.367841 seconds

Process-2031, Find cars processing time: 0.3679 seconds

Process-2033, Step 1: Divide with 255, processing time: 0.010526 seconds

Process-2033, Step 2: Resize if scale is not 1: 0.000762 seconds

Process-2033, Step 3: Get HOG channels: 5e-06 seconds

Process-2033, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2033, Step 5: Compute individual channel HOG features for the entire image: 0.199343 seconds

Process-2033, Step 6: Misc initializations: 0.0 seconds

Process-2033, Step 7: for loop: 0.012027 seconds

Process-2033All steps time: 0.222668 seconds

Process-2033, Find cars processing time: 0.222721 seconds

Process-2033, Step 1: Divide with 255, processing time: 0.010565 seconds

Process-2033, Step 2: Resize if scale is not 1: 0.000841 seconds

Process-2033, Step 3: Get HOG channels: 7e-06 seconds

Process-2033, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2033, Step 5: Compute individual channel HOG features for the entire image: 0.133365 seconds

Process-2033, Step 6: Misc initializations: 0.0 seconds

Process-2033, Step 7: for loop: 0.007519 seconds

Process-2033All steps time: 0.152304 seconds

Process-2033, Find cars processing time: 0.152367 seconds

Process-2032, Step 1: Divide with 255, processing time: 0.010319 seconds

Process-2032, Step 2: Resize if scale is not 1: 0.001204 seconds

Process-2032, Step 3: Get HOG channels: 1.5e-05 seconds

Process-2032, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-2032, Step 5: Compute individual channel HOG features for the entire image: 0.20481 seconds

Process-2032, Step 6: Misc initializations: 0.0 seconds

Process-2032, Step 7: for loop: 0.008943 seconds

Process-2032All steps time: 0.225303 seconds

Process-2032, Find cars processing time: 0.225399 seconds

Process-2031, Step 1: Divide with 255, processing time: 0.009294 seconds

Process-2031, Step 2: Resize if scale is not 1: 0.000979 seconds

Process-2031, Step 3: Get HOG channels: 6e-06 seconds

Process-2031, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2031, Step 5: Compute individual channel HOG features for the entire image: 0.141596 seconds

Process-2031, Step 6: Misc initializations: 1e-06 seconds

Process-2031, Step 7: for loop: 0.007888 seconds

Process-2031All steps time: 0.15977 seconds

Process-2031, Find cars processing time: 0.159827 seconds

Process-2034, Step 1: Divide with 255, processing time: 0.010803 seconds

Process-2034, Step 2: Resize if scale is not 1: 0.000923 seconds

Process-2034, Step 3: Get HOG channels: 8e-06 seconds

Process-2034, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2034, Step 5: Compute individual channel HOG features for the entire image: 0.264833 seconds

Process-2034, Step 6: Misc initializations: 1e-06 seconds

Process-2034, Step 7: for loop: 0.023646 seconds

Process-2034All steps time: 0.3002229999999996 seconds

Process-2034, Find cars processing time: 0.300317 seconds

Process-2033, Step 1: Divide with 255, processing time: 0.009189 seconds

Process-2033, Step 2: Resize if scale is not 1: 0.000626 seconds

Process-2033, Step 3: Get HOG channels: 6e-06 seconds

Process-2033, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2033, Step 5: Compute individual channel HOG features for the entire image: 0.147959 seconds

Process-2033, Step 6: Misc initializations: 1e-06 seconds

Process-2033, Step 7: for loop: 0.007429 seconds

Process-2033All steps time: 0.165216 seconds

Process-2033, Find cars processing time: 0.165282 seconds

Process-2032, Step 1: Divide with 255, processing time: 0.007627 seconds

Process-2032, Step 2: Resize if scale is not 1: 0.000998 seconds

Process-2032, Step 3: Get HOG channels: 6e-06 seconds

Process-2032, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2032, Step 5: Compute individual channel HOG features for the entire image: 0.137147 seconds

Process-2032, Step 6: Misc initializations: 1e-06 seconds

Process-2032, Step 7: for loop: 0.006089 seconds

Process-2032All steps time: 0.151874 seconds

Process-2032, Find cars processing time: 0.151948 seconds

Process-2031, Step 1: Divide with 255, processing time: 0.010867 seconds

Process-2031, Step 2: Resize if scale is not 1: 0.001316 seconds

Process-2031, Step 3: Get HOG channels: 6e-06 seconds

Process-2031, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2031, Step 5: Compute individual channel HOG features for the entire image: 0.107197 seconds

Process-2031, Step 6: Misc initializations: 1e-06 seconds

Process-2031, Step 7: for loop: 0.005783 seconds

Process-2031All steps time: 0.125177 seconds

Process-2031, Find cars processing time: 0.125263 seconds

Process-2034, Step 1: Divide with 255, processing time: 0.011878 seconds

Process-2034, Step 2: Resize if scale is not 1: 0.000867 seconds

Process-2034, Step 3: Get HOG channels: 7e-06 seconds

Process-2034, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2032, Step 1: Divide with 255, processing time: 0.007846 seconds

Process-2034, Step 5: Compute individual channel HOG features for the entire image: 0.137311 seconds

Process-2032, Step 6: Resize if scale is not 1: 0.000697 seconds

Process-2034, Step 7: Misc initializations: 0.0 seconds

Process-2032, Step 8: Get HOG channels: 7e-06 seconds

Process-2034, Step 9: for loop: 0.007643 seconds

Process-2032, Step 10: Define blocks and steps as above: 5e-06 seconds

Process-2032, Step 11: Compute individual channel HOG features for the entire image: 0.105184 seconds

Process-2032, Step 12: Misc initializations: 1e-06 seconds

Process-2034All steps time: 0.157712 seconds

Process-2032, Step 13: for loop: 0.005675 seconds

Process-2034, Find cars processing time: 0.157788 seconds

Process-2032All steps time: 0.11941500000000001 seconds

Process-2032, Find cars processing time: 0.11948 seconds

Process-2033, Step 1: Divide with 255, processing time: 0.010288 seconds

Process-2033, Step 2: Resize if scale is not 1: 0.001054 seconds

Process-2033, Step 3: Get HOG channels: 7e-06 seconds

Process-2033, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2033, Step 5: Compute individual channel HOG features for the entire image: 0.098235 seconds

Process-2033, Step 6: Misc initializations: 0.0 seconds

Process-2033, Step 7: for loop: 0.005476 seconds

Process-2033All steps time: 0.11506799999999999 seconds

Process-2033, Find cars processing time: 0.115129 seconds

The times for each task are: [0.564921, 0.59814, 0.318212, 0.622652, 0.331183, 0.623865, 0.32194

Minimum: 0.115129 Maximum: 0.623865 Average: 0.2934 seconds

Number of processes used: 4 window size 260

Length of task list: 21

Number of processes used: 4

Process-2035, Step 1: Divide with 255, processing time: 0.027204 seconds

Process-2035, Step 2: Resize if scale is not 1: 0.012217 seconds

Process-2035, Step 3: Get HOG channels: 7.1e-05 seconds

Process-2035, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2035, Step 5: Compute individual channel HOG features for the entire image: 0.496144 seconds

Process-2035, Step 6: Misc initializations: 1e-06 seconds

Process-2035, Step 7: for loop: 0.037102 seconds

Process-2035All steps time: 0.572749 seconds

Process-2035, Find cars processing time: 0.573025 seconds

Process-2036, Step 1: Divide with 255, processing time: 0.027519 seconds

Process-2036, Step 2: Resize if scale is not 1: 0.011846 seconds

Process-2036, Step 3: Get HOG channels: 8.9e-05 seconds

Process-2036, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-2036, Step 5: Compute individual channel HOG features for the entire image: 0.60502 seconds

Process-2036, Step 6: Misc initializations: 1e-06 seconds

Process-2036, Step 7: for loop: 0.0398 seconds

Process-2036All steps time: 0.684288 seconds

Process-2036, Find cars processing time: 0.684602 seconds

Process-2035, Step 1: Divide with 255, processing time: 0.008737 seconds

Process-2035, Step 2: Resize if scale is not 1: 0.000801 seconds
Process-2035, Step 3: Get HOG channels: 6e-06 seconds
Process-2035, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2035, Step 5: Compute individual channel HOG features for the entire image: 0.35751 seconds
Process-2035, Step 6: Misc initializations: 0.0 seconds
Process-2035, Step 7: for loop: 0.020927 seconds

Process-2035All steps time: 0.38798699999999997 seconds

Process-2035, Find cars processing time: 0.388046 seconds

Process-2038, Step 1: Divide with 255, processing time: 0.025452 seconds
Process-2038, Step 2: Resize if scale is not 1: 0.01048 seconds
Process-2038, Step 3: Get HOG channels: 6.5e-05 seconds
Process-2038, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2038, Step 5: Compute individual channel HOG features for the entire image: 0.482718 seconds
Process-2038, Step 6: Misc initializations: 1e-06 seconds
Process-2038, Step 7: for loop: 0.033861 seconds

Process-2038All steps time: 0.5525840000000001 seconds

Process-2038, Find cars processing time: 0.552854 seconds

Process-2037, Step 1: Divide with 255, processing time: 0.029387 seconds
Process-2037, Step 5: Compute individual channel HOG features for the entire image: 0.569235 seconds
Process-2037, Step 2: Resize if scale is not 1: 0.010936 seconds
Process-2037, Step 3: Get HOG channels: 8.5e-05 seconds
Process-2037, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2037, Step 6: Misc initializations: 1e-06 seconds
Process-2037, Step 7: for loop: 0.040737 seconds

Process-2037All steps time: 0.6503900000000001 seconds

Process-2037, Find cars processing time: 0.650681 seconds

Process-2035, Step 1: Divide with 255, processing time: 0.011598 seconds
Process-2035, Step 2: Resize if scale is not 1: 0.001111 seconds
Process-2035, Step 3: Get HOG channels: 8e-06 seconds
Process-2035, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2035, Step 5: Compute individual channel HOG features for the entire image: 0.274577 seconds
Process-2035, Step 6: Misc initializations: 1e-06 seconds
Process-2035, Step 7: for loop: 0.019485 seconds

Process-2035All steps time: 0.306789 seconds

Process-2035, Find cars processing time: 0.30688 seconds

Process-2036, Step 1: Divide with 255, processing time: 0.01181 seconds

Process-2036, Step 2: Resize if scale is not 1: 0.001133 seconds
Process-2036, Step 3: Get HOG channels: 9e-06 seconds
Process-2036, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2036, Step 5: Compute individual channel HOG features for the entire image: 0.343235 seconds
Process-2036, Step 6: Misc initializations: 1e-06 seconds
Process-2036, Step 7: for loop: 0.019488 seconds

Process-2036All steps time: 0.375683 seconds

Process-2036, Find cars processing time: 0.375746 seconds

Process-2037, Step 1: Divide with 255, processing time: 0.010424 seconds
Process-2037, Step 2: Resize if scale is not 1: 0.001032 seconds
Process-2037, Step 3: Get HOG channels: 5e-06 seconds
Process-2037, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2037, Step 5: Compute individual channel HOG features for the entire image: 0.267469 seconds
Process-2037, Step 6: Misc initializations: 1e-06 seconds
Process-2037, Step 7: for loop: 0.017191 seconds

Process-2037All steps time: 0.296127 seconds

Process-2037, Find cars processing time: 0.296188 seconds

Process-2038, Step 1: Divide with 255, processing time: 0.009561 seconds
Process-2038, Step 2: Resize if scale is not 1: 0.001201 seconds
Process-2038, Step 3: Get HOG channels: 5e-06 seconds
Process-2038, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2038, Step 5: Compute individual channel HOG features for the entire image: 0.331008 seconds
Process-2038, Step 6: Misc initializations: 0.0 seconds
Process-2038, Step 7: for loop: 0.019827 seconds

Process-2038All steps time: 0.361607 seconds

Process-2038, Find cars processing time: 0.361664 seconds

Process-2035, Step 1: Divide with 255, processing time: 0.007412 seconds
Process-2035, Step 2: Resize if scale is not 1: 0.000549 seconds
Process-2035, Step 3: Get HOG channels: 5e-06 seconds
Process-2035, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2035, Step 5: Compute individual channel HOG features for the entire image: 0.208225 seconds
Process-2035, Step 6: Misc initializations: 1e-06 seconds
Process-2035, Step 7: for loop: 0.012814 seconds

Process-2035All steps time: 0.22901 seconds

Process-2035, Find cars processing time: 0.229068 seconds

Process-2037, Step 1: Divide with 255, processing time: 0.009485 seconds

Process-2037, Step 2: Resize if scale is not 1: 0.001024 seconds
Process-2037, Step 3: Get HOG channels: 1.3e-05 seconds
Process-2037, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2037, Step 5: Compute individual channel HOG features for the entire image: 0.245299 seconds
Process-2037, Step 6: Misc initializations: 1e-06 seconds
Process-2037, Step 7: for loop: 0.030266 seconds

Process-2037All steps time: 0.28609799999999996 seconds

Process-2037, Find cars processing time: 0.286188 seconds

Process-2036, Step 1: Divide with 255, processing time: 0.010323 seconds
Process-2036, Step 2: Resize if scale is not 1: 0.000872 seconds
Process-2036, Step 3: Get HOG channels: 8e-06 seconds
Process-2036, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2036, Step 5: Compute individual channel HOG features for the entire image: 0.24618 seconds
Process-2036, Step 6: Misc initializations: 1e-06 seconds
Process-2036, Step 7: for loop: 0.01339 seconds

Process-2036All steps time: 0.270783 seconds

Process-2036, Find cars processing time: 0.27086 seconds

Process-2035, Step 1: Divide with 255, processing time: 0.010314 seconds
Process-2035, Step 2: Resize if scale is not 1: 0.000896 seconds
Process-2035, Step 3: Get HOG channels: 7e-06 seconds
Process-2035, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2035, Step 5: Compute individual channel HOG features for the entire image: 0.177572 seconds
Process-2035, Step 6: Misc initializations: 1e-06 seconds
Process-2035, Step 7: for loop: 0.009692 seconds

Process-2035All steps time: 0.19848900000000003 seconds

Process-2035, Find cars processing time: 0.19856 seconds

Process-2038, Step 1: Divide with 255, processing time: 0.010428 seconds
Process-2038, Step 2: Resize if scale is not 1: 0.000613 seconds
Process-2038, Step 3: Get HOG channels: 4e-06 seconds
Process-2038, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2038, Step 5: Compute individual channel HOG features for the entire image: 0.147218 seconds
Process-2038, Step 6: Misc initializations: 1e-06 seconds
Process-2038, Step 7: for loop: 0.007772 seconds

Process-2038All steps time: 0.166041 seconds

Process-2038, Find cars processing time: 0.166112 seconds

Process-2038, Step 1: Divide with 255, processing time: 0.010397 seconds

Process-2038, Step 2: Resize if scale is not 1: 0.001162 seconds
Process-2038, Step 3: Get HOG channels: 8e-06 seconds
Process-2038, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2038, Step 5: Compute individual channel HOG features for the entire image: 0.09868 seconds
Process-2038, Step 6: Misc initializations: 0.0 seconds
Process-2038, Step 7: for loop: 0.005668 seconds

Process-2038All steps time: 0.11592400000000001 seconds

Process-2038, Find cars processing time: 0.115992 seconds

Process-2037, Step 1: Divide with 255, processing time: 0.010988 seconds
Process-2037, Step 2: Resize if scale is not 1: 0.000846 seconds
Process-2037, Step 3: Get HOG channels: 9e-06 seconds
Process-2037, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2037, Step 5: Compute individual channel HOG features for the entire image: 0.186496 seconds
Process-2037, Step 6: Misc initializations: 1e-06 seconds
Process-2037, Step 7: for loop: 0.010456 seconds

Process-2037All steps time: 0.208804 seconds

Process-2037, Find cars processing time: 0.2089 seconds

Process-2035, Step 1: Divide with 255, processing time: 0.008205 seconds
Process-2035, Step 4: Define blocks and steps as above: 1.4e-05 seconds
Process-2035, Step 2: Resize if scale is not 1: 0.000934 seconds
Process-2035, Step 3: Get HOG channels: 6e-06 seconds
Process-2035, Step 5: Compute individual channel HOG features for the entire image: 0.13249 seconds
Process-2035, Step 6: Misc initializations: 1e-06 seconds
Process-2035, Step 7: for loop: 0.007163 seconds

Process-2035All steps time: 0.148813 seconds

Process-2035, Find cars processing time: 0.148874 seconds

Process-2036, Step 1: Divide with 255, processing time: 0.010572 seconds
Process-2036, Step 2: Resize if scale is not 1: 0.000831 seconds
Process-2036, Step 3: Get HOG channels: 5e-06 seconds
Process-2036, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2036, Step 5: Compute individual channel HOG features for the entire image: 0.134977 seconds
Process-2036, Step 6: Misc initializations: 1e-06 seconds
Process-2036, Step 7: for loop: 0.007735 seconds

Process-2036All steps time: 0.154126 seconds

Process-2036, Find cars processing time: 0.154189 seconds

Process-2038, Step 1: Divide with 255, processing time: 0.019256 seconds

```
Process-2038, Step 2: Resize if scale is not 1: 0.000505 seconds
Process-2038, Step 3: Get HOG channels: 5e-06 seconds
Process-2038, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2038, Step 5: Compute individual channel HOG features for the entire image: 0.124312 seconds
Process-2038, Step 6: Misc initializations: 0.0 seconds
Process-2038, Step 7: for loop: 0.007029 seconds

Process-2038All steps time: 0.151113 seconds

Process-2038, Find cars processing time: 0.151176 seconds

Process-2037, Step 1: Divide with 255, processing time: 0.008684 seconds
Process-2037, Step 2: Resize if scale is not 1: 0.001119 seconds
Process-2037, Step 3: Get HOG channels: 8e-06 seconds
Process-2037, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2037, Step 5: Compute individual channel HOG features for the entire image: 0.154412 seconds
Process-2037, Step 6: Misc initializations: 0.0 seconds
Process-2037, Step 7: for loop: 0.007657 seconds

Process-2037All steps time: 0.17188599999999998 seconds

Process-2035, Step 1: Divide with 255, processing time: 0.006366 seconds
Process-2037, Find cars processing time: 0.171958 seconds

Process-2035, Step 2: Resize if scale is not 1: 0.000555 seconds
Process-2035, Step 3: Get HOG channels: 6e-06 seconds
Process-2035, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2035, Step 5: Compute individual channel HOG features for the entire image: 0.114035 seconds
Process-2035, Step 6: Misc initializations: 0.0 seconds
Process-2035, Step 7: for loop: 0.00612 seconds

Process-2035All steps time: 0.12708699999999998 seconds

Process-2035, Find cars processing time: 0.127162 seconds

The times for each task are: [0.573025, 0.684602, 0.388046, 0.552854, 0.650681, 0.30688, 0.37574
```

Number of processes used: 4 window size 260
Length of task list: 21
Number of processes used: 4

```
Process-2039, Step 1: Divide with 255, processing time: 0.028814 seconds
```

```
Process-2039, Step 2: Resize if scale is not 1: 0.012279 seconds
Process-2039, Step 3: Get HOG channels: 7.7e-05 seconds
```

Process-2039, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-2039, Step 5: Compute individual channel HOG features for the entire image: 0.507643 seconds
Process-2039, Step 6: Misc initializations: 1e-06 seconds
Process-2039, Step 7: for loop: 0.037066 seconds

Process-2039All steps time: 0.585892 seconds

Process-2039, Find cars processing time: 0.586199 seconds
Process-2040, Step 1: Divide with 255, processing time: 0.02912 seconds
Process-2040, Step 2: Resize if scale is not 1: 0.012004 seconds
Process-2040, Step 3: Get HOG channels: 7.5e-05 seconds
Process-2040, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2040, Step 5: Compute individual channel HOG features for the entire image: 0.48435 seconds
Process-2040, Step 6: Misc initializations: 1e-06 seconds
Process-2040, Step 7: for loop: 0.037522 seconds

Process-2040All steps time: 0.5630809999999999 seconds

Process-2040, Find cars processing time: 0.563426 seconds

Process-2041, Step 1: Divide with 255, processing time: 0.02579 seconds
Process-2041, Step 2: Resize if scale is not 1: 0.012199 seconds
Process-2041, Step 3: Get HOG channels: 8.3e-05 seconds
Process-2041, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-2041, Step 5: Compute individual channel HOG features for the entire image: 0.510263 seconds
Process-2041, Step 6: Misc initializations: 1e-06 seconds
Process-2041, Step 7: for loop: 0.036302 seconds

Process-2041All steps time: 0.584649 seconds

Process-2041, Find cars processing time: 0.585006 seconds

Process-2040, Step 1: Divide with 255, processing time: 0.009566 seconds
Process-2040, Step 2: Resize if scale is not 1: 0.000968 seconds
Process-2040, Step 3: Get HOG channels: 7e-06 seconds
Process-2040, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2040, Step 5: Compute individual channel HOG features for the entire image: 0.293399 seconds
Process-2040, Step 6: Misc initializations: 1e-06 seconds
Process-2040, Step 7: for loop: 0.01947 seconds

Process-2040All steps time: 0.3234189999999996 seconds

Process-2040, Find cars processing time: 0.323505 seconds

Process-2042, Step 1: Divide with 255, processing time: 0.028943 seconds
Process-2042, Step 2: Resize if scale is not 1: 0.002471 seconds
Process-2042, Step 3: Get HOG channels: 1.8e-05 seconds
Process-2042, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-2042, Step 5: Compute individual channel HOG features for the entire image: 0.336891 seconds
Process-2042, Step 6: Misc initializations: 1e-06 seconds
Process-2039, Step 1: Divide with 255, processing time: 0.010422 seconds
Process-2042, Step 7: for loop: 0.021 seconds
Process-2039, Step 2: Resize if scale is not 1: 0.002998 seconds

Process-2039, Step 3: Get HOG channels: 7e-06 seconds
Process-2042All steps time: 0.389337 seconds
Process-2039, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2039, Step 5: Compute individual channel HOG features for the entire image: 0.579256 seconds
Process-2042, Find cars processing time: 0.389486 seconds
Process-2039, Step 6: Misc initializations: 1e-06 seconds

Process-2039, Step 7: for loop: 0.034869 seconds

Process-2039All steps time: 0.627561 seconds

Process-2039, Find cars processing time: 0.627639 seconds

Process-2041, Step 1: Divide with 255, processing time: 0.008805 seconds
Process-2041, Step 2: Resize if scale is not 1: 0.000927 seconds
Process-2041, Step 3: Get HOG channels: 7e-06 seconds
Process-2041, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2041, Step 5: Compute individual channel HOG features for the entire image: 0.331931 seconds
Process-2041, Step 6: Misc initializations: 1e-06 seconds
Process-2041, Step 7: for loop: 0.020295 seconds

Process-2041All steps time: 0.3619729999999993 seconds

Process-2041, Find cars processing time: 0.362041 seconds

Process-2040, Step 1: Divide with 255, processing time: 0.01006 seconds
Process-2040, Step 2: Resize if scale is not 1: 0.000937 seconds
Process-2040, Step 3: Get HOG channels: 7e-06 seconds
Process-2040, Step 4: Define blocks and steps as above: 1.7e-05 seconds
Process-2040, Step 5: Compute individual channel HOG features for the entire image: 0.282887 seconds
Process-2040, Step 6: Misc initializations: 1e-06 seconds
Process-2040, Step 7: for loop: 0.018303 seconds

Process-2040All steps time: 0.312212 seconds

Process-2040, Find cars processing time: 0.31227 seconds

Process-2039, Step 1: Divide with 255, processing time: 0.009971 seconds
Process-2039, Step 2: Resize if scale is not 1: 0.001302 seconds
Process-2039, Step 3: Get HOG channels: 6e-06 seconds
Process-2039, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2039, Step 5: Compute individual channel HOG features for the entire image: 0.237886 seconds
Process-2039, Step 6: Misc initializations: 0.0 seconds
Process-2039, Step 7: for loop: 0.01674 seconds

Process-2039All steps time: 0.265912 seconds

Process-2039, Find cars processing time: 0.265975 seconds

Process-2042, Step 1: Divide with 255, processing time: 0.010827 seconds
Process-2042, Step 2: Resize if scale is not 1: 0.001075 seconds
Process-2042, Step 3: Get HOG channels: 1e-05 seconds
Process-2042, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2042, Step 5: Compute individual channel HOG features for the entire image: 0.200393 seconds
Process-2042, Step 6: Misc initializations: 1e-06 seconds
Process-2042, Step 7: for loop: 0.01367 seconds

Process-2042All steps time: 0.225983 seconds

Process-2042, Find cars processing time: 0.226079 seconds

Process-2039, Step 1: Divide with 255, processing time: 0.009923 seconds
Process-2039, Step 2: Resize if scale is not 1: 0.000829 seconds
Process-2039, Step 3: Get HOG channels: 7e-06 seconds
Process-2039, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2041, Step 1: Divide with 255, processing time: 0.008888 seconds
Process-2039, Step 5: Compute individual channel HOG features for the entire image: 0.188669 seconds
Process-2041, Step 2: Resize if scale is not 1: 0.000651 seconds
Process-2039, Step 6: Misc initializations: 0.0 seconds
Process-2041, Step 3: Get HOG channels: 5e-06 seconds
Process-2039, Step 7: for loop: 0.009485 seconds
Process-2041, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2041, Step 5: Compute individual channel HOG features for the entire image: 0.205586 seconds

Process-2039All steps time: 0.20892 seconds

Process-2041, Step 6: Misc initializations: 1e-06 seconds

Process-2041, Step 7: for loop: 0.013421 seconds

Process-2039, Find cars processing time: 0.20893 seconds

Process-2041All steps time: 0.2285579999999998 seconds

Process-2041, Find cars processing time: 0.228625 seconds

Process-2040, Step 1: Divide with 255, processing time: 0.009903 seconds
Process-2040, Step 2: Resize if scale is not 1: 0.00062 seconds
Process-2040, Step 3: Get HOG channels: 6e-06 seconds
Process-2040, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2040, Step 5: Compute individual channel HOG features for the entire image: 0.261292 seconds
Process-2040, Step 6: Misc initializations: 1e-06 seconds
Process-2040, Step 7: for loop: 0.013402 seconds

Process-2040All steps time: 0.28523000000000004 seconds

Process-2040, Find cars processing time: 0.285289 seconds

Process-2041, Step 1: Divide with 255, processing time: 0.010305 seconds
Process-2041, Step 2: Resize if scale is not 1: 0.000932 seconds
Process-2041, Step 3: Get HOG channels: 8e-06 seconds
Process-2041, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2041, Step 5: Compute individual channel HOG features for the entire image: 0.15779 seconds
Process-2041, Step 6: Misc initializations: 0.0 seconds
Process-2041, Step 7: for loop: 0.007968 seconds

Process-2041All steps time: 0.177011 seconds

Process-2041, Find cars processing time: 0.177084 seconds

Process-2039, Step 1: Divide with 255, processing time: 0.009305 seconds
Process-2039, Step 2: Resize if scale is not 1: 0.000525 seconds
Process-2039, Step 3: Get HOG channels: 5e-06 seconds
Process-2039, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2039, Step 5: Compute individual channel HOG features for the entire image: 0.167577 seconds
Process-2039, Step 6: Misc initializations: 1e-06 seconds
Process-2039, Step 7: for loop: 0.009629 seconds

Process-2039All steps time: 0.187048 seconds

Process-2039, Find cars processing time: 0.187108 seconds

Process-2042, Step 1: Divide with 255, processing time: 0.011057 seconds
Process-2042, Step 2: Resize if scale is not 1: 0.000871 seconds
Process-2042, Step 3: Get HOG channels: 8e-06 seconds
Process-2042, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2042, Step 5: Compute individual channel HOG features for the entire image: 0.147165 seconds
Process-2042, Step 6: Misc initializations: 1e-06 seconds
Process-2042, Step 7: for loop: 0.008639 seconds

Process-2042All steps time: 0.167749 seconds

Process-2042, Find cars processing time: 0.16782 seconds

Process-2042, Step 1: Divide with 255, processing time: 0.010103 seconds
Process-2042, Step 2: Resize if scale is not 1: 0.001208 seconds
Process-2042, Step 3: Get HOG channels: 8e-06 seconds
Process-2042, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2042, Step 5: Compute individual channel HOG features for the entire image: 0.101267 seconds
Process-2042, Step 6: Misc initializations: 1e-06 seconds
Process-2042, Step 7: for loop: 0.005992 seconds

Process-2042All steps time: 0.118588 seconds

Process-2042, Find cars processing time: 0.118659 seconds

Process-2040, Step 1: Divide with 255, processing time: 0.010338 seconds
Process-2040, Step 2: Resize if scale is not 1: 0.000948 seconds
Process-2040, Step 3: Get HOG channels: 4e-06 seconds
Process-2040, Step 4: Define blocks and steps as above: 2.1e-05 seconds
Process-2040, Step 5: Compute individual channel HOG features for the entire image: 0.100028 seconds
Process-2040, Step 6: Misc initializations: 0.0 seconds
Process-2040, Step 7: for loop: 0.005635 seconds

Process-2040All steps time: 0.11697400000000001 seconds

Process-2040, Find cars processing time: 0.117027 seconds

Process-2042, Step 1: Divide with 255, processing time: 0.007734 seconds
Process-2042, Step 2: Resize if scale is not 1: 0.000671 seconds
Process-2042, Step 3: Get HOG channels: 6e-06 seconds
Process-2042, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2042, Step 5: Compute individual channel HOG features for the entire image: 0.08788 seconds
Process-2042, Step 6: Misc initializations: 0.0 seconds
Process-2042, Step 7: for loop: 0.004749 seconds

Process-2042All steps time: 0.10104600000000001 seconds

Process-2042, Find cars processing time: 0.101091 seconds

Process-2041, Step 1: Divide with 255, processing time: 0.010117 seconds
Process-2041, Step 2: Resize if scale is not 1: 0.000729 seconds
Process-2041, Step 3: Get HOG channels: 5e-06 seconds
Process-2041, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2041, Step 5: Compute individual channel HOG features for the entire image: 0.141033 seconds
Process-2041, Step 6: Misc initializations: 1e-06 seconds
Process-2041, Step 7: for loop: 0.007411 seconds

Process-2041All steps time: 0.159302 seconds

Process-2041, Find cars processing time: 0.15936 seconds

Process-2039, Step 1: Divide with 255, processing time: 0.010574 seconds
Process-2039, Step 2: Resize if scale is not 1: 0.000767 seconds
Process-2039, Step 3: Get HOG channels: 6e-06 seconds
Process-2039, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2039, Step 5: Compute individual channel HOG features for the entire image: 0.126558 seconds
Process-2039, Step 6: Misc initializations: 1e-06 seconds
Process-2039, Step 7: for loop: 0.007238 seconds

Process-2039All steps time: 0.14515 seconds

Process-2039, Find cars processing time: 0.14523 seconds

The times for each task are: [0.586199, 0.563426, 0.585006, 0.323505, 0.389486, 0.627639, 0.3620

Minimum: 0.101091 Maximum: 0.627639 Average: 0.2923 seconds

Number of processes used: 4 window size 260
Length of task list: 21
Number of processes used: 4

Process-2043, Step 1: Divide with 255, processing time: 0.028435 seconds
Process-2044, Step 1: Divide with 255, processing time: 0.029744 seconds
Process-2043, Step 2: Resize if scale is not 1: 0.013644 seconds
Process-2044, Step 2: Resize if scale is not 1: 0.011458 seconds
Process-2043, Step 3: Get HOG channels: 9.7e-05 seconds
Process-2044, Step 3: Get HOG channels: 9.7e-05 seconds
Process-2044, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-2043, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-2044, Step 5: Compute individual channel HOG features for the entire image: 0.528491 seconds
Process-2043, Step 5: Compute individual channel HOG features for the entire image: 0.567319 seconds
Process-2044, Step 6: Misc initializations: 1e-06 seconds
Process-2043, Step 6: Misc initializations: 1e-06 seconds
Process-2044, Step 7: for loop: 0.037563 seconds
Process-2043, Step 7: for loop: 0.040031 seconds

Process-2044All steps time: 0.607365 seconds

Process-2043All steps time: 0.6495400000000001 seconds
Process-2044, Find cars processing time: 0.607636 seconds

Process-2043, Find cars processing time: 0.649814 seconds

Process-2044, Step 1: Divide with 255, processing time: 0.007092 seconds
Process-2044, Step 2: Resize if scale is not 1: 0.000732 seconds
Process-2044, Step 3: Get HOG channels: 5e-06 seconds
Process-2044, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2044, Step 5: Compute individual channel HOG features for the entire image: 0.294212 seconds
Process-2044, Step 6: Misc initializations: 1e-06 seconds
Process-2044, Step 7: for loop: 0.019516 seconds

Process-2044All steps time: 0.3215639999999999 seconds

Process-2044, Find cars processing time: 0.321628 seconds

Process-2045, Step 1: Divide with 255, processing time: 0.028762 seconds

Process-2045, Step 2: Resize if scale is not 1: 0.010693 seconds

Process-2045, Step 3: Get HOG channels: 5.5e-05 seconds

Process-2045, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2045, Step 5: Compute individual channel HOG features for the entire image: 0.507071 seconds

Process-2045, Step 6: Misc initializations: 1e-06 seconds

Process-2045, Step 7: for loop: 0.037113 seconds

Process-2045All steps time: 0.583704 seconds

Process-2045, Find cars processing time: 0.583958 seconds

Process-2043, Step 1: Divide with 255, processing time: 0.007562 seconds

Process-2043, Step 2: Resize if scale is not 1: 0.000921 seconds

Process-2043, Step 3: Get HOG channels: 6e-06 seconds

Process-2043, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2043, Step 5: Compute individual channel HOG features for the entire image: 0.30392 seconds

Process-2043, Step 6: Misc initializations: 1e-06 seconds

Process-2043, Step 7: for loop: 0.019818 seconds

Process-2043All steps time: 0.332235 seconds

Process-2043, Find cars processing time: 0.332289 seconds

Process-2046, Step 1: Divide with 255, processing time: 0.02656 seconds

Process-2046, Step 2: Resize if scale is not 1: 0.010473 seconds

Process-2046, Step 3: Get HOG channels: 0.00011 seconds

Process-2046, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-2046, Step 5: Compute individual channel HOG features for the entire image: 0.513227 seconds

Process-2046, Step 6: Misc initializations: 1e-06 seconds

Process-2046, Step 7: for loop: 0.031857 seconds

Process-2046All steps time: 0.582241 seconds

Process-2046, Find cars processing time: 0.582489 seconds

Process-2044, Step 1: Divide with 255, processing time: 0.010373 seconds

Process-2044, Step 2: Resize if scale is not 1: 0.00086 seconds

Process-2044, Step 3: Get HOG channels: 8e-06 seconds

Process-2044, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2044, Step 5: Compute individual channel HOG features for the entire image: 0.375371 seconds

Process-2044, Step 6: Misc initializations: 0.0 seconds

Process-2044, Step 7: for loop: 0.019561 seconds

Process-2044All steps time: 0.406181 seconds

Process-2044, Find cars processing time: 0.406255 seconds

Process-2043, Step 1: Divide with 255, processing time: 0.010175 seconds

Process-2043, Step 2: Resize if scale is not 1: 0.000879 seconds

Process-2043, Step 3: Get HOG channels: 8e-06 seconds

Process-2043, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2043, Step 5: Compute individual channel HOG features for the entire image: 0.188088 seconds

Process-2043, Step 6: Misc initializations: 0.0 seconds

Process-2043, Step 7: for loop: 0.013196 seconds

Process-2043All steps time: 0.21235300000000001 seconds

Process-2043, Find cars processing time: 0.212419 seconds

Process-2046, Step 1: Divide with 255, processing time: 0.010567 seconds

Process-2046, Step 2: Resize if scale is not 1: 0.001234 seconds

Process-2046, Step 3: Get HOG channels: 5e-06 seconds

Process-2046, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2046, Step 5: Compute individual channel HOG features for the entire image: 0.196321 seconds

Process-2046, Step 6: Misc initializations: 0.0 seconds

Process-2046, Step 7: for loop: 0.013478 seconds

Process-2046All steps time: 0.22161099999999997 seconds

Process-2046, Find cars processing time: 0.221679 seconds

Process-2043, Step 1: Divide with 255, processing time: 0.009315 seconds

Process-2043, Step 2: Resize if scale is not 1: 0.000471 seconds

Process-2043, Step 3: Get HOG channels: 5e-06 seconds

Process-2043, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2043, Step 5: Compute individual channel HOG features for the entire image: 0.205986 seconds

Process-2043, Step 6: Misc initializations: 1e-06 seconds

Process-2043, Step 7: for loop: 0.0138 seconds

Process-2043All steps time: 0.229582 seconds

Process-2043, Find cars processing time: 0.229641 seconds

Process-2045, Step 1: Divide with 255, processing time: 0.010669 seconds

Process-2045, Step 2: Resize if scale is not 1: 0.000798 seconds

Process-2045, Step 3: Get HOG channels: 7e-06 seconds

Process-2045, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2045, Step 5: Compute individual channel HOG features for the entire image: 0.379511 seconds

Process-2045, Step 6: Misc initializations: 1e-06 seconds

Process-2045, Step 7: for loop: 0.023146 seconds

Process-2045All steps time: 0.414139 seconds

Process-2045, Find cars processing time: 0.414207 seconds

Process-2044, Step 1: Divide with 255, processing time: 0.010865 seconds

Process-2044, Step 2: Resize if scale is not 1: 0.000625 seconds

Process-2044, Step 3: Get HOG channels: 4e-06 seconds

Process-2044, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2044, Step 5: Compute individual channel HOG features for the entire image: 0.208742 seconds

Process-2044, Step 6: Misc initializations: 0.0 seconds

Process-2044, Step 7: for loop: 0.013223 seconds

Process-2044All steps time: 0.23346300000000003 seconds

Process-2044, Find cars processing time: 0.233521 seconds

Process-2043, Step 1: Divide with 255, processing time: 0.009939 seconds

Process-2043, Step 2: Resize if scale is not 1: 0.000695 seconds

Process-2043, Step 3: Get HOG channels: 6e-06 seconds

Process-2043, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2043, Step 5: Compute individual channel HOG features for the entire image: 0.136553 seconds

Process-2043, Step 6: Misc initializations: 1e-06 seconds

Process-2043, Step 7: for loop: 0.007938 seconds

Process-2043All steps time: 0.155138 seconds

Process-2043, Find cars processing time: 0.155197 seconds

Process-2044, Step 1: Divide with 255, processing time: 0.008606 seconds

Process-2044, Step 2: Resize if scale is not 1: 0.000802 seconds

Process-2044, Step 3: Get HOG channels: 7e-06 seconds

Process-2044, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2044, Step 5: Compute individual channel HOG features for the entire image: 0.136723 seconds

Process-2044, Step 6: Misc initializations: 0.0 seconds

Process-2044, Step 7: for loop: 0.008586 seconds

Process-2044All steps time: 0.15473000000000003 seconds

Process-2044, Find cars processing time: 0.154786 seconds

Process-2045, Step 1: Divide with 255, processing time: 0.010424 seconds

Process-2045, Step 2: Resize if scale is not 1: 0.001488 seconds

Process-2045, Step 3: Get HOG channels: 8e-06 seconds

Process-2045, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2045, Step 5: Compute individual channel HOG features for the entire image: 0.171806 seconds

Process-2045, Step 6: Misc initializations: 1e-06 seconds

Process-2045, Step 7: for loop: 0.007647 seconds

Process-2045All steps time: 0.1913809999999997 seconds

Process-2045, Find cars processing time: 0.191449 seconds

Process-2046, Step 1: Divide with 255, processing time: 0.010186 seconds

Process-2046, Step 2: Resize if scale is not 1: 0.000538 seconds

Process-2046, Step 3: Get HOG channels: 5e-06 seconds

Process-2046, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2046, Step 5: Compute individual channel HOG features for the entire image: 0.132076 seconds

Process-2046, Step 6: Misc initializations: 0.0 seconds

Process-2046, Step 7: for loop: 0.007657 seconds

Process-2046All steps time: 0.150467 seconds

Process-2046, Find cars processing time: 0.150537 seconds

Process-2044, Step 1: Divide with 255, processing time: 0.010463 seconds

Process-2044, Step 2: Resize if scale is not 1: 0.001053 seconds

Process-2044, Step 3: Get HOG channels: 5e-06 seconds

Process-2044, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2044, Step 5: Compute individual channel HOG features for the entire image: 0.105849 seconds

Process-2044, Step 6: Misc initializations: 0.0 seconds

Process-2044, Step 7: for loop: 0.006377 seconds

Process-2044All steps time: 0.1237519999999999 seconds

Process-2044, Find cars processing time: 0.123817 seconds

Process-2043, Step 1: Divide with 255, processing time: 0.010609 seconds

Process-2043, Step 2: Resize if scale is not 1: 0.001213 seconds

Process-2043, Step 3: Get HOG channels: 1.7e-05 seconds

Process-2043, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2043, Step 5: Compute individual channel HOG features for the entire image: 0.105596 seconds

Process-2043, Step 6: Misc initializations: 0.0 seconds

Process-2043, Step 7: for loop: 0.005669 seconds

Process-2043All steps time: 0.123112 seconds

Process-2043, Find cars processing time: 0.123176 seconds

Process-2045, Step 1: Divide with 255, processing time: 0.010656 seconds

Process-2045, Step 2: Resize if scale is not 1: 0.001174 seconds

Process-2045, Step 3: Get HOG channels: 5e-06 seconds

Process-2045, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2045, Step 5: Compute individual channel HOG features for the entire image: 0.10039 seconds

Process-2045, Step 6: Misc initializations: 1e-06 seconds

Process-2045, Step 7: for loop: 0.006248 seconds

Process-2045All steps time: 0.11848 seconds

Process-2045, Find cars processing time: 0.118568 seconds

Process-2046, Step 1: Divide with 255, processing time: 0.010688 seconds

Process-2046, Step 2: Resize if scale is not 1: 0.00116 seconds

Process-2046, Step 3: Get HOG channels: 8e-06 seconds

Process-2046, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2046, Step 5: Compute individual channel HOG features for the entire image: 0.13255 seconds

Process-2046, Step 6: Misc initializations: 1e-06 seconds

Process-2046, Step 7: for loop: 0.007251 seconds

Process-2046All steps time: 0.151666 seconds

Process-2046, Find cars processing time: 0.151738 seconds

Process-2044, Step 1: Divide with 255, processing time: 0.009775 seconds

Process-2044, Step 2: Resize if scale is not 1: 0.000578 seconds

Process-2044, Step 3: Get HOG channels: 6e-06 seconds

Process-2044, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2044, Step 5: Compute individual channel HOG features for the entire image: 0.117894 seconds

Process-2044, Step 6: Misc initializations: 1e-06 seconds

Process-2044, Step 7: for loop: 0.006212 seconds

Process-2044All steps time: 0.134472 seconds

Process-2044, Find cars processing time: 0.134543 seconds

The times for each task are: [0.607636, 0.649814, 0.321628, 0.583958, 0.332289, 0.582489, 0.4062

Minimum: 0.118568 Maximum: 0.649814 Average: 0.2904 seconds

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

4 processes used for testing 5 window sizes

Processing times for each image [3.775, 3.8649, 4.2954, 4.2212, 4.0892, 4.1394] with an average

Time elapsed so far... 258.36190000000005

#####

Number of processes used: 5 window size 260

Length of task list: 21

Number of processes used: 5

Process-2047, Step 1: Divide with 255, processing time: 0.02783 seconds

Process-2047, Step 2: Resize if scale is not 1: 0.011583 seconds

Process-2047, Step 3: Get HOG channels: 9.8e-05 seconds
Process-2047, Step 4: Define blocks and steps as above: 1.4e-05 seconds
Process-2047, Step 5: Compute individual channel HOG features for the entire image: 0.537655 seconds
Process-2047, Step 6: Misc initializations: 1e-06 seconds
Process-2047, Step 7: for loop: 0.038504 seconds

Process-2047All steps time: 0.615685 seconds

Process-2047, Find cars processing time: 0.615987 seconds
Process-2048, Step 1: Divide with 255, processing time: 0.029398 seconds
Process-2048, Step 2: Resize if scale is not 1: 0.011182 seconds
Process-2048, Step 3: Get HOG channels: 8.7e-05 seconds
Process-2048, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-2048, Step 5: Compute individual channel HOG features for the entire image: 0.528348 seconds
Process-2048, Step 6: Misc initializations: 1e-06 seconds
Process-2048, Step 7: for loop: 0.038061 seconds

Process-2048All steps time: 0.6070880000000001 seconds

Process-2048, Find cars processing time: 0.607349 seconds

Process-2049, Step 1: Divide with 255, processing time: 0.026063 seconds
Process-2049, Step 2: Resize if scale is not 1: 0.012386 seconds
Process-2049, Step 3: Get HOG channels: 6.9e-05 seconds
Process-2049, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2049, Step 5: Compute individual channel HOG features for the entire image: 0.521041 seconds
Process-2049, Step 6: Misc initializations: 1e-06 seconds
Process-2049, Step 7: for loop: 0.037567 seconds

Process-2049All steps time: 0.597135 seconds

Process-2049, Find cars processing time: 0.597335 seconds

Process-2050, Step 1: Divide with 255, processing time: 0.026515 seconds
Process-2050, Step 2: Resize if scale is not 1: 0.011662 seconds
Process-2050, Step 3: Get HOG channels: 9.6e-05 seconds
Process-2050, Step 4: Define blocks and steps as above: 1.4e-05 seconds
Process-2050, Step 5: Compute individual channel HOG features for the entire image: 0.459503 seconds
Process-2050, Step 6: Misc initializations: 1e-06 seconds
Process-2050, Step 7: for loop: 0.032008 seconds

Process-2050All steps time: 0.529799 seconds

Process-2050, Find cars processing time: 0.530042 seconds

Process-2049, Step 1: Divide with 255, processing time: 0.010256 seconds
Process-2049, Step 2: Resize if scale is not 1: 0.00083 seconds
Process-2049, Step 3: Get HOG channels: 7e-06 seconds

Process-2049, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2049, Step 5: Compute individual channel HOG features for the entire image: 0.368288 seconds

Process-2049, Step 6: Misc initializations: 1e-06 seconds

Process-2049, Step 7: for loop: 0.020706 seconds

Process-2049All steps time: 0.400094 seconds

Process-2049, Find cars processing time: 0.400187 seconds

Process-2050, Step 1: Divide with 255, processing time: 0.011157 seconds

Process-2050, Step 2: Resize if scale is not 1: 0.001058 seconds

Process-2050, Step 3: Get HOG channels: 5e-06 seconds

Process-2050, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2050, Step 5: Compute individual channel HOG features for the entire image: 0.238453 seconds

Process-2050, Step 6: Misc initializations: 1e-06 seconds

Process-2050, Step 7: for loop: 0.01282 seconds

Process-2050All steps time: 0.2634979999999995 seconds

Process-2050, Find cars processing time: 0.263546 seconds

Process-2051, Step 1: Divide with 255, processing time: 0.025661 seconds

Process-2051, Step 2: Resize if scale is not 1: 0.002606 seconds

Process-2051, Step 3: Get HOG channels: 1.9e-05 seconds

Process-2051, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2051, Step 5: Compute individual channel HOG features for the entire image: 0.419159 seconds

Process-2051, Step 6: Misc initializations: 1e-06 seconds

Process-2051, Step 7: for loop: 0.028285 seconds

Process-2051All steps time: 0.475736 seconds

Process-2051, Find cars processing time: 0.475883 seconds

Process-2048, Step 1: Divide with 255, processing time: 0.011184 seconds

Process-2048, Step 2: Resize if scale is not 1: 0.000738 seconds

Process-2048, Step 3: Get HOG channels: 5e-06 seconds

Process-2048, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2048, Step 5: Compute individual channel HOG features for the entire image: 0.344014 seconds

Process-2048, Step 6: Misc initializations: 0.0 seconds

Process-2048, Step 7: for loop: 0.019641 seconds

Process-2048All steps time: 0.3755880000000003 seconds

Process-2048, Find cars processing time: 0.375643 seconds

Process-2047, Step 1: Divide with 255, processing time: 0.010597 seconds

Process-2047, Step 2: Resize if scale is not 1: 0.001043 seconds

Process-2047, Step 3: Get HOG channels: 8e-06 seconds

Process-2047, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2047, Step 5: Compute individual channel HOG features for the entire image: 0.435575 seconds

Process-2047, Step 6: Misc initializations: 1e-06 seconds

Process-2047, Step 7: for loop: 0.01996 seconds

Process-2047All steps time: 0.4671909999999997 seconds

Process-2047, Find cars processing time: 0.467263 seconds

Process-2049, Step 1: Divide with 255, processing time: 0.009289 seconds

Process-2049, Step 2: Resize if scale is not 1: 0.000777 seconds

Process-2049, Step 3: Get HOG channels: 7e-06 seconds

Process-2049, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2049, Step 5: Compute individual channel HOG features for the entire image: 0.18553 seconds

Process-2049, Step 6: Misc initializations: 1e-06 seconds

Process-2049, Step 7: for loop: 0.012653 seconds

Process-2049All steps time: 0.208263 seconds

Process-2049, Find cars processing time: 0.208332 seconds

Process-2050, Step 1: Divide with 255, processing time: 0.00972 seconds

Process-2050, Step 2: Resize if scale is not 1: 0.000873 seconds

Process-2050, Step 3: Get HOG channels: 8e-06 seconds

Process-2050, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2050, Step 5: Compute individual channel HOG features for the entire image: 0.20838 seconds

Process-2050, Step 6: Misc initializations: 0.0 seconds

Process-2050, Step 7: for loop: 0.012674 seconds

Process-2050All steps time: 0.231663 seconds

Process-2050, Find cars processing time: 0.231732 seconds

Process-2047, Step 1: Divide with 255, processing time: 0.005358 seconds

Process-2047, Step 2: Resize if scale is not 1: 0.000979 seconds

Process-2047, Step 3: Get HOG channels: 7e-06 seconds

Process-2047, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2047, Step 5: Compute individual channel HOG features for the entire image: 0.182504 seconds

Process-2047, Step 6: Misc initializations: 1e-06 seconds

Process-2047, Step 7: for loop: 0.008083 seconds

Process-2047All steps time: 0.196938 seconds

Process-2047, Find cars processing time: 0.197004 seconds

Process-2051, Step 1: Divide with 255, processing time: 0.009883 seconds

Process-2051, Step 2: Resize if scale is not 1: 0.000695 seconds

Process-2051, Step 3: Get HOG channels: 7e-06 seconds

Process-2051, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2051, Step 5: Compute individual channel HOG features for the entire image: 0.274484 seconds
Process-2051, Step 6: Misc initializations: 1e-06 seconds
Process-2051, Step 7: for loop: 0.014472 seconds

Process-2051All steps time: 0.299548 seconds

Process-2051, Find cars processing time: 0.299612 seconds

Process-2048, Step 1: Divide with 255, processing time: 0.010857 seconds
Process-2048, Step 2: Resize if scale is not 1: 0.001614 seconds
Process-2048, Step 3: Get HOG channels: 8e-06 seconds
Process-2048, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2048, Step 5: Compute individual channel HOG features for the entire image: 0.208551 seconds
Process-2048, Step 6: Misc initializations: 1e-06 seconds
Process-2048, Step 7: for loop: 0.008096 seconds

Process-2048All steps time: 0.2291339999999998 seconds

Process-2048, Find cars processing time: 0.229203 seconds

Process-2049, Step 1: Divide with 255, processing time: 0.007211 seconds
Process-2049, Step 2: Resize if scale is not 1: 0.000719 seconds
Process-2049, Step 3: Get HOG channels: 7e-06 seconds
Process-2049, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2049, Step 5: Compute individual channel HOG features for the entire image: 0.215858 seconds
Process-2049, Step 6: Misc initializations: 1e-06 seconds
Process-2049, Step 7: for loop: 0.009512 seconds

Process-2049All steps time: 0.233315 seconds

Process-2049, Find cars processing time: 0.233385 seconds

Process-2048, Step 1: Divide with 255, processing time: 0.011272 seconds
Process-2048, Step 2: Resize if scale is not 1: 0.000976 seconds
Process-2048, Step 3: Get HOG channels: 5e-06 seconds
Process-2048, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2048, Step 5: Compute individual channel HOG features for the entire image: 0.103869 seconds
Process-2048, Step 6: Misc initializations: 1e-06 seconds
Process-2048, Step 7: for loop: 0.006598 seconds

Process-2048All steps time: 0.122726 seconds

Process-2048, Find cars processing time: 0.122789 seconds

Process-2049, Step 1: Divide with 255, processing time: 0.010763 seconds
Process-2049, Step 2: Resize if scale is not 1: 0.001114 seconds
Process-2049, Step 3: Get HOG channels: 7e-06 seconds

Process-2049, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2049, Step 5: Compute individual channel HOG features for the entire image: 0.12931 seconds
Process-2049, Step 6: Misc initializations: 0.0 seconds
Process-2049, Step 7: for loop: 0.007197 seconds

Process-2049All steps time: 0.14839700000000003 seconds
Process-2050, Step 1: Divide with 255, processing time: 0.010138 seconds

Process-2050, Step 2: Resize if scale is not 1: 0.000523 seconds
Process-2050, Step 3: Get HOG channels: 5e-06 seconds
Process-2049, Find cars processing time: 0.148454 seconds
Process-2050, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2050, Step 5: Compute individual channel HOG features for the entire image: 0.170187 seconds
Process-2050, Step 6: Misc initializations: 1e-06 seconds
Process-2050, Step 7: for loop: 0.009345 seconds

Process-2050All steps time: 0.1902049999999999 seconds

Process-2050, Find cars processing time: 0.19026 seconds

Process-2051, Step 1: Divide with 255, processing time: 0.006185 seconds
Process-2051, Step 2: Resize if scale is not 1: 0.000765 seconds
Process-2051, Step 3: Get HOG channels: 6e-06 seconds
Process-2051, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2051, Step 5: Compute individual channel HOG features for the entire image: 0.1576 seconds
Process-2051, Step 6: Misc initializations: 1e-06 seconds
Process-2051, Step 7: for loop: 0.008871 seconds

Process-2051All steps time: 0.1734339999999998 seconds

Process-2051, Find cars processing time: 0.173498 seconds

Process-2047, Step 1: Divide with 255, processing time: 0.010362 seconds
Process-2047, Step 2: Resize if scale is not 1: 0.001111 seconds
Process-2047, Step 3: Get HOG channels: 8e-06 seconds
Process-2047, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2047, Step 5: Compute individual channel HOG features for the entire image: 0.132153 seconds
Process-2047, Step 6: Misc initializations: 0.0 seconds
Process-2047, Step 7: for loop: 0.007073 seconds

Process-2047All steps time: 0.150714 seconds

Process-2047, Find cars processing time: 0.150773 seconds

Process-2048, Step 1: Divide with 255, processing time: 0.009854 seconds
Process-2048, Step 2: Resize if scale is not 1: 0.000979 seconds
Process-2048, Step 3: Get HOG channels: 9e-06 seconds

Process-2048, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2048, Step 5: Compute individual channel HOG features for the entire image: 0.096718 seconds

Process-2048, Step 6: Misc initializations: 0.0 seconds

Process-2048, Step 7: for loop: 0.005131 seconds

Process-2048All steps time: 0.112701 seconds

Process-2048, Find cars processing time: 0.112776 seconds

The times for each task are: [0.615987, 0.607349, 0.597335, 0.530042, 0.400187, 0.263546, 0.4758]

Minimum: 0.112776 Maximum: 0.615987 Average: 0.3158 seconds

Number of processes used: 5 window size 260

Length of task list: 21

Number of processes used: 5

Process-2052, Step 1: Divide with 255, processing time: 0.025481 seconds

Process-2052, Step 2: Resize if scale is not 1: 0.011513 seconds

Process-2052, Step 3: Get HOG channels: 0.0001 seconds

Process-2052, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-2052, Step 5: Compute individual channel HOG features for the entire image: 0.507298 seconds

Process-2052, Step 6: Misc initializations: 1e-06 seconds

Process-2052, Step 7: for loop: 0.041083 seconds

Process-2052All steps time: 0.585488 seconds

Process-2052, Find cars processing time: 0.585701 seconds

Process-2053, Step 1: Divide with 255, processing time: 0.028358 seconds

Process-2053, Step 2: Resize if scale is not 1: 0.01056 seconds

Process-2053, Step 3: Get HOG channels: 0.000103 seconds

Process-2053, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2053, Step 5: Compute individual channel HOG features for the entire image: 0.676147 seconds

Process-2053, Step 6: Misc initializations: 1e-06 seconds

Process-2053, Step 7: for loop: 0.040122 seconds

Process-2053All steps time: 0.7553000000000001 seconds

Process-2053, Find cars processing time: 0.755468 seconds

Process-2054, Step 1: Divide with 255, processing time: 0.027448 seconds

Process-2054, Step 2: Resize if scale is not 1: 0.011621 seconds

Process-2054, Step 3: Get HOG channels: 7.2e-05 seconds

Process-2054, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2054, Step 5: Compute individual channel HOG features for the entire image: 0.557547 seconds

Process-2054, Step 6: Misc initializations: 1e-06 seconds

Process-2054, Step 7: for loop: 0.038355 seconds

Process-2054All steps time: 0.6350540000000001 seconds

Process-2054, Find cars processing time: 0.635222 seconds

Process-2052, Step 1: Divide with 255, processing time: 0.007756 seconds

Process-2052, Step 2: Resize if scale is not 1: 0.000908 seconds

Process-2052, Step 3: Get HOG channels: 7e-06 seconds

Process-2052, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2052, Step 5: Compute individual channel HOG features for the entire image: 0.315307 sec

Process-2052, Step 6: Misc initializations: 1e-06 seconds

Process-2052, Step 7: for loop: 0.025511 seconds

Process-2052All steps time: 0.349497 seconds

Process-2052, Find cars processing time: 0.349591 seconds

Process-2055, Step 1: Divide with 255, processing time: 0.030187 seconds

Process-2055, Step 2: Resize if scale is not 1: 0.012421 seconds

Process-2055, Step 3: Get HOG channels: 9e-05 seconds

Process-2055, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-2055, Step 5: Compute individual channel HOG features for the entire image: 0.538357 sec

Process-2055, Step 6: Misc initializations: 1e-06 seconds

Process-2055, Step 7: for loop: 0.032862 seconds

Process-2055All steps time: 0.6139319999999999 seconds

Process-2055, Find cars processing time: 0.61413 seconds

Process-2053, Step 1: Divide with 255, processing time: 0.007701 seconds

Process-2053, Step 2: Resize if scale is not 1: 0.000868 seconds

Process-2053, Step 3: Get HOG channels: 7e-06 seconds

Process-2053, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2053, Step 5: Compute individual channel HOG features for the entire image: 0.347598 sec

Process-2053, Step 6: Misc initializations: 1e-06 seconds

Process-2053, Step 7: for loop: 0.020934 seconds

Process-2053All steps time: 0.377115 seconds

Process-2053, Find cars processing time: 0.377208 seconds

Process-2056, Step 1: Divide with 255, processing time: 0.02704 seconds

Process-2056, Step 2: Resize if scale is not 1: 0.00269 seconds

Process-2056, Step 3: Get HOG channels: 3.6e-05 seconds

Process-2056, Step 4: Define blocks and steps as above: 2e-05 seconds

Process-2056, Step 5: Compute individual channel HOG features for the entire image: 0.387309 sec

Process-2056, Step 6: Misc initializations: 1e-06 seconds

Process-2056, Step 7: for loop: 0.031654 seconds

Process-2056All steps time: 0.44875 seconds

Process-2056, Find cars processing time: 0.448916 seconds

Process-2055, Step 1: Divide with 255, processing time: 0.010274 seconds

Process-2055, Step 2: Resize if scale is not 1: 0.001596 seconds

Process-2055, Step 3: Get HOG channels: 7e-06 seconds

Process-2055, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2055, Step 5: Compute individual channel HOG features for the entire image: 0.227118 seconds

Process-2055, Step 6: Misc initializations: 0.0 seconds

Process-2055, Step 7: for loop: 0.013207 seconds

Process-2055All steps time: 0.252209 seconds

Process-2055, Find cars processing time: 0.25228 seconds

Process-2052, Step 1: Divide with 255, processing time: 0.005888 seconds

Process-2052, Step 2: Resize if scale is not 1: 0.000602 seconds

Process-2052, Step 3: Get HOG channels: 6e-06 seconds

Process-2052, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2052, Step 5: Compute individual channel HOG features for the entire image: 0.232639 seconds

Process-2052, Step 6: Misc initializations: 1e-06 seconds

Process-2052, Step 7: for loop: 0.013072 seconds

Process-2054, Step 1: Divide with 255, processing time: 0.010762 seconds

Process-2054, Step 2: Resize if scale is not 1: 0.001073 seconds

Process-2052All steps time: 0.2522140000000005 seconds

Process-2054, Step 3: Get HOG channels: 8e-06 seconds

Process-2054, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2052, Find cars processing time: 0.252294 seconds

Process-2054, Step 5: Compute individual channel HOG features for the entire image: 0.350486 seconds

Process-2054, Step 6: Misc initializations: 1e-06 seconds

Process-2054, Step 7: for loop: 0.024877 seconds

Process-2054All steps time: 0.3872139999999995 seconds

Process-2054, Find cars processing time: 0.387295 seconds

Process-2053, Step 1: Divide with 255, processing time: 0.008306 seconds

Process-2053, Step 2: Resize if scale is not 1: 0.00081 seconds

Process-2053, Step 3: Get HOG channels: 8e-06 seconds

Process-2053, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2053, Step 5: Compute individual channel HOG features for the entire image: 0.237231 seconds

Process-2053, Step 6: Misc initializations: 2e-06 seconds

Process-2053, Step 7: for loop: 0.015952 seconds

Process-2053All steps time: 0.262317 seconds

Process-2053, Find cars processing time: 0.262417 seconds

Process-2056, Step 1: Divide with 255, processing time: 0.011134 seconds

Process-2053, Step 1: Divide with 255, processing time: 0.010823 seconds

Process-2053, Step 2: Resize if scale is not 1: 0.000769 seconds

Process-2056, Step 2: Resize if scale is not 1: 0.001032 seconds

Process-2053, Step 3: Get HOG channels: 9e-06 seconds

Process-2056, Step 3: Get HOG channels: 2e-05 seconds

Process-2053, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2056, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-2053, Step 5: Compute individual channel HOG features for the entire image: 0.16047 seconds

Process-2056, Step 5: Compute individual channel HOG features for the entire image: 0.246624 seconds

Process-2053, Step 6: Misc initializations: 0.0 seconds

Process-2056, Step 6: Misc initializations: 1e-06 seconds

Process-2053, Step 7: for loop: 0.008375 seconds

Process-2056, Step 7: for loop: 0.015485 seconds

Process-2053All steps time: 0.180452 seconds

Process-2056All steps time: 0.274308 seconds

Process-2053, Find cars processing time: 0.180533 seconds

Process-2056, Find cars processing time: 0.274423 seconds

Process-2052, Step 1: Divide with 255, processing time: 0.01087 seconds

Process-2052, Step 2: Resize if scale is not 1: 0.000874 seconds

Process-2052, Step 3: Get HOG channels: 8e-06 seconds

Process-2052, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2052, Step 5: Compute individual channel HOG features for the entire image: 0.172059 seconds

Process-2052, Step 6: Misc initializations: 1e-06 seconds

Process-2052, Step 7: for loop: 0.010509 seconds

Process-2052All steps time: 0.1943279999999997 seconds

Process-2052, Find cars processing time: 0.194393 seconds

Process-2054, Step 1: Divide with 255, processing time: 0.01307 seconds

Process-2054, Step 2: Resize if scale is not 1: 0.000906 seconds

Process-2054, Step 3: Get HOG channels: 5e-06 seconds

Process-2054, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2054, Step 5: Compute individual channel HOG features for the entire image: 0.144176 seconds

Process-2054, Step 6: Misc initializations: 1e-06 seconds

Process-2054, Step 7: for loop: 0.009072 seconds

Process-2054All steps time: 0.167234 seconds

Process-2054, Find cars processing time: 0.167296 seconds

Process-2055, Step 1: Divide with 255, processing time: 0.008659 seconds

Process-2055, Step 2: Resize if scale is not 1: 0.000729 seconds

Process-2055, Step 3: Get HOG channels: 9e-06 seconds

Process-2055, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2055, Step 5: Compute individual channel HOG features for the entire image: 0.187368 seconds

Process-2055, Step 6: Misc initializations: 1e-06 seconds

Process-2055, Step 7: for loop: 0.008869 seconds

Process-2055All steps time: 0.205642 seconds

Process-2055, Find cars processing time: 0.205802 seconds

Process-2053, Step 1: Divide with 255, processing time: 0.008023 seconds

Process-2053, Step 2: Resize if scale is not 1: 0.000937 seconds

Process-2053, Step 3: Get HOG channels: 5e-06 seconds

Process-2053, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2053, Step 5: Compute individual channel HOG features for the entire image: 0.130095 seconds

Process-2053, Step 6: Misc initializations: 1e-06 seconds

Process-2053, Step 7: for loop: 0.006686 seconds

Process-2053All steps time: 0.145751 seconds

Process-2053, Find cars processing time: 0.145809 seconds

Process-2054, Step 1: Divide with 255, processing time: 0.010631 seconds

Process-2054, Step 2: Resize if scale is not 1: 0.001157 seconds

Process-2054, Step 3: Get HOG channels: 8e-06 seconds

Process-2054, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2054, Step 5: Compute individual channel HOG features for the entire image: 0.106796 seconds

Process-2054, Step 6: Misc initializations: 1e-06 seconds

Process-2054, Step 7: for loop: 0.006101 seconds

Process-2054All steps time: 0.1247009999999999 seconds

Process-2054, Find cars processing time: 0.124761 seconds

Process-2055, Step 1: Divide with 255, processing time: 0.011451 seconds

Process-2055, Step 2: Resize if scale is not 1: 0.001084 seconds

Process-2055, Step 3: Get HOG channels: 5e-06 seconds

Process-2055, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2055, Step 5: Compute individual channel HOG features for the entire image: 0.095732 seconds

Process-2055, Step 6: Misc initializations: 1e-06 seconds

Process-2055, Step 7: for loop: 0.005796 seconds

Process-2055All steps time: 0.114074 seconds

Process-2055, Find cars processing time: 0.114153 seconds

Process-2052, Step 1: Divide with 255, processing time: 0.012927 seconds

Process-2052, Step 2: Resize if scale is not 1: 0.000917 seconds

Process-2052, Step 3: Get HOG channels: 5e-06 seconds

Process-2052, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2052, Step 5: Compute individual channel HOG features for the entire image: 0.133475 seconds

Process-2052, Step 6: Misc initializations: 1e-06 seconds

Process-2052, Step 7: for loop: 0.006909 seconds

Process-2052All steps time: 0.15423900000000001 seconds

Process-2052, Find cars processing time: 0.154299 seconds

Process-2056, Step 1: Divide with 255, processing time: 0.007189 seconds

Process-2056, Step 2: Resize if scale is not 1: 0.000806 seconds

Process-2056, Step 3: Get HOG channels: 5e-06 seconds

Process-2056, Step 4: Define blocks and steps as above: 1.7e-05 seconds

Process-2056, Step 5: Compute individual channel HOG features for the entire image: 0.1104 seconds

Process-2056, Step 6: Misc initializations: 0.0 seconds

Process-2056, Step 7: for loop: 0.005645 seconds

Process-2056All steps time: 0.1240619999999999 seconds

Process-2056, Find cars processing time: 0.12411 seconds

The times for each task are: [0.585701, 0.755468, 0.635222, 0.349591, 0.61413, 0.377208, 0.44891

Minimum: 0.114153 Maximum: 0.755468 Average: 0.3146 seconds

Number of processes used: 5 window size 260

Length of task list: 21

Number of processes used: 5

Process-2057, Step 1: Divide with 255, processing time: 0.036585 seconds

Process-2057, Step 2: Resize if scale is not 1: 0.012716 seconds

Process-2057, Step 3: Get HOG channels: 0.000123 seconds

Process-2057, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-2057, Step 5: Compute individual channel HOG features for the entire image: 0.645504 seconds

Process-2058, Step 1: Divide with 255, processing time: 0.057282 seconds

Process-2057, Step 6: Misc initializations: 1e-06 seconds

Process-2058, Step 2: Resize if scale is not 1: 0.01176 seconds

Process-2057, Step 7: for loop: 0.053641 seconds

Process-2058, Step 3: Get HOG channels: 7.5e-05 seconds

Process-2058, Step 4: Define blocks and steps as above: 1.2e-05 seconds

Process-2057All steps time: 0.748584 seconds

Process-2058, Step 5: Compute individual channel HOG features for the entire image: 0.52288 seconds

Process-2058, Step 6: Misc initializations: 1e-06 seconds

Process-2057, Find cars processing time: 0.748832 seconds

Process-2058, Step 7: for loop: 0.038463 seconds

Process-2058All steps time: 0.6304730000000001 seconds

Process-2058, Find cars processing time: 0.630709 seconds

Process-2059, Step 1: Divide with 255, processing time: 0.026557 seconds

Process-2059, Step 2: Resize if scale is not 1: 0.01319 seconds

Process-2059, Step 3: Get HOG channels: 9.1e-05 seconds

Process-2059, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2059, Step 5: Compute individual channel HOG features for the entire image: 0.515592 seconds

Process-2059, Step 6: Misc initializations: 1e-06 seconds

Process-2059, Step 7: for loop: 0.036142 seconds

Process-2059All steps time: 0.591582 seconds

Process-2059, Find cars processing time: 0.591751 seconds

Process-2060, Step 1: Divide with 255, processing time: 0.028539 seconds

Process-2060, Step 2: Resize if scale is not 1: 0.010938 seconds

Process-2060, Step 3: Get HOG channels: 8.6e-05 seconds

Process-2060, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2060, Step 5: Compute individual channel HOG features for the entire image: 0.608264 seconds

Process-2060, Step 6: Misc initializations: 1e-06 seconds

Process-2060, Step 7: for loop: 0.034841 seconds

Process-2060All steps time: 0.682679 seconds

Process-2060, Find cars processing time: 0.682879 seconds

Process-2061, Step 1: Divide with 255, processing time: 0.026971 seconds

Process-2061, Step 2: Resize if scale is not 1: 0.003179 seconds

Process-2061, Step 3: Get HOG channels: 4.7e-05 seconds

Process-2061, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2061, Step 5: Compute individual channel HOG features for the entire image: 0.393833 seconds

Process-2061, Step 6: Misc initializations: 1e-06 seconds

Process-2061, Step 7: for loop: 0.027387 seconds

Process-2061All steps time: 0.4514239999999994 seconds

Process-2061, Find cars processing time: 0.451638 seconds

Process-2057, Step 1: Divide with 255, processing time: 0.012722 seconds

Process-2057, Step 2: Resize if scale is not 1: 0.001063 seconds

Process-2057, Step 3: Get HOG channels: 9e-06 seconds

Process-2057, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2057, Step 5: Compute individual channel HOG features for the entire image: 0.410465 seconds

Process-2057, Step 6: Misc initializations: 1e-06 seconds

Process-2057, Step 7: for loop: 0.021078 seconds

Process-2057All steps time: 0.445345 seconds

Process-2057, Find cars processing time: 0.445427 seconds

Process-2058, Step 1: Divide with 255, processing time: 0.010452 seconds

Process-2058, Step 2: Resize if scale is not 1: 0.001016 seconds

Process-2058, Step 3: Get HOG channels: 5e-06 seconds

Process-2058, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2058, Step 5: Compute individual channel HOG features for the entire image: 0.330425 seconds

Process-2058, Step 6: Misc initializations: 0.0 seconds

Process-2058, Step 7: for loop: 0.018791 seconds

Process-2058All steps time: 0.360694 seconds

Process-2058, Find cars processing time: 0.360756 seconds

Process-2059, Step 1: Divide with 255, processing time: 0.010154 seconds

Process-2059, Step 2: Resize if scale is not 1: 0.000567 seconds

Process-2059, Step 3: Get HOG channels: 5e-06 seconds

Process-2059, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2059, Step 5: Compute individual channel HOG features for the entire image: 0.288525 seconds

Process-2059, Step 6: Misc initializations: 1e-06 seconds

Process-2059, Step 7: for loop: 0.021108 seconds

Process-2059All steps time: 0.3203649999999996 seconds

Process-2059, Find cars processing time: 0.320452 seconds

Process-2060, Step 1: Divide with 255, processing time: 0.011232 seconds

Process-2060, Step 2: Resize if scale is not 1: 0.001811 seconds

Process-2060, Step 3: Get HOG channels: 8e-06 seconds

Process-2060, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2060, Step 5: Compute individual channel HOG features for the entire image: 0.208143 seconds

Process-2060, Step 6: Misc initializations: 1e-06 seconds

Process-2060, Step 7: for loop: 0.013362 seconds

Process-2060All steps time: 0.234564 seconds

Process-2060, Find cars processing time: 0.23463 seconds

Process-2058, Step 1: Divide with 255, processing time: 0.008765 seconds

Process-2058, Step 2: Resize if scale is not 1: 0.000741 seconds

Process-2058, Step 3: Get HOG channels: 7e-06 seconds

Process-2058, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2058, Step 5: Compute individual channel HOG features for the entire image: 0.217475 seconds

Process-2058, Step 6: Misc initializations: 1e-06 seconds

Process-2058, Step 7: for loop: 0.012884 seconds

Process-2058All steps time: 0.239879 seconds

Process-2058, Find cars processing time: 0.239935 seconds

Process-2061, Step 1: Divide with 255, processing time: 0.00996 seconds

Process-2061, Step 2: Resize if scale is not 1: 0.000698 seconds

Process-2061, Step 3: Get HOG channels: 7e-06 seconds

Process-2061, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2061, Step 5: Compute individual channel HOG features for the entire image: 0.248916 seconds

Process-2061, Step 6: Misc initializations: 1e-06 seconds

Process-2061, Step 7: for loop: 0.013812 seconds

Process-2061All steps time: 0.2734 seconds

Process-2061, Find cars processing time: 0.273456 seconds

Process-2060, Step 1: Divide with 255, processing time: 0.010605 seconds

Process-2060, Step 2: Resize if scale is not 1: 0.000838 seconds

Process-2060, Step 3: Get HOG channels: 7e-06 seconds

Process-2060, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2060, Step 5: Compute individual channel HOG features for the entire image: 0.179823 seconds

Process-2060, Step 6: Misc initializations: 1e-06 seconds

Process-2060, Step 7: for loop: 0.009533 seconds

Process-2060All steps time: 0.2008130000000002 seconds

Process-2060, Find cars processing time: 0.20087 seconds

Process-2061, Step 1: Divide with 255, processing time: 0.006529 seconds

Process-2061, Step 2: Resize if scale is not 1: 0.000585 seconds

Process-2061, Step 3: Get HOG channels: 5e-06 seconds

Process-2061, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2061, Step 5: Compute individual channel HOG features for the entire image: 0.137942 seconds

Process-2061, Step 6: Misc initializations: 1e-06 seconds

Process-2061, Step 7: for loop: 0.008056 seconds

Process-2061All steps time: 0.153123 seconds

Process-2061, Find cars processing time: 0.153175 seconds

Process-2057, Step 1: Divide with 255, processing time: 0.010013 seconds

Process-2057, Step 2: Resize if scale is not 1: 0.000757 seconds

Process-2057, Step 3: Get HOG channels: 7e-06 seconds

Process-2057, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2057, Step 5: Compute individual channel HOG features for the entire image: 0.256751 seconds

Process-2057, Step 6: Misc initializations: 1e-06 seconds

Process-2057, Step 7: for loop: 0.016293 seconds

Process-2057All steps time: 0.283829 seconds

Process-2057, Find cars processing time: 0.283896 seconds

Process-2059, Step 1: Divide with 255, processing time: 0.006982 seconds

Process-2059, Step 2: Resize if scale is not 1: 0.001049 seconds

Process-2059, Step 3: Get HOG channels: 5e-06 seconds

Process-2059, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2059, Step 5: Compute individual channel HOG features for the entire image: 0.150381 seconds

Process-2059, Step 6: Misc initializations: 1e-06 seconds

Process-2059, Step 7: for loop: 0.008384 seconds

Process-2059All steps time: 0.1668069999999998 seconds

Process-2059, Find cars processing time: 0.166862 seconds

Process-2061, Step 1: Divide with 255, processing time: 0.006406 seconds

Process-2061, Step 2: Resize if scale is not 1: 0.000648 seconds

Process-2061, Step 3: Get HOG channels: 5e-06 seconds

Process-2061, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2061, Step 5: Compute individual channel HOG features for the entire image: 0.140895 seconds

Process-2061, Step 6: Misc initializations: 1e-06 seconds

Process-2061, Step 7: for loop: 0.007625 seconds

Process-2061All steps time: 0.1555849999999997 seconds

Process-2061, Find cars processing time: 0.155636 seconds

Process-2058, Step 1: Divide with 255, processing time: 0.008311 seconds

Process-2058, Step 2: Resize if scale is not 1: 0.000685 seconds

Process-2058, Step 3: Get HOG channels: 7e-06 seconds

Process-2058, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2058, Step 5: Compute individual channel HOG features for the entire image: 0.135088 seconds

Process-2058, Step 6: Misc initializations: 0.0 seconds

Process-2058, Step 7: for loop: 0.007559 seconds

Process-2061, Step 1: Divide with 255, processing time: 0.010153 seconds

Process-2058All steps time: 0.151656 seconds
Process-2061, Step 2: Resize if scale is not 1: 0.000875 seconds
Process-2061, Step 3: Get HOG channels: 6e-06 seconds

Process-2061, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2058, Find cars processing time: 0.151715 seconds
Process-2061, Step 5: Compute individual channel HOG features for the entire image: 0.128493 seconds
Process-2061, Step 6: Misc initializations: 0.0 seconds

Process-2061, Step 7: for loop: 0.006345 seconds

Process-2061All steps time: 0.1458769999999998 seconds

Process-2061, Find cars processing time: 0.145948 seconds

Process-2057, Step 1: Divide with 255, processing time: 0.009262 seconds
Process-2057, Step 2: Resize if scale is not 1: 0.001201 seconds
Process-2057, Step 3: Get HOG channels: 7e-06 seconds
Process-2057, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2057, Step 5: Compute individual channel HOG features for the entire image: 0.139506 seconds
Process-2057, Step 6: Misc initializations: 1e-06 seconds
Process-2057, Step 7: for loop: 0.007454 seconds

Process-2057All steps time: 0.1574379999999997 seconds

Process-2057, Find cars processing time: 0.157512 seconds

Process-2060, Step 1: Divide with 255, processing time: 0.009434 seconds
Process-2060, Step 2: Resize if scale is not 1: 0.000997 seconds
Process-2060, Step 3: Get HOG channels: 6e-06 seconds
Process-2060, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2060, Step 5: Compute individual channel HOG features for the entire image: 0.138023 seconds
Process-2060, Step 6: Misc initializations: 1e-06 seconds
Process-2060, Step 7: for loop: 0.007242 seconds

Process-2060All steps time: 0.1557090000000001 seconds

Process-2060, Find cars processing time: 0.155768 seconds

Process-2059, Step 1: Divide with 255, processing time: 0.010656 seconds
Process-2059, Step 2: Resize if scale is not 1: 0.000849 seconds
Process-2059, Step 3: Get HOG channels: 5e-06 seconds
Process-2059, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2059, Step 5: Compute individual channel HOG features for the entire image: 0.134955 seconds
Process-2059, Step 6: Misc initializations: 1e-06 seconds
Process-2059, Step 7: for loop: 0.007357 seconds

Process-2059All steps time: 0.153829 seconds

Process-2059, Find cars processing time: 0.153904 seconds

The times for each task are: [0.748832, 0.630709, 0.591751, 0.682879, 0.451638, 0.445427, 0.3607

Minimum: 0.145948 Maximum: 0.748832 Average: 0.3193 seconds

Number of processes used: 5 window size 260

Length of task list: 21

Number of processes used: 5

Process-2062, Step 1: Divide with 255, processing time: 0.026014 seconds

Process-2062, Step 2: Resize if scale is not 1: 0.010817 seconds

Process-2062, Step 3: Get HOG channels: 7.5e-05 seconds

Process-2062, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2062, Step 5: Compute individual channel HOG features for the entire image: 0.508956 sec

Process-2062, Step 6: Misc initializations: 1e-06 seconds

Process-2062, Step 7: for loop: 0.037019 seconds

Process-2062All steps time: 0.582892 seconds

Process-2063, Step 1: Divide with 255, processing time: 0.024355 seconds

Process-2063, Step 2: Resize if scale is not 1: 0.01203 seconds

Process-2062, Find cars processing time: 0.583127 seconds

Process-2063, Step 3: Get HOG channels: 7.7e-05 seconds

Process-2063, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2063, Step 5: Compute individual channel HOG features for the entire image: 0.558164 sec

Process-2063, Step 6: Misc initializations: 1e-06 seconds

Process-2063, Step 7: for loop: 0.038848 seconds

Process-2063All steps time: 0.633484 seconds

Process-2063, Find cars processing time: 0.633729 seconds

Process-2065, Step 1: Divide with 255, processing time: 0.029957 seconds

Process-2065, Step 2: Resize if scale is not 1: 0.017563 seconds

Process-2065, Step 3: Get HOG channels: 5.5e-05 seconds

Process-2065, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-2065, Step 5: Compute individual channel HOG features for the entire image: 0.483725 sec

Process-2065, Step 6: Misc initializations: 1e-06 seconds

Process-2065, Step 7: for loop: 0.032034 seconds

Process-2065All steps time: 0.5633490000000001 seconds

Process-2065, Find cars processing time: 0.563606 seconds

Process-2064, Step 1: Divide with 255, processing time: 0.029573 seconds
Process-2064, Step 2: Resize if scale is not 1: 0.014576 seconds
Process-2064, Step 3: Get HOG channels: 7.9e-05 seconds
Process-2064, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-2064, Step 5: Compute individual channel HOG features for the entire image: 0.5296 seconds
Process-2064, Step 6: Misc initializations: 1e-06 seconds
Process-2064, Step 7: for loop: 0.037188 seconds

Process-2064All steps time: 0.6110289999999999 seconds

Process-2064, Find cars processing time: 0.611304 seconds

Process-2064, Step 1: Divide with 255, processing time: 0.008346 seconds
Process-2064, Step 2: Resize if scale is not 1: 0.001323 seconds
Process-2064, Step 3: Get HOG channels: 7e-06 seconds
Process-2064, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2064, Step 5: Compute individual channel HOG features for the entire image: 0.232793 seconds
Process-2064, Step 6: Misc initializations: 1e-06 seconds
Process-2064, Step 7: for loop: 0.016401 seconds

Process-2064All steps time: 0.258878 seconds

Process-2064, Find cars processing time: 0.258981 seconds

Process-2066, Step 1: Divide with 255, processing time: 0.029289 seconds
Process-2066, Step 2: Resize if scale is not 1: 0.002751 seconds
Process-2066, Step 3: Get HOG channels: 2.2e-05 seconds
Process-2066, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2066, Step 5: Compute individual channel HOG features for the entire image: 0.363171 seconds
Process-2066, Step 6: Misc initializations: 1e-06 seconds
Process-2066, Step 7: for loop: 0.027145 seconds

Process-2066All steps time: 0.4223849999999995 seconds

Process-2066, Find cars processing time: 0.422566 seconds

Process-2065, Step 1: Divide with 255, processing time: 0.006868 seconds
Process-2065, Step 2: Resize if scale is not 1: 0.001192 seconds
Process-2065, Step 3: Get HOG channels: 9e-06 seconds
Process-2065, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2065, Step 5: Compute individual channel HOG features for the entire image: 0.372145 seconds
Process-2065, Step 6: Misc initializations: 1e-06 seconds
Process-2065, Step 7: for loop: 0.024537 seconds

Process-2065All steps time: 0.4047599999999995 seconds

Process-2065, Find cars processing time: 0.40485 seconds

Process-2062, Step 1: Divide with 255, processing time: 0.006575 seconds
Process-2062, Step 2: Resize if scale is not 1: 0.000665 seconds
Process-2062, Step 3: Get HOG channels: 5e-06 seconds
Process-2062, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2062, Step 5: Compute individual channel HOG features for the entire image: 0.339213 seconds
Process-2062, Step 6: Misc initializations: 1e-06 seconds
Process-2062, Step 7: for loop: 0.019146 seconds

Process-2062All steps time: 0.3656099999999994 seconds

Process-2062, Find cars processing time: 0.365664 seconds

Process-2063, Step 1: Divide with 255, processing time: 0.010601 seconds
Process-2063, Step 2: Resize if scale is not 1: 0.000883 seconds
Process-2063, Step 3: Get HOG channels: 7e-06 seconds
Process-2063, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2063, Step 5: Compute individual channel HOG features for the entire image: 0.386847 seconds
Process-2063, Step 6: Misc initializations: 1e-06 seconds
Process-2063, Step 7: for loop: 0.023919 seconds

Process-2063All steps time: 0.422265 seconds

Process-2063, Find cars processing time: 0.422334 seconds

Process-2065, Step 1: Divide with 255, processing time: 0.009949 seconds
Process-2065, Step 2: Resize if scale is not 1: 0.000832 seconds
Process-2065, Step 3: Get HOG channels: 7e-06 seconds
Process-2065, Step 4: Define blocks and steps as above: 1.7e-05 seconds
Process-2065, Step 5: Compute individual channel HOG features for the entire image: 0.205517 seconds
Process-2065, Step 6: Misc initializations: 0.0 seconds
Process-2065, Step 7: for loop: 0.012489 seconds

Process-2065All steps time: 0.2288110000000001 seconds

Process-2065, Find cars processing time: 0.228878 seconds

Process-2066, Step 1: Divide with 255, processing time: 0.010794 seconds
Process-2066, Step 2: Resize if scale is not 1: 0.0009 seconds
Process-2066, Step 3: Get HOG channels: 9e-06 seconds
Process-2066, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2066, Step 5: Compute individual channel HOG features for the entire image: 0.27691 seconds
Process-2066, Step 6: Misc initializations: 1e-06 seconds
Process-2066, Step 7: for loop: 0.01608 seconds

Process-2066All steps time: 0.304702 seconds

Process-2066, Find cars processing time: 0.304789 seconds

Process-2064, Step 1: Divide with 255, processing time: 0.010267 seconds
Process-2064, Step 2: Resize if scale is not 1: 0.00071 seconds
Process-2064, Step 3: Get HOG channels: 7e-06 seconds
Process-2064, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2064, Step 5: Compute individual channel HOG features for the entire image: 0.196577 seconds
Process-2064, Step 6: Misc initializations: 1e-06 seconds
Process-2064, Step 7: for loop: 0.028506 seconds

Process-2064All steps time: 0.236074 seconds

Process-2064, Find cars processing time: 0.236143 seconds

Process-2063, Step 1: Divide with 255, processing time: 0.008353 seconds
Process-2063, Step 2: Resize if scale is not 1: 0.001225 seconds
Process-2063, Step 3: Get HOG channels: 6e-06 seconds
Process-2063, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2063, Step 5: Compute individual channel HOG features for the entire image: 0.16961 seconds
Process-2063, Step 6: Misc initializations: 1e-06 seconds
Process-2063, Step 7: for loop: 0.009503 seconds

Process-2063All steps time: 0.1887040000000004 seconds

Process-2063, Find cars processing time: 0.188761 seconds

Process-2062, Step 1: Divide with 255, processing time: 0.010429 seconds
Process-2062, Step 2: Resize if scale is not 1: 0.000842 seconds
Process-2062, Step 3: Get HOG channels: 5e-06 seconds
Process-2062, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2062, Step 5: Compute individual channel HOG features for the entire image: 0.141761 seconds
Process-2062, Step 6: Misc initializations: 1e-06 seconds
Process-2062, Step 7: for loop: 0.008129 seconds

Process-2062All steps time: 0.161171 seconds

Process-2062, Find cars processing time: 0.161224 seconds

Process-2065, Step 1: Divide with 255, processing time: 0.009337 seconds
Process-2065, Step 2: Resize if scale is not 1: 0.000642 seconds
Process-2065, Step 3: Get HOG channels: 6e-06 seconds
Process-2065, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2065, Step 5: Compute individual channel HOG features for the entire image: 0.140496 seconds
Process-2065, Step 6: Misc initializations: 1e-06 seconds
Process-2065, Step 7: for loop: 0.008341 seconds

Process-2065All steps time: 0.158829 seconds

Process-2065, Find cars processing time: 0.158879 seconds

Process-2063, Step 1: Divide with 255, processing time: 0.009182 seconds
Process-2063, Step 2: Resize if scale is not 1: 0.00096 seconds
Process-2063, Step 3: Get HOG channels: 6e-06 seconds
Process-2063, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2063, Step 5: Compute individual channel HOG features for the entire image: 0.11536 seconds
Process-2063, Step 6: Misc initializations: 1e-06 seconds
Process-2063, Step 7: for loop: 0.005936 seconds

Process-2063All steps time: 0.131451 seconds

Process-2063, Find cars processing time: 0.131508 seconds

Process-2062, Step 1: Divide with 255, processing time: 0.008051 seconds
Process-2062, Step 2: Resize if scale is not 1: 0.001052 seconds
Process-2062, Step 3: Get HOG channels: 6e-06 seconds

Process-2062, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2062, Step 5: Compute individual channel HOG features for the entire image: 0.107692 seconds
Process-2062, Step 6: Misc initializations: 0.0 seconds
Process-2062, Step 7: for loop: 0.006048 seconds

Process-2062All steps time: 0.12285499999999999 seconds

Process-2062, Find cars processing time: 0.122911 seconds

Process-2064, Step 1: Divide with 255, processing time: 0.010161 seconds
Process-2064, Step 2: Resize if scale is not 1: 0.000738 seconds
Process-2064, Step 3: Get HOG channels: 5e-06 seconds
Process-2064, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2064, Step 5: Compute individual channel HOG features for the entire image: 0.105968 seconds
Process-2064, Step 6: Misc initializations: 0.0 seconds
Process-2064, Step 7: for loop: 0.006006 seconds

Process-2064All steps time: 0.122883 seconds

Process-2064, Find cars processing time: 0.122939 seconds

Process-2066, Step 1: Divide with 255, processing time: 0.009846 seconds
Process-2066, Step 2: Resize if scale is not 1: 0.000744 seconds
Process-2066, Step 3: Get HOG channels: 1e-05 seconds
Process-2066, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2066, Step 5: Compute individual channel HOG features for the entire image: 0.191432 seconds
Process-2066, Step 6: Misc initializations: 0.0 seconds
Process-2066, Step 7: for loop: 0.009123 seconds

Process-2066All steps time: 0.211163 seconds

Process-2066, Find cars processing time: 0.21125 seconds

```
Process-2063, Step 1: Divide with 255, processing time: 0.010821 seconds
Process-2063, Step 2: Resize if scale is not 1: 0.000942 seconds
Process-2063, Step 3: Get HOG channels: 7e-06 seconds
Process-2063, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2063, Step 5: Compute individual channel HOG features for the entire image: 0.097318 seconds
Process-2063, Step 6: Misc initializations: 0.0 seconds
Process-2063, Step 7: for loop: 0.005 seconds
```

Process-2063All steps time: 0.114095 seconds

Process-2063, Find cars processing time: 0.114158 seconds

```
Process-2065, Step 1: Divide with 255, processing time: 0.011427 seconds
Process-2065, Step 2: Resize if scale is not 1: 0.000658 seconds
Process-2065, Step 3: Get HOG channels: 5e-06 seconds
Process-2065, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2065, Step 5: Compute individual channel HOG features for the entire image: 0.109059 seconds
Process-2065, Step 6: Misc initializations: 0.0 seconds
Process-2065, Step 7: for loop: 0.006004 seconds
```

Process-2065All steps time: 0.12715700000000002 seconds

Process-2065, Find cars processing time: 0.127203 seconds

The times for each task are: [0.583127, 0.633729, 0.563606, 0.611304, 0.258981, 0.422566, 0.4048

Minimum: 0.114158 Maximum: 0.633729 Average: 0.3036 seconds

```
*****
Number of processes used: 5 window size 260
Length of task list: 21
Number of processes used: 5
```

```
Process-2068, Step 1: Divide with 255, processing time: 0.026529 seconds
Process-2067, Step 1: Divide with 255, processing time: 0.02768 seconds
Process-2068, Step 2: Resize if scale is not 1: 0.011474 seconds
Process-2067, Step 2: Resize if scale is not 1: 0.014019 seconds
Process-2068, Step 3: Get HOG channels: 6.7e-05 seconds
Process-2067, Step 3: Get HOG channels: 8.4e-05 seconds
Process-2068, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2067, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-2068, Step 5: Compute individual channel HOG features for the entire image: 0.532846 seconds
Process-2067, Step 5: Compute individual channel HOG features for the entire image: 0.550667 seconds
Process-2068, Step 6: Misc initializations: 1e-06 seconds
Process-2067, Step 6: Misc initializations: 1e-06 seconds
Process-2068, Step 7: for loop: 0.039241 seconds
```

Process-2067, Step 7: for loop: 0.038732 seconds

Process-2068All steps time: 0.610168 seconds

Process-2067All steps time: 0.6311950000000001 seconds

Process-2068, Find cars processing time: 0.610414 seconds

Process-2067, Find cars processing time: 0.63145 seconds

Process-2069, Step 1: Divide with 255, processing time: 0.025397 seconds

Process-2069, Step 2: Resize if scale is not 1: 0.01358 seconds

Process-2069, Step 3: Get HOG channels: 8e-05 seconds

Process-2069, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2069, Step 5: Compute individual channel HOG features for the entire image: 0.540819 seconds

Process-2069, Step 6: Misc initializations: 2e-06 seconds

Process-2069, Step 7: for loop: 0.047116 seconds

Process-2069All steps time: 0.627004 seconds

Process-2069, Find cars processing time: 0.627289 seconds

Process-2070, Step 1: Divide with 255, processing time: 0.025479 seconds

Process-2070, Step 2: Resize if scale is not 1: 0.013003 seconds

Process-2070, Step 3: Get HOG channels: 0.000107 seconds

Process-2070, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-2070, Step 5: Compute individual channel HOG features for the entire image: 0.42381 seconds

Process-2070, Step 6: Misc initializations: 0.0 seconds

Process-2070, Step 7: for loop: 0.031871 seconds

Process-2070All steps time: 0.4942830000000003 seconds

Process-2070, Find cars processing time: 0.494581 seconds

Process-2069, Step 1: Divide with 255, processing time: 0.007067 seconds

Process-2069, Step 2: Resize if scale is not 1: 0.000823 seconds

Process-2069, Step 3: Get HOG channels: 6e-06 seconds

Process-2069, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2069, Step 5: Compute individual channel HOG features for the entire image: 0.279817 seconds

Process-2069, Step 6: Misc initializations: 1e-06 seconds

Process-2069, Step 7: for loop: 0.018339 seconds

Process-2069All steps time: 0.3060579999999994 seconds

Process-2069, Find cars processing time: 0.306113 seconds

Process-2069, Step 1: Divide with 255, processing time: 0.005615 seconds

Process-2069, Step 2: Resize if scale is not 1: 0.000502 seconds

Process-2069, Step 3: Get HOG channels: 5e-06 seconds

Process-2069, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2069, Step 5: Compute individual channel HOG features for the entire image: 0.222903 seconds

Process-2069, Step 6: Misc initializations: 1e-06 seconds

Process-2069, Step 7: for loop: 0.013502 seconds

Process-2069All steps time: 0.242533 seconds

Process-2069, Find cars processing time: 0.242599 seconds

Process-2070, Step 1: Divide with 255, processing time: 0.010667 seconds

Process-2070, Step 2: Resize if scale is not 1: 0.001112 seconds

Process-2070, Step 3: Get HOG channels: 5e-06 seconds

Process-2070, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2070, Step 5: Compute individual channel HOG features for the entire image: 0.209871 seconds

Process-2070, Step 6: Misc initializations: 1e-06 seconds

Process-2070, Step 7: for loop: 0.013848 seconds

Process-2070All steps time: 0.235509 seconds

Process-2070, Find cars processing time: 0.235571 seconds

Process-2068, Step 1: Divide with 255, processing time: 0.010966 seconds

Process-2068, Step 2: Resize if scale is not 1: 0.001156 seconds

Process-2068, Step 3: Get HOG channels: 8e-06 seconds

Process-2068, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2068, Step 5: Compute individual channel HOG features for the entire image: 0.34465 seconds

Process-2068, Step 6: Misc initializations: 1e-06 seconds

Process-2068, Step 7: for loop: 0.019634 seconds

Process-2071, Step 1: Divide with 255, processing time: 0.031297 seconds

Process-2071, Step 2: Resize if scale is not 1: 0.003129 seconds

Process-2068All steps time: 0.376424 seconds

Process-2067, Step 1: Divide with 255, processing time: 0.017006 seconds

Process-2071, Step 3: Get HOG channels: 1.9e-05 seconds

Process-2067, Step 2: Resize if scale is not 1: 0.000577 seconds

Process-2071, Step 4: Define blocks and steps as above: 2e-05 seconds

Process-2068, Find cars processing time: 0.3765 seconds

Process-2067, Step 3: Get HOG channels: 4e-06 seconds

Process-2071, Step 5: Compute individual channel HOG features for the entire image: 0.359906 seconds

Process-2067, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2071, Step 6: Misc initializations: 1e-06 seconds

Process-2067, Step 5: Compute individual channel HOG features for the entire image: 0.343832 seconds

Process-2067, Step 6: Misc initializations: 1e-06 seconds

Process-2071, Step 7: for loop: 0.0254 seconds

Process-2067, Step 7: for loop: 0.02299 seconds

Process-2071All steps time: 0.419772 seconds
Process-2067All steps time: 0.384415 seconds

Process-2071, Find cars processing time: 0.419973 seconds
Process-2067, Find cars processing time: 0.384464 seconds

Process-2070, Step 1: Divide with 255, processing time: 0.005671 seconds
Process-2070, Step 2: Resize if scale is not 1: 0.000564 seconds
Process-2070, Step 3: Get HOG channels: 6e-06 seconds
Process-2070, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2070, Step 5: Compute individual channel HOG features for the entire image: 0.253301 seconds
Process-2070, Step 6: Misc initializations: 1e-06 seconds
Process-2070, Step 7: for loop: 0.012883 seconds

Process-2070All steps time: 0.2724329999999999 seconds

Process-2070, Find cars processing time: 0.272499 seconds

Process-2071, Step 1: Divide with 255, processing time: 0.00812 seconds
Process-2071, Step 2: Resize if scale is not 1: 0.001381 seconds
Process-2071, Step 3: Get HOG channels: 8e-06 seconds
Process-2071, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2071, Step 5: Compute individual channel HOG features for the entire image: 0.179034 seconds
Process-2071, Step 6: Misc initializations: 1e-06 seconds
Process-2071, Step 7: for loop: 0.009764 seconds

Process-2071All steps time: 0.198314 seconds

Process-2071, Find cars processing time: 0.19838 seconds

Process-2069, Step 1: Divide with 255, processing time: 0.010598 seconds
Process-2069, Step 2: Resize if scale is not 1: 0.000971 seconds
Process-2069, Step 3: Get HOG channels: 9e-06 seconds
Process-2069, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2069, Step 5: Compute individual channel HOG features for the entire image: 0.198171 seconds
Process-2069, Step 6: Misc initializations: 1e-06 seconds
Process-2069, Step 7: for loop: 0.012558 seconds

Process-2069All steps time: 0.222316 seconds

Process-2069, Find cars processing time: 0.222385 seconds

Process-2067, Step 1: Divide with 255, processing time: 0.010372 seconds
Process-2067, Step 2: Resize if scale is not 1: 0.000893 seconds
Process-2067, Step 3: Get HOG channels: 5e-06 seconds

Process-2067, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2071, Step 1: Divide with 255, processing time: 0.008381 seconds
Process-2067, Step 5: Compute individual channel HOG features for the entire image: 0.185581 seconds
Process-2071, Step 2: Resize if scale is not 1: 0.001102 seconds
Process-2067, Step 6: Misc initializations: 1e-06 seconds
Process-2067, Step 7: for loop: 0.010133 seconds
Process-2071, Step 3: Get HOG channels: 7e-06 seconds

Process-2071, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2067All steps time: 0.20699 seconds
Process-2071, Step 5: Compute individual channel HOG features for the entire image: 0.13866 seconds

Process-2071, Step 6: Misc initializations: 0.0 seconds
Process-2067, Find cars processing time: 0.207047 seconds
Process-2071, Step 7: for loop: 0.00735 seconds

Process-2071All steps time: 0.155505 seconds

Process-2071, Find cars processing time: 0.155563 seconds

Process-2068, Step 1: Divide with 255, processing time: 0.010277 seconds
Process-2068, Step 2: Resize if scale is not 1: 0.001494 seconds
Process-2068, Step 3: Get HOG channels: 8e-06 seconds
Process-2068, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2068, Step 5: Compute individual channel HOG features for the entire image: 0.137223 seconds
Process-2068, Step 6: Misc initializations: 1e-06 seconds
Process-2068, Step 7: for loop: 0.009146 seconds

Process-2068All steps time: 0.158156 seconds

Process-2068, Find cars processing time: 0.158224 seconds

Process-2071, Step 1: Divide with 255, processing time: 0.008837 seconds
Process-2071, Step 2: Resize if scale is not 1: 0.000556 seconds
Process-2071, Step 3: Get HOG channels: 6e-06 seconds
Process-2071, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2071, Step 5: Compute individual channel HOG features for the entire image: 0.12902 seconds
Process-2071, Step 6: Misc initializations: 0.0 seconds
Process-2071, Step 7: for loop: 0.007416 seconds

Process-2071All steps time: 0.145842 seconds

Process-2071, Find cars processing time: 0.14591 seconds

Process-2070, Step 1: Divide with 255, processing time: 0.009466 seconds
Process-2070, Step 2: Resize if scale is not 1: 0.000689 seconds
Process-2070, Step 3: Get HOG channels: 6e-06 seconds

Process-2070, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2070, Step 5: Compute individual channel HOG features for the entire image: 0.137205 seconds

Process-2070, Step 6: Misc initializations: 0.0 seconds

Process-2070, Step 7: for loop: 0.007818 seconds

Process-2070All steps time: 0.15519 seconds

Process-2070, Find cars processing time: 0.155248 seconds

Process-2067, Step 1: Divide with 255, processing time: 0.010015 seconds

Process-2067, Step 2: Resize if scale is not 1: 0.000848 seconds

Process-2067, Step 3: Get HOG channels: 7e-06 seconds

Process-2067, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2067, Step 5: Compute individual channel HOG features for the entire image: 0.163296 seconds

Process-2067, Step 6: Misc initializations: 1e-06 seconds

Process-2067, Step 7: for loop: 0.008021 seconds

Process-2067All steps time: 0.182194 seconds

Process-2067, Find cars processing time: 0.18226 seconds

Process-2069, Step 1: Divide with 255, processing time: 0.01038 seconds

Process-2069, Step 2: Resize if scale is not 1: 0.001329 seconds

Process-2069, Step 3: Get HOG channels: 9e-06 seconds

Process-2069, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2069, Step 5: Compute individual channel HOG features for the entire image: 0.123876 seconds

Process-2069, Step 6: Misc initializations: 0.0 seconds

Process-2069, Step 7: for loop: 0.005521 seconds

Process-2069All steps time: 0.141123 seconds

Process-2069, Find cars processing time: 0.141179 seconds

Process-2068, Step 1: Divide with 255, processing time: 0.010206 seconds

Process-2068, Step 2: Resize if scale is not 1: 0.000901 seconds

Process-2068, Step 3: Get HOG channels: 9e-06 seconds

Process-2068, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2068, Step 5: Compute individual channel HOG features for the entire image: 0.099066 seconds

Process-2068, Step 6: Misc initializations: 0.0 seconds

Process-2068, Step 7: for loop: 0.004746 seconds

Process-2068All steps time: 0.114936 seconds

Process-2068, Find cars processing time: 0.115003 seconds

The times for each task are: [0.610414, 0.63145, 0.627289, 0.494581, 0.306113, 0.242599, 0.23557

Minimum: 0.115003 Maximum: 0.63145 Average: 0.2992 seconds

```
*****
Number of processes used: 5 window size 260
Length of task list: 21
Number of processes used: 5

Process-2072, Step 1: Divide with 255, processing time: 0.024437 seconds
Process-2073, Step 1: Divide with 255, processing time: 0.031916 seconds
Process-2072, Step 2: Resize if scale is not 1: 0.009493 seconds
Process-2073, Step 2: Resize if scale is not 1: 0.015585 seconds
Process-2072, Step 3: Get HOG channels: 0.000116 seconds
Process-2073, Step 3: Get HOG channels: 8e-05 seconds
Process-2072, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-2072, Step 5: Compute individual channel HOG features for the entire image: 0.481556 sec
Process-2073, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-2072, Step 6: Misc initializations: 4e-06 seconds
Process-2072, Step 7: for loop: 0.039374 seconds
Process-2073, Step 5: Compute individual channel HOG features for the entire image: 0.561716 sec

Process-2073, Step 6: Misc initializations: 1e-06 seconds
Process-2072All steps time: 0.554992 seconds

Process-2073, Step 7: for loop: 0.046318 seconds
Process-2072, Find cars processing time: 0.555231 seconds

Process-2073All steps time: 0.655628 seconds

Process-2073, Find cars processing time: 0.655915 seconds

Process-2074, Step 1: Divide with 255, processing time: 0.027031 seconds
Process-2074, Step 2: Resize if scale is not 1: 0.014733 seconds
Process-2074, Step 3: Get HOG channels: 6.4e-05 seconds
Process-2074, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2074, Step 5: Compute individual channel HOG features for the entire image: 0.515413 sec
Process-2074, Step 6: Misc initializations: 1e-06 seconds
Process-2074, Step 7: for loop: 0.038089 seconds

Process-2074All steps time: 0.595338 seconds

Process-2074, Find cars processing time: 0.595553 seconds

Process-2072, Step 1: Divide with 255, processing time: 0.0089 seconds
Process-2072, Step 2: Resize if scale is not 1: 0.000936 seconds
Process-2072, Step 3: Get HOG channels: 7e-06 seconds
Process-2072, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2072, Step 5: Compute individual channel HOG features for the entire image: 0.338505 sec
Process-2072, Step 6: Misc initializations: 0.0 seconds
```

Process-2072, Step 7: for loop: 0.018864 seconds

Process-2072All steps time: 0.367219 seconds

Process-2072, Find cars processing time: 0.367289 seconds

Process-2075, Step 1: Divide with 255, processing time: 0.029901 seconds

Process-2075, Step 2: Resize if scale is not 1: 0.014895 seconds

Process-2075, Step 3: Get HOG channels: 6.2e-05 seconds

Process-2075, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2075, Step 5: Compute individual channel HOG features for the entire image: 0.429746 seconds

Process-2075, Step 6: Misc initializations: 1e-06 seconds

Process-2075, Step 7: for loop: 0.031951 seconds

Process-2075All steps time: 0.506566 seconds

Process-2075, Find cars processing time: 0.506779 seconds

Process-2073, Step 1: Divide with 255, processing time: 0.026041 seconds

Process-2073, Step 2: Resize if scale is not 1: 0.000797 seconds

Process-2073, Step 3: Get HOG channels: 5e-06 seconds

Process-2073, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2073, Step 5: Compute individual channel HOG features for the entire image: 0.310355 seconds

Process-2073, Step 6: Misc initializations: 0.0 seconds

Process-2073, Step 7: for loop: 0.019091 seconds

Process-2073All steps time: 0.356294 seconds

Process-2073, Find cars processing time: 0.356355 seconds

Process-2076, Step 1: Divide with 255, processing time: 0.027302 seconds

Process-2076, Step 2: Resize if scale is not 1: 0.003043 seconds

Process-2076, Step 3: Get HOG channels: 1.8e-05 seconds

Process-2076, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2076, Step 5: Compute individual channel HOG features for the entire image: 0.357886 seconds

Process-2076, Step 6: Misc initializations: 1e-06 seconds

Process-2076, Step 7: for loop: 0.025367 seconds

Process-2076All steps time: 0.4136219999999993 seconds

Process-2076, Find cars processing time: 0.413769 seconds

Process-2072, Step 1: Divide with 255, processing time: 0.006104 seconds

Process-2072, Step 2: Resize if scale is not 1: 0.000741 seconds

Process-2072, Step 3: Get HOG channels: 7e-06 seconds

Process-2072, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2072, Step 5: Compute individual channel HOG features for the entire image: 0.198353 seconds

Process-2072, Step 6: Misc initializations: 1e-06 seconds

Process-2072, Step 7: for loop: 0.012914 seconds

Process-2072All steps time: 0.21812600000000001 seconds

Process-2072, Find cars processing time: 0.218194 seconds

Process-2074, Step 1: Divide with 255, processing time: 0.010714 seconds

Process-2074, Step 2: Resize if scale is not 1: 0.001182 seconds

Process-2074, Step 3: Get HOG channels: 7e-06 seconds

Process-2074, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2074, Step 5: Compute individual channel HOG features for the entire image: 0.354086 seconds

Process-2074, Step 6: Misc initializations: 1e-06 seconds

Process-2074, Step 7: for loop: 0.019567 seconds

Process-2074All steps time: 0.38556399999999996 seconds

Process-2074, Find cars processing time: 0.385656 seconds

Process-2073, Step 1: Divide with 255, processing time: 0.008796 seconds

Process-2073, Step 2: Resize if scale is not 1: 0.000793 seconds

Process-2073, Step 3: Get HOG channels: 7e-06 seconds

Process-2073, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2073, Step 5: Compute individual channel HOG features for the entire image: 0.242826 seconds

Process-2073, Step 6: Misc initializations: 1e-06 seconds

Process-2073, Step 7: for loop: 0.013224 seconds

Process-2073All steps time: 0.265653 seconds

Process-2073, Find cars processing time: 0.26571 seconds

Process-2075, Step 1: Divide with 255, processing time: 0.010789 seconds

Process-2075, Step 2: Resize if scale is not 1: 0.00172 seconds

Process-2075, Step 3: Get HOG channels: 9e-06 seconds

Process-2075, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2075, Step 5: Compute individual channel HOG features for the entire image: 0.215799 seconds

Process-2075, Step 6: Misc initializations: 1e-06 seconds

Process-2075, Step 7: for loop: 0.014376 seconds

Process-2075All steps time: 0.242701 seconds

Process-2075, Find cars processing time: 0.242771 seconds

Process-2074, Step 1: Divide with 255, processing time: 0.00912 seconds

Process-2074, Step 2: Resize if scale is not 1: 0.000904 seconds

Process-2074, Step 3: Get HOG channels: 5e-06 seconds

Process-2074, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2074, Step 5: Compute individual channel HOG features for the entire image: 0.134229 seconds

Process-2072, Step 1: Divide with 255, processing time: 0.021997 seconds

Process-2074, Step 6: Misc initializations: 1e-06 seconds

Process-2072, Step 2: Resize if scale is not 1: 0.000587 seconds

Process-2074, Step 7: for loop: 0.007378 seconds

Process-2072, Step 3: Get HOG channels: 6e-06 seconds

Process-2072, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2074All steps time: 0.15164199999999997 seconds

Process-2072, Step 5: Compute individual channel HOG features for the entire image: 0.176083 seconds

Process-2072, Step 6: Misc initializations: 1e-06 seconds

Process-2074, Find cars processing time: 0.151689 seconds

Process-2072, Step 7: for loop: 0.010256 seconds

Process-2072All steps time: 0.2089349999999998 seconds

Process-2072, Find cars processing time: 0.209016 seconds

Process-2075, Step 1: Divide with 255, processing time: 0.010407 seconds

Process-2075, Step 2: Resize if scale is not 1: 0.000866 seconds

Process-2075, Step 3: Get HOG channels: 9e-06 seconds

Process-2075, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2075, Step 5: Compute individual channel HOG features for the entire image: 0.12837 seconds

Process-2075, Step 6: Misc initializations: 1e-06 seconds

Process-2075, Step 7: for loop: 0.008022 seconds

Process-2075All steps time: 0.147685 seconds

Process-2075, Find cars processing time: 0.14776 seconds

Process-2072, Step 1: Divide with 255, processing time: 0.008658 seconds

Process-2072, Step 2: Resize if scale is not 1: 0.001211 seconds

Process-2072, Step 3: Get HOG channels: 8e-06 seconds

Process-2072, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2072, Step 5: Compute individual channel HOG features for the entire image: 0.147531 seconds

Process-2072, Step 6: Misc initializations: 0.0 seconds

Process-2072, Step 7: for loop: 0.007411 seconds

Process-2072All steps time: 0.164826 seconds

Process-2072, Find cars processing time: 0.164897 seconds

Process-2074, Step 1: Divide with 255, processing time: 0.01006 seconds

Process-2074, Step 2: Resize if scale is not 1: 0.001331 seconds

Process-2074, Step 3: Get HOG channels: 7e-06 seconds

Process-2074, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2074, Step 5: Compute individual channel HOG features for the entire image: 0.117161 seconds

Process-2074, Step 6: Misc initializations: 1e-06 seconds

Process-2074, Step 7: for loop: 0.006076 seconds

Process-2074All steps time: 0.134643 seconds

Process-2074, Find cars processing time: 0.134705 seconds

Process-2076, Step 1: Divide with 255, processing time: 0.01192 seconds

Process-2076, Step 2: Resize if scale is not 1: 0.000865 seconds

Process-2076, Step 3: Get HOG channels: 9e-06 seconds

Process-2076, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2076, Step 5: Compute individual channel HOG features for the entire image: 0.265891 seconds

Process-2076, Step 6: Misc initializations: 1e-06 seconds

Process-2076, Step 7: for loop: 0.018979 seconds

Process-2076All steps time: 0.297674 seconds

Process-2076, Find cars processing time: 0.297775 seconds

Process-2073, Step 1: Divide with 255, processing time: 0.010198 seconds

Process-2073, Step 2: Resize if scale is not 1: 0.000795 seconds

Process-2073, Step 3: Get HOG channels: 7e-06 seconds

Process-2073, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2073, Step 5: Compute individual channel HOG features for the entire image: 0.14534 seconds

Process-2073, Step 6: Misc initializations: 1e-06 seconds

Process-2073, Step 7: for loop: 0.00799 seconds

Process-2073All steps time: 0.1643369999999998 seconds

Process-2073, Find cars processing time: 0.1644 seconds

Process-2074, Step 1: Divide with 255, processing time: 0.009867 seconds

Process-2074, Step 2: Resize if scale is not 1: 0.000865 seconds

Process-2074, Step 3: Get HOG channels: 8e-06 seconds

Process-2074, Step 4: Define blocks and steps as above: 1.7e-05 seconds

Process-2074, Step 5: Compute individual channel HOG features for the entire image: 0.089107 seconds

Process-2074, Step 6: Misc initializations: 0.0 seconds

Process-2074, Step 7: for loop: 0.005109 seconds

Process-2074All steps time: 0.1049730000000001 seconds

Process-2074, Find cars processing time: 0.105027 seconds

Process-2075, Step 1: Divide with 255, processing time: 0.010847 seconds

Process-2075, Step 2: Resize if scale is not 1: 0.001091 seconds

Process-2075, Step 3: Get HOG channels: 5e-06 seconds

Process-2075, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2075, Step 5: Compute individual channel HOG features for the entire image: 0.099552 seconds

Process-2075, Step 6: Misc initializations: 0.0 seconds

Process-2075, Step 7: for loop: 0.005358 seconds

Process-2075All steps time: 0.116857 seconds

Process-2075, Find cars processing time: 0.11691 seconds

Process-2072, Step 1: Divide with 255, processing time: 0.010302 seconds

Process-2072, Step 2: Resize if scale is not 1: 0.000768 seconds

Process-2072, Step 3: Get HOG channels: 7e-06 seconds

Process-2072, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2072, Step 5: Compute individual channel HOG features for the entire image: 0.126316 sec

Process-2072, Step 6: Misc initializations: 1e-06 seconds

Process-2072, Step 7: for loop: 0.007202 seconds

Process-2072All steps time: 0.144604 seconds

Process-2072, Find cars processing time: 0.144674 seconds

The times for each task are: [0.555231, 0.655915, 0.595553, 0.367289, 0.506779, 0.356355, 0.4137]

Minimum: 0.105027 Maximum: 0.655915 Average: 0.2952 seconds

Window sizes used: [0.75, 0.95, 1.15, 1.35, 1.55]

5 processes used for testing 5 window sizes

Processing times for each image [4.0421, 3.7092, 3.6933, 3.7667, 3.7143, 3.8683] with an average

Time elapsed so far... 281.15580000000006

#####

=====
=====

Number of processes used: 1 window size 260

Length of task list: 22

Number of processes used: 1

Process-2077. Step 1: Divide with 255. processing time: 0.023723 seconds

Process-2077. Step 2: Resize if scale is not 1: 0.001572 seconds

Process-2077. Step 3: Get HOG channels: 2.5e-05 seconds

Process-2077. Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2077. Step 5: Compute individual channel HOG features for the entire image: 0.189818 sec

Process-2077 Step 6: Misc initializations: 0.0 seconds

Process-2077 Step 7: for loop: 0 013369 seconds

Process-2077All steps time: 0.228512 seconds

Process-2077: Find cars processing time: 0.228678 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.010337 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000942 seconds
Process-2077, Step 3: Get HOG channels: 8e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.184153 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.012775 seconds

Process-2077All steps time: 0.20822400000000002 seconds

Process-2077, Find cars processing time: 0.208299 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.010771 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000923 seconds
Process-2077, Step 3: Get HOG channels: 9e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.237412 seconds
Process-2077, Step 6: Misc initializations: 0.0 seconds
Process-2077, Step 7: for loop: 0.015967 seconds

Process-2077All steps time: 0.26509 seconds

Process-2077, Find cars processing time: 0.265169 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.006665 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000576 seconds
Process-2077, Step 3: Get HOG channels: 5e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.206044 seconds
Process-2077, Step 6: Misc initializations: 0.0 seconds
Process-2077, Step 7: for loop: 0.011877 seconds

Process-2077All steps time: 0.225159 seconds

Process-2077, Find cars processing time: 0.225212 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.010467 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000869 seconds
Process-2077, Step 3: Get HOG channels: 8e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.154126 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.009943 seconds

Process-2077All steps time: 0.17542300000000002 seconds

Process-2077, Find cars processing time: 0.175492 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.0109 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000944 seconds
Process-2077, Step 3: Get HOG channels: 9e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.151043 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.009688 seconds

Process-2077All steps time: 0.17259500000000003 seconds

Process-2077, Find cars processing time: 0.172677 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.008409 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000504 seconds
Process-2077, Step 3: Get HOG channels: 5e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.151729 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.011391 seconds

Process-2077All steps time: 0.172044 seconds

Process-2077, Find cars processing time: 0.172096 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.013919 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.00055 seconds
Process-2077, Step 3: Get HOG channels: 6e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.196991 seconds
Process-2077, Step 6: Misc initializations: 0.0 seconds
Process-2077, Step 7: for loop: 0.012401 seconds

Process-2077All steps time: 0.223873 seconds

Process-2077, Find cars processing time: 0.223935 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.010877 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000918 seconds
Process-2077, Step 3: Get HOG channels: 9e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.172518 seconds
Process-2077, Step 6: Misc initializations: 0.0 seconds
Process-2077, Step 7: for loop: 0.009508 seconds

Process-2077All steps time: 0.1938399999999998 seconds

Process-2077, Find cars processing time: 0.193928 seconds

```
Process-2077, Step 1: Divide with 255, processing time: 0.010844 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000882 seconds
Process-2077, Step 3: Get HOG channels: 1e-05 seconds
Process-2077, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.1774 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.009627 seconds
```

Process-2077All steps time: 0.198774 seconds

Process-2077, Find cars processing time: 0.198866 seconds

```
Process-2077, Step 1: Divide with 255, processing time: 0.007936 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000555 seconds
Process-2077, Step 3: Get HOG channels: 6e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.17015 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.009756 seconds
```

Process-2077All steps time: 0.1884100000000002 seconds

Process-2077, Find cars processing time: 0.188483 seconds

```
Process-2077, Step 1: Divide with 255, processing time: 0.010338 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000826 seconds
Process-2077, Step 3: Get HOG channels: 9e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.196173 seconds
Process-2077, Step 6: Misc initializations: 0.0 seconds
Process-2077, Step 7: for loop: 0.010327 seconds
```

Process-2077All steps time: 0.2176809999999999 seconds

Process-2077, Find cars processing time: 0.217761 seconds

```
Process-2077, Step 1: Divide with 255, processing time: 0.010511 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000931 seconds
Process-2077, Step 3: Get HOG channels: 9e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.173247 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.008358 seconds
```

Process-2077All steps time: 0.1930660000000002 seconds

Process-2077, Find cars processing time: 0.193159 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.008646 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000791 seconds
Process-2077, Step 3: Get HOG channels: 2.1e-05 seconds
Process-2077, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.142836 seconds
Process-2077, Step 6: Misc initializations: 0.0 seconds
Process-2077, Step 7: for loop: 0.007195 seconds

Process-2077All steps time: 0.1595 seconds

Process-2077, Find cars processing time: 0.159583 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.010895 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000843 seconds
Process-2077, Step 3: Get HOG channels: 9e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.144838 seconds
Process-2077, Step 6: Misc initializations: 0.0 seconds
Process-2077, Step 7: for loop: 0.007177 seconds

Process-2077All steps time: 0.16377199999999997 seconds

Process-2077, Find cars processing time: 0.163851 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.009525 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000524 seconds
Process-2077, Step 3: Get HOG channels: 5e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.141663 seconds
Process-2077, Step 6: Misc initializations: 0.0 seconds
Process-2077, Step 7: for loop: 0.007263 seconds

Process-2077All steps time: 0.1589870000000002 seconds

Process-2077, Find cars processing time: 0.159055 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.01102 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.00106 seconds
Process-2077, Step 3: Get HOG channels: 9e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.122792 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.005049 seconds

Process-2077All steps time: 0.139941 seconds

Process-2077, Find cars processing time: 0.140027 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.00648 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000864 seconds
Process-2077, Step 3: Get HOG channels: 6e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.127518 seconds
Process-2077, Step 6: Misc initializations: 0.0 seconds
Process-2077, Step 7: for loop: 0.00727 seconds

Process-2077All steps time: 0.142144 seconds

Process-2077, Find cars processing time: 0.142196 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.008492 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.00062 seconds
Process-2077, Step 3: Get HOG channels: 6e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.128711 seconds
Process-2077, Step 6: Misc initializations: 2e-06 seconds
Process-2077, Step 7: for loop: 0.007384 seconds

Process-2077All steps time: 0.145222 seconds

Process-2077, Find cars processing time: 0.145314 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.008071 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000754 seconds
Process-2077, Step 3: Get HOG channels: 7e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.101083 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.006103 seconds

Process-2077All steps time: 0.116027 seconds

Process-2077, Find cars processing time: 0.116106 seconds

Process-2077, Step 1: Divide with 255, processing time: 0.010676 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.000812 seconds
Process-2077, Step 3: Get HOG channels: 7e-06 seconds
Process-2077, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.105931 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.006519 seconds

Process-2077All steps time: 0.123955 seconds

Process-2077, Find cars processing time: 0.124034 seconds

```
Process-2077, Step 1: Divide with 255, processing time: 0.010625 seconds
Process-2077, Step 2: Resize if scale is not 1: 0.00204 seconds
Process-2077, Step 3: Get HOG channels: 1.3e-05 seconds
Process-2077, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-2077, Step 5: Compute individual channel HOG features for the entire image: 0.104777 seconds
Process-2077, Step 6: Misc initializations: 1e-06 seconds
Process-2077, Step 7: for loop: 0.005676 seconds
```

Process-2077All steps time: 0.123144 seconds

Process-2077, Find cars processing time: 0.123237 seconds

The times for each task are: [0.228678, 0.208299, 0.265169, 0.225212, 0.175492, 0.172677, 0.1720

Minimum: 0.116106 Maximum: 0.265169 Average: 0.179 seconds

```
*****
Number of processes used: 1 window size 260
Length of task list: 22
Number of processes used: 1
```

```
Process-2078, Step 1: Divide with 255, processing time: 0.031464 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.002382 seconds
Process-2078, Step 3: Get HOG channels: 3.7e-05 seconds
Process-2078, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.242493 seconds
Process-2078, Step 6: Misc initializations: 0.0 seconds
Process-2078, Step 7: for loop: 0.014868 seconds
```

Process-2078All steps time: 0.291251 seconds

Process-2078, Find cars processing time: 0.29146 seconds

```
Process-2078, Step 1: Divide with 255, processing time: 0.01076 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000645 seconds
Process-2078, Step 3: Get HOG channels: 7e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.240564 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.01409 seconds
```

Process-2078All steps time: 0.266074 seconds

Process-2078, Find cars processing time: 0.266182 seconds

```
Process-2078, Step 1: Divide with 255, processing time: 0.010823 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.001347 seconds
```

Process-2078, Step 3: Get HOG channels: 1.3e-05 seconds
Process-2078, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.238054 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.015628 seconds

Process-2078All steps time: 0.2658769999999999 seconds

Process-2078, Find cars processing time: 0.265964 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.0105 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000537 seconds
Process-2078, Step 3: Get HOG channels: 6e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.199267 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.012794 seconds

Process-2078All steps time: 0.223112 seconds

Process-2078, Find cars processing time: 0.223185 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.015331 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000751 seconds
Process-2078, Step 3: Get HOG channels: 5e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.164872 seconds
Process-2078, Step 6: Misc initializations: 0.0 seconds
Process-2078, Step 7: for loop: 0.009839 seconds

Process-2078All steps time: 0.190805 seconds

Process-2078, Find cars processing time: 0.190876 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010901 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000607 seconds
Process-2078, Step 3: Get HOG channels: 7e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 1.6e-05 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.208187 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.013184 seconds

Process-2078All steps time: 0.2329030000000003 seconds

Process-2078, Find cars processing time: 0.233007 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.011892 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.00081 seconds

Process-2078, Step 3: Get HOG channels: 7e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.206311 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.010575 seconds

Process-2078All steps time: 0.229603 seconds

Process-2078, Find cars processing time: 0.22968 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.011058 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000848 seconds
Process-2078, Step 3: Get HOG channels: 9e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.172739 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.010286 seconds

Process-2078All steps time: 0.1949489999999998 seconds

Process-2078, Find cars processing time: 0.195032 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010802 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000892 seconds
Process-2078, Step 3: Get HOG channels: 9e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.178119 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.009831 seconds

Process-2078All steps time: 0.199662 seconds

Process-2078, Find cars processing time: 0.199764 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.00992 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000536 seconds
Process-2078, Step 3: Get HOG channels: 6e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.168988 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.008437 seconds

Process-2078All steps time: 0.187893 seconds

Process-2078, Find cars processing time: 0.187984 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010464 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000637 seconds

Process-2078, Step 3: Get HOG channels: 8e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.154784 seconds
Process-2078, Step 6: Misc initializations: 0.0 seconds
Process-2078, Step 7: for loop: 0.007562 seconds

Process-2078All steps time: 0.173462 seconds

Process-2078, Find cars processing time: 0.173546 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010973 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.00053 seconds
Process-2078, Step 3: Get HOG channels: 6e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.138079 seconds
Process-2078, Step 6: Misc initializations: 0.0 seconds
Process-2078, Step 7: for loop: 0.007507 seconds

Process-2078All steps time: 0.157101 seconds

Process-2078, Find cars processing time: 0.15717 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.009955 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000716 seconds
Process-2078, Step 3: Get HOG channels: 7e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.122374 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.006077 seconds

Process-2078All steps time: 0.1391369999999998 seconds

Process-2078, Find cars processing time: 0.13922 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010709 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.00082 seconds
Process-2078, Step 3: Get HOG channels: 8e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.116946 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.005618 seconds

Process-2078All steps time: 0.134111 seconds

Process-2078, Find cars processing time: 0.134187 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010369 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000791 seconds

Process-2078, Step 3: Get HOG channels: 8e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.125772 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.00602 seconds

Process-2078All steps time: 0.1429699999999999 seconds

Process-2078, Find cars processing time: 0.143054 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010855 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000819 seconds
Process-2078, Step 3: Get HOG channels: 9e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.120217 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.006997 seconds

Process-2078All steps time: 0.138908 seconds

Process-2078, Find cars processing time: 0.139006 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.009954 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000615 seconds
Process-2078, Step 3: Get HOG channels: 7e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.0984 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.00424 seconds

Process-2078All steps time: 0.113226 seconds

Process-2078, Find cars processing time: 0.113315 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010554 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.001402 seconds
Process-2078, Step 3: Get HOG channels: 8e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.107622 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.005955 seconds

Process-2078All steps time: 0.125552 seconds

Process-2078, Find cars processing time: 0.125641 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010848 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000811 seconds

Process-2078, Step 3: Get HOG channels: 1e-05 seconds
Process-2078, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.138297 seconds
Process-2078, Step 6: Misc initializations: 0.0 seconds
Process-2078, Step 7: for loop: 0.00635 seconds

Process-2078All steps time: 0.156326 seconds

Process-2078, Find cars processing time: 0.156433 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.011174 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000839 seconds
Process-2078, Step 3: Get HOG channels: 6e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.128016 seconds
Process-2078, Step 6: Misc initializations: 0.0 seconds
Process-2078, Step 7: for loop: 0.007002 seconds

Process-2078All steps time: 0.147044 seconds

Process-2078, Find cars processing time: 0.147152 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010461 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.000758 seconds
Process-2078, Step 3: Get HOG channels: 9e-06 seconds
Process-2078, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.128439 seconds
Process-2078, Step 6: Misc initializations: 1e-06 seconds
Process-2078, Step 7: for loop: 0.007138 seconds

Process-2078All steps time: 0.146815 seconds

Process-2078, Find cars processing time: 0.146908 seconds

Process-2078, Step 1: Divide with 255, processing time: 0.010818 seconds
Process-2078, Step 2: Resize if scale is not 1: 0.003366 seconds
Process-2078, Step 3: Get HOG channels: 1.1e-05 seconds
Process-2078, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2078, Step 5: Compute individual channel HOG features for the entire image: 0.128867 seconds
Process-2078, Step 6: Misc initializations: 0.0 seconds
Process-2078, Step 7: for loop: 0.00607 seconds

Process-2078All steps time: 0.149142 seconds

Process-2078, Find cars processing time: 0.149245 seconds

The times for each task are: [0.29146, 0.266182, 0.265964, 0.223185, 0.190876, 0.233007, 0.22968]

Minimum: 0.113315 Maximum: 0.29146 Average: 0.1822 seconds

Number of processes used: 1 window size 260
Length of task list: 22
Number of processes used: 1

Process-2079, Step 1: Divide with 255, processing time: 0.028876 seconds

Process-2079, Step 2: Resize if scale is not 1: 0.001995 seconds

Process-2079, Step 3: Get HOG channels: 1.9e-05 seconds

Process-2079, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.22065 seconds

Process-2079, Step 6: Misc initializations: 1e-06 seconds

Process-2079, Step 7: for loop: 0.017742 seconds

Process-2079All steps time: 0.2692889999999995 seconds

Process-2079, Find cars processing time: 0.269454 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010829 seconds

Process-2079, Step 2: Resize if scale is not 1: 0.000711 seconds

Process-2079, Step 3: Get HOG channels: 6e-06 seconds

Process-2079, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.228929 seconds

Process-2079, Step 6: Misc initializations: 0.0 seconds

Process-2079, Step 7: for loop: 0.012647 seconds

Process-2079All steps time: 0.25313 seconds

Process-2079, Find cars processing time: 0.25322 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010867 seconds

Process-2079, Step 2: Resize if scale is not 1: 0.001021 seconds

Process-2079, Step 3: Get HOG channels: 1e-05 seconds

Process-2079, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.182081 seconds

Process-2079, Step 6: Misc initializations: 0.0 seconds

Process-2079, Step 7: for loop: 0.011988 seconds

Process-2079All steps time: 0.205977 seconds

Process-2079, Find cars processing time: 0.206059 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.015898 seconds

Process-2079, Step 2: Resize if scale is not 1: 0.000647 seconds

Process-2079, Step 3: Get HOG channels: 6e-06 seconds

Process-2079, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.179882 seconds

Process-2079, Step 6: Misc initializations: 0.0 seconds
Process-2079, Step 7: for loop: 0.01206 seconds

Process-2079All steps time: 0.2084999999999996 seconds

Process-2079, Find cars processing time: 0.208591 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010865 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000919 seconds
Process-2079, Step 3: Get HOG channels: 9e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.167578 seconds
Process-2079, Step 6: Misc initializations: 2e-06 seconds
Process-2079, Step 7: for loop: 0.009949 seconds

Process-2079All steps time: 0.189332 seconds

Process-2079, Find cars processing time: 0.189422 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010306 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000495 seconds
Process-2079, Step 3: Get HOG channels: 5e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.162217 seconds
Process-2079, Step 6: Misc initializations: 1e-06 seconds
Process-2079, Step 7: for loop: 0.009995 seconds

Process-2079All steps time: 0.183025 seconds

Process-2079, Find cars processing time: 0.183099 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010689 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000985 seconds
Process-2079, Step 3: Get HOG channels: 1e-05 seconds
Process-2079, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.158243 seconds
Process-2079, Step 6: Misc initializations: 0.0 seconds
Process-2079, Step 7: for loop: 0.009578 seconds

Process-2079All steps time: 0.179515 seconds

Process-2079, Find cars processing time: 0.179606 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.013507 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000649 seconds
Process-2079, Step 3: Get HOG channels: 1.7e-05 seconds
Process-2079, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.169318 seconds

Process-2079, Step 6: Misc initializations: 1e-06 seconds
Process-2079, Step 7: for loop: 0.010097 seconds

Process-2079All steps time: 0.193595 seconds

Process-2079, Find cars processing time: 0.193673 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010489 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000625 seconds
Process-2079, Step 3: Get HOG channels: 7e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.198259 seconds
Process-2079, Step 6: Misc initializations: 1e-06 seconds
Process-2079, Step 7: for loop: 0.010403 seconds

Process-2079All steps time: 0.219791 seconds

Process-2079, Find cars processing time: 0.219878 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010922 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000879 seconds
Process-2079, Step 3: Get HOG channels: 1e-05 seconds
Process-2079, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.206394 seconds
Process-2079, Step 6: Misc initializations: 1e-06 seconds
Process-2079, Step 7: for loop: 0.008572 seconds

Process-2079All steps time: 0.226787 seconds

Process-2079, Find cars processing time: 0.226886 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010826 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000834 seconds
Process-2079, Step 3: Get HOG channels: 9e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.165076 seconds
Process-2079, Step 6: Misc initializations: 0.0 seconds
Process-2079, Step 7: for loop: 0.008995 seconds

Process-2079All steps time: 0.185749 seconds

Process-2079, Find cars processing time: 0.185839 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010826 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000639 seconds
Process-2079, Step 3: Get HOG channels: 6e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.16896 seconds

Process-2079, Step 6: Misc initializations: 0.0 seconds
Process-2079, Step 7: for loop: 0.009536 seconds

Process-2079All steps time: 0.1899749999999998 seconds

Process-2079, Find cars processing time: 0.190055 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.009758 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.00088 seconds
Process-2079, Step 3: Get HOG channels: 8e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.154141 seconds
Process-2079, Step 6: Misc initializations: 1e-06 seconds
Process-2079, Step 7: for loop: 0.007419 seconds

Process-2079All steps time: 0.172216 seconds

Process-2079, Find cars processing time: 0.172302 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010301 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000547 seconds
Process-2079, Step 3: Get HOG channels: 6e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.141452 seconds
Process-2079, Step 6: Misc initializations: 0.0 seconds
Process-2079, Step 7: for loop: 0.007146 seconds

Process-2079All steps time: 0.15946 seconds

Process-2079, Find cars processing time: 0.159531 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010897 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000644 seconds
Process-2079, Step 3: Get HOG channels: 7e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.148472 seconds
Process-2079, Step 6: Misc initializations: 1e-06 seconds
Process-2079, Step 7: for loop: 0.005846 seconds

Process-2079All steps time: 0.165875 seconds

Process-2079, Find cars processing time: 0.165958 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.009724 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000675 seconds
Process-2079, Step 3: Get HOG channels: 7e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.118822 seconds

Process-2079, Step 6: Misc initializations: 0.0 seconds
Process-2079, Step 7: for loop: 0.00596 seconds

Process-2079All steps time: 0.1351979999999998 seconds

Process-2079, Find cars processing time: 0.135276 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.008857 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000585 seconds
Process-2079, Step 3: Get HOG channels: 7e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.126094 seconds
Process-2079, Step 6: Misc initializations: 0.0 seconds
Process-2079, Step 7: for loop: 0.004985 seconds

Process-2079All steps time: 0.140536 seconds

Process-2079, Find cars processing time: 0.140605 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.009107 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000862 seconds
Process-2079, Step 3: Get HOG channels: 6e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.126258 seconds
Process-2079, Step 6: Misc initializations: 0.0 seconds

Process-2079, Step 7: for loop: 0.007154 seconds

Process-2079All steps time: 0.143393 seconds

Process-2079, Find cars processing time: 0.14345 seconds
Process-2079, Step 1: Divide with 255, processing time: 0.009408 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000496 seconds
Process-2079, Step 3: Get HOG channels: 5e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.126785 seconds
Process-2079, Step 6: Misc initializations: 1e-06 seconds
Process-2079, Step 7: for loop: 0.007065 seconds

Process-2079All steps time: 0.143765 seconds

Process-2079, Find cars processing time: 0.143838 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.01037 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000562 seconds
Process-2079, Step 3: Get HOG channels: 6e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.127003 seconds

Process-2079, Step 6: Misc initializations: 1e-06 seconds
Process-2079, Step 7: for loop: 0.007177 seconds

Process-2079All steps time: 0.145126 seconds

Process-2079, Find cars processing time: 0.145199 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.010775 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.000794 seconds
Process-2079, Step 3: Get HOG channels: 9e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.146904 seconds
Process-2079, Step 6: Misc initializations: 0.0 seconds
Process-2079, Step 7: for loop: 0.007259 seconds

Process-2079All steps time: 0.1657499999999998 seconds

Process-2079, Find cars processing time: 0.165831 seconds

Process-2079, Step 1: Divide with 255, processing time: 0.009568 seconds
Process-2079, Step 2: Resize if scale is not 1: 0.001273 seconds
Process-2079, Step 3: Get HOG channels: 7e-06 seconds
Process-2079, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2079, Step 5: Compute individual channel HOG features for the entire image: 0.111861 seconds
Process-2079, Step 6: Misc initializations: 0.0 seconds
Process-2079, Step 7: for loop: 0.006141 seconds

Process-2079All steps time: 0.128858 seconds

Process-2079, Find cars processing time: 0.128927 seconds

The times for each task are: [0.269454, 0.25322, 0.206059, 0.208591, 0.189422, 0.183099, 0.179601]

Minimum: 0.128927 Maximum: 0.269454 Average: 0.1821 seconds

Number of processes used: 1 window size 260
Length of task list: 22
Number of processes used: 1

Process-2080, Step 1: Divide with 255, processing time: 0.033853 seconds
Process-2080, Step 2: Resize if scale is not 1: 0.002464 seconds
Process-2080, Step 3: Get HOG channels: 2.7e-05 seconds
Process-2080, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.247261 seconds
Process-2080, Step 6: Misc initializations: 1e-06 seconds
Process-2080, Step 7: for loop: 0.020902 seconds

Process-2080All steps time: 0.304515 seconds

Process-2080, Find cars processing time: 0.304713 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010578 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000665 seconds

Process-2080, Step 3: Get HOG channels: 7e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.231258 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.013219 seconds

Process-2080All steps time: 0.25573599999999996 seconds

Process-2080, Find cars processing time: 0.25581 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010039 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000638 seconds

Process-2080, Step 3: Get HOG channels: 6e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.196002 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.012832 seconds

Process-2080All steps time: 0.21952500000000003 seconds

Process-2080, Find cars processing time: 0.2196 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.00966 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000657 seconds

Process-2080, Step 3: Get HOG channels: 7e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.240699 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.016885 seconds

Process-2080All steps time: 0.267916 seconds

Process-2080, Find cars processing time: 0.268001 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.01087 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000929 seconds

Process-2080, Step 3: Get HOG channels: 9e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.221556 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.012908 seconds

Process-2080All steps time: 0.246281 seconds

Process-2080, Find cars processing time: 0.246373 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.009561 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000608 seconds

Process-2080, Step 3: Get HOG channels: 7e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.198981 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.012386 seconds

Process-2080All steps time: 0.221552 seconds

Process-2080, Find cars processing time: 0.221629 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010905 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.001032 seconds

Process-2080, Step 3: Get HOG channels: 1e-05 seconds

Process-2080, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.189977 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.009857 seconds

Process-2080All steps time: 0.211791 seconds

Process-2080, Find cars processing time: 0.211878 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.011103 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000933 seconds

Process-2080, Step 3: Get HOG channels: 8e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.160571 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.009553 seconds

Process-2080All steps time: 0.182178 seconds

Process-2080, Find cars processing time: 0.182262 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.009154 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000673 seconds

Process-2080, Step 3: Get HOG channels: 6e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.169743 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.009644 seconds

Process-2080All steps time: 0.1892269999999998 seconds

Process-2080, Find cars processing time: 0.189294 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010191 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000696 seconds

Process-2080, Step 3: Get HOG channels: 8e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.177014 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.009591 seconds

Process-2080All steps time: 0.19751 seconds

Process-2080, Find cars processing time: 0.197594 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010849 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000637 seconds

Process-2080, Step 3: Get HOG channels: 7e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.183926 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.008347 seconds

Process-2080All steps time: 0.203775 seconds

Process-2080, Find cars processing time: 0.203862 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010821 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000855 seconds

Process-2080, Step 3: Get HOG channels: 9e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.129783 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.007353 seconds

Process-2080All steps time: 0.1488300000000002 seconds

Process-2080, Find cars processing time: 0.148911 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.011038 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000907 seconds

Process-2080, Step 3: Get HOG channels: 9e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.115907 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.005953 seconds

Process-2080All steps time: 0.133824 seconds

Process-2080, Find cars processing time: 0.133926 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.00925 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000637 seconds

Process-2080, Step 3: Get HOG channels: 7e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.114202 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.006194 seconds

Process-2080All steps time: 0.130298 seconds

Process-2080, Find cars processing time: 0.130372 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010252 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000816 seconds

Process-2080, Step 3: Get HOG channels: 9e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.11399 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.005783 seconds

Process-2080All steps time: 0.130858 seconds

Process-2080, Find cars processing time: 0.130927 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.009257 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000605 seconds

Process-2080, Step 3: Get HOG channels: 6e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.141634 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.007276 seconds

Process-2080All steps time: 0.158786 seconds

Process-2080, Find cars processing time: 0.158857 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010754 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000731 seconds

Process-2080, Step 3: Get HOG channels: 1e-05 seconds

Process-2080, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.112031 seconds

Process-2080, Step 6: Misc initializations: 0.0 seconds

Process-2080, Step 7: for loop: 0.005008 seconds

Process-2080All steps time: 0.12854400000000002 seconds

Process-2080, Find cars processing time: 0.128632 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010504 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000998 seconds

Process-2080, Step 3: Get HOG channels: 7e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.16694 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.007436 seconds

Process-2080All steps time: 0.185893 seconds

Process-2080, Find cars processing time: 0.185983 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.009191 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000468 seconds

Process-2080, Step 3: Get HOG channels: 6e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.12495 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.007388 seconds

Process-2080All steps time: 0.1420110000000003 seconds

Process-2080, Find cars processing time: 0.142106 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.010684 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000795 seconds

Process-2080, Step 3: Get HOG channels: 9e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.126109 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.007101 seconds

Process-2080All steps time: 0.144708 seconds

Process-2080, Find cars processing time: 0.14479 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.008518 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.000492 seconds

Process-2080, Step 3: Get HOG channels: 5e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.123285 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.00701 seconds

Process-2080All steps time: 0.139317 seconds

Process-2080, Find cars processing time: 0.13937 seconds

Process-2080, Step 1: Divide with 255, processing time: 0.011117 seconds

Process-2080, Step 2: Resize if scale is not 1: 0.001723 seconds

Process-2080, Step 3: Get HOG channels: 8e-06 seconds

Process-2080, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2080, Step 5: Compute individual channel HOG features for the entire image: 0.127635 seconds

Process-2080, Step 6: Misc initializations: 1e-06 seconds

Process-2080, Step 7: for loop: 0.006324 seconds

Process-2080All steps time: 0.146816 seconds

Process-2080, Find cars processing time: 0.146904 seconds

The times for each task are: [0.304713, 0.25581, 0.2196, 0.268001, 0.246373, 0.221629, 0.211878,

Minimum: 0.128632 Maximum: 0.304713 Average: 0.186 seconds

Number of processes used: 1 window size 260

Length of task list: 22

Number of processes used: 1

Process-2081, Step 1: Divide with 255, processing time: 0.03362 seconds

Process-2081, Step 2: Resize if scale is not 1: 0.00206 seconds

Process-2081, Step 3: Get HOG channels: 2.3e-05 seconds

Process-2081, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.202992 seconds

Process-2081, Step 6: Misc initializations: 2e-06 seconds

Process-2081, Step 7: for loop: 0.019658 seconds

Process-2081All steps time: 0.258361 seconds

Process-2081, Find cars processing time: 0.258538 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.011495 seconds

Process-2081, Step 2: Resize if scale is not 1: 0.000986 seconds

Process-2081, Step 3: Get HOG channels: 9e-06 seconds

Process-2081, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.230725 seconds

Process-2081, Step 6: Misc initializations: 0.0 seconds

Process-2081, Step 7: for loop: 0.01251 seconds

Process-2081All steps time: 0.255734 seconds

Process-2081, Find cars processing time: 0.255836 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010577 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000625 seconds
Process-2081, Step 3: Get HOG channels: 7e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.199111 seconds
Process-2081, Step 6: Misc initializations: 0.0 seconds
Process-2081, Step 7: for loop: 0.012373 seconds

Process-2081All steps time: 0.2227 seconds

Process-2081, Find cars processing time: 0.222775 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.011195 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000933 seconds
Process-2081, Step 3: Get HOG channels: 9e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.244802 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.015033 seconds

Process-2081All steps time: 0.271981 seconds

Process-2081, Find cars processing time: 0.272078 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010249 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000865 seconds
Process-2081, Step 3: Get HOG channels: 8e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.214556 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.012614 seconds

Process-2081All steps time: 0.238302 seconds

Process-2081, Find cars processing time: 0.238384 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010666 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000818 seconds
Process-2081, Step 3: Get HOG channels: 8e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.174333 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.011201 seconds

Process-2081All steps time: 0.197035 seconds

Process-2081, Find cars processing time: 0.197129 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010617 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000836 seconds
Process-2081, Step 3: Get HOG channels: 8e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.205602 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.012605 seconds

Process-2081All steps time: 0.22967800000000002 seconds

Process-2081, Find cars processing time: 0.229763 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010802 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000641 seconds
Process-2081, Step 3: Get HOG channels: 6e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.216321 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.012427 seconds

Process-2081All steps time: 0.240204 seconds

Process-2081, Find cars processing time: 0.24029 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010608 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000608 seconds
Process-2081, Step 3: Get HOG channels: 6e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.165024 seconds
Process-2081, Step 6: Misc initializations: 0.0 seconds
Process-2081, Step 7: for loop: 0.008221 seconds

Process-2081All steps time: 0.184474 seconds

Process-2081, Find cars processing time: 0.184546 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010921 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000782 seconds
Process-2081, Step 3: Get HOG channels: 6e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.139759 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.008271 seconds

Process-2081All steps time: 0.159747 seconds

Process-2081, Find cars processing time: 0.159845 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.009389 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000592 seconds
Process-2081, Step 3: Get HOG channels: 5e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.18783 seconds
Process-2081, Step 6: Misc initializations: 0.0 seconds
Process-2081, Step 7: for loop: 0.009703 seconds

Process-2081All steps time: 0.207527 seconds

Process-2081, Find cars processing time: 0.207592 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010923 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000853 seconds
Process-2081, Step 3: Get HOG channels: 8e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.170054 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.009736 seconds

Process-2081All steps time: 0.191584 seconds

Process-2081, Find cars processing time: 0.191675 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.00873 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000731 seconds
Process-2081, Step 3: Get HOG channels: 7e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.143728 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.006037 seconds

Process-2081All steps time: 0.159242 seconds

Process-2081, Find cars processing time: 0.15932 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.008935 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000727 seconds
Process-2081, Step 3: Get HOG channels: 7e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.126272 seconds
Process-2081, Step 6: Misc initializations: 0.0 seconds
Process-2081, Step 7: for loop: 0.00594 seconds

Process-2081All steps time: 0.141889 seconds

Process-2081, Find cars processing time: 0.14195 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010472 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000621 seconds
Process-2081, Step 3: Get HOG channels: 6e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.117259 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.005627 seconds

Process-2081All steps time: 0.133994 seconds

Process-2081, Find cars processing time: 0.134066 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010815 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000823 seconds
Process-2081, Step 3: Get HOG channels: 9e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.118701 seconds
Process-2081, Step 6: Misc initializations: 0.0 seconds
Process-2081, Step 7: for loop: 0.005738 seconds

Process-2081All steps time: 0.136095 seconds

Process-2081, Find cars processing time: 0.13617 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010621 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000748 seconds
Process-2081, Step 3: Get HOG channels: 9e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.106753 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.004379 seconds

Process-2081All steps time: 0.1225209999999999 seconds

Process-2081, Find cars processing time: 0.122607 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010485 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000886 seconds
Process-2081, Step 3: Get HOG channels: 6e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.108839 seconds
Process-2081, Step 6: Misc initializations: 0.0 seconds
Process-2081, Step 7: for loop: 0.006024 seconds

Process-2081All steps time: 0.126247 seconds

Process-2081, Find cars processing time: 0.126323 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010948 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000624 seconds
Process-2081, Step 3: Get HOG channels: 5e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.102843 seconds
Process-2081, Step 6: Misc initializations: 0.0 seconds
Process-2081, Step 7: for loop: 0.00595 seconds

Process-2081All steps time: 0.120377 seconds

Process-2081, Find cars processing time: 0.12045 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010718 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000561 seconds
Process-2081, Step 3: Get HOG channels: 6e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.126658 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.007179 seconds

Process-2081All steps time: 0.1451299999999998 seconds

Process-2081, Find cars processing time: 0.145198 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.010149 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.000725 seconds
Process-2081, Step 3: Get HOG channels: 8e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.132142 seconds
Process-2081, Step 6: Misc initializations: 1e-06 seconds
Process-2081, Step 7: for loop: 0.011245 seconds

Process-2081All steps time: 0.1542780000000003 seconds

Process-2081, Find cars processing time: 0.154378 seconds

Process-2081, Step 1: Divide with 255, processing time: 0.008845 seconds
Process-2081, Step 2: Resize if scale is not 1: 0.001244 seconds
Process-2081, Step 3: Get HOG channels: 6e-06 seconds
Process-2081, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2081, Step 5: Compute individual channel HOG features for the entire image: 0.113531 seconds
Process-2081, Step 6: Misc initializations: 0.0 seconds
Process-2081, Step 7: for loop: 0.007218 seconds

Process-2081All steps time: 0.13085 seconds

Process-2081, Find cars processing time: 0.130901 seconds

The times for each task are: [0.258538, 0.255836, 0.222775, 0.272078, 0.238384, 0.197129, 0.2297

Minimum: 0.12045 Maximum: 0.272078 Average: 0.1832 seconds

Number of processes used: 1 window size 260

Length of task list: 22

Number of processes used: 1

Process-2082, Step 1: Divide with 255, processing time: 0.035192 seconds

Process-2082, Step 2: Resize if scale is not 1: 0.002493 seconds

Process-2082, Step 3: Get HOG channels: 3.3e-05 seconds

Process-2082, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.249205 seconds

Process-2082, Step 6: Misc initializations: 1e-06 seconds

Process-2082, Step 7: for loop: 0.017597 seconds

Process-2082All steps time: 0.304528 seconds

Process-2082, Find cars processing time: 0.30472 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.009828 seconds

Process-2082, Step 2: Resize if scale is not 1: 0.000727 seconds

Process-2082, Step 3: Get HOG channels: 8e-06 seconds

Process-2082, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.23548 seconds

Process-2082, Step 6: Misc initializations: 0.0 seconds

Process-2082, Step 7: for loop: 0.015936 seconds

Process-2082All steps time: 0.261986 seconds

Process-2082, Find cars processing time: 0.262067 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.010952 seconds

Process-2082, Step 2: Resize if scale is not 1: 0.001006 seconds

Process-2082, Step 3: Get HOG channels: 9e-06 seconds

Process-2082, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.199509 seconds

Process-2082, Step 6: Misc initializations: 1e-06 seconds

Process-2082, Step 7: for loop: 0.013113 seconds

Process-2082All steps time: 0.2245999999999997 seconds

Process-2082, Find cars processing time: 0.224691 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.010517 seconds

Process-2082, Step 2: Resize if scale is not 1: 0.000843 seconds

Process-2082, Step 3: Get HOG channels: 8e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.200756 seconds
Process-2082, Step 6: Misc initializations: 0.0 seconds
Process-2082, Step 7: for loop: 0.01287 seconds

Process-2082All steps time: 0.2250029999999998 seconds

Process-2082, Find cars processing time: 0.225085 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.010083 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000835 seconds
Process-2082, Step 3: Get HOG channels: 8e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.16756 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.009796 seconds

Process-2082All steps time: 0.188292 seconds

Process-2082, Find cars processing time: 0.188373 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.010608 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000872 seconds
Process-2082, Step 3: Get HOG channels: 9e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.153043 seconds
Process-2082, Step 6: Misc initializations: 0.0 seconds
Process-2082, Step 7: for loop: 0.009827 seconds

Process-2082All steps time: 0.1743690000000002 seconds

Process-2082, Find cars processing time: 0.174447 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.009656 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000668 seconds
Process-2082, Step 3: Get HOG channels: 9e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.17252 seconds
Process-2082, Step 6: Misc initializations: 0.0 seconds
Process-2082, Step 7: for loop: 0.00965 seconds

Process-2082All steps time: 0.192513 seconds

Process-2082, Find cars processing time: 0.192597 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.010229 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000675 seconds

Process-2082, Step 3: Get HOG channels: 6e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.152565 seconds
Process-2082, Step 6: Misc initializations: 0.0 seconds
Process-2082, Step 7: for loop: 0.009811 seconds

Process-2082All steps time: 0.173292 seconds

Process-2082, Find cars processing time: 0.173361 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.010533 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.00096 seconds
Process-2082, Step 3: Get HOG channels: 9e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.172018 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.009812 seconds

Process-2082All steps time: 0.193343 seconds

Process-2082, Find cars processing time: 0.19343 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.008657 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000665 seconds
Process-2082, Step 3: Get HOG channels: 9e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.187165 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.009798 seconds

Process-2082All steps time: 0.206305 seconds

Process-2082, Find cars processing time: 0.206388 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.011378 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000875 seconds
Process-2082, Step 3: Get HOG channels: 8e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.179821 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.009931 seconds

Process-2082All steps time: 0.202023 seconds

Process-2082, Find cars processing time: 0.202114 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.010533 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000928 seconds

Process-2082, Step 3: Get HOG channels: 9e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.189461 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.009877 seconds

Process-2082All steps time: 0.2108179999999998 seconds

Process-2082, Find cars processing time: 0.210909 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.008977 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000767 seconds
Process-2082, Step 3: Get HOG channels: 7e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.12716 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.006047 seconds

Process-2082All steps time: 0.1429669999999998 seconds

Process-2082, Find cars processing time: 0.143042 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.011498 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000694 seconds
Process-2082, Step 3: Get HOG channels: 7e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.151638 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.007209 seconds

Process-2082All steps time: 0.1710539999999998 seconds

Process-2082, Find cars processing time: 0.171138 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.01037 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000513 seconds
Process-2082, Step 3: Get HOG channels: 6e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.167581 seconds
Process-2082, Step 6: Misc initializations: 0.0 seconds
Process-2082, Step 7: for loop: 0.008923 seconds

Process-2082All steps time: 0.1873990000000004 seconds

Process-2082, Find cars processing time: 0.187503 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.014321 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000615 seconds

Process-2082, Step 3: Get HOG channels: 6e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.149079 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.009632 seconds

Process-2082All steps time: 0.1736599999999998 seconds

Process-2082, Find cars processing time: 0.173746 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.0137 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000594 seconds
Process-2082, Step 3: Get HOG channels: 8e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.113124 seconds
Process-2082, Step 6: Misc initializations: 0.0 seconds
Process-2082, Step 7: for loop: 0.005061 seconds

Process-2082All steps time: 0.132495 seconds

Process-2082, Find cars processing time: 0.132581 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.007161 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.001123 seconds
Process-2082, Step 3: Get HOG channels: 7e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.133661 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.007405 seconds

Process-2082All steps time: 0.149366 seconds

Process-2082, Find cars processing time: 0.149444 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.009357 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000536 seconds
Process-2082, Step 3: Get HOG channels: 6e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.133011 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.007262 seconds

Process-2082All steps time: 0.1501799999999998 seconds

Process-2082, Find cars processing time: 0.150255 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.008728 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000466 seconds

Process-2082, Step 3: Get HOG channels: 5e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.12548 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.008231 seconds

Process-2082All steps time: 0.142917 seconds

Process-2082, Find cars processing time: 0.142992 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.009428 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.000614 seconds
Process-2082, Step 3: Get HOG channels: 6e-06 seconds
Process-2082, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.128399 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.007242 seconds

Process-2082All steps time: 0.14569700000000002 seconds

Process-2082, Find cars processing time: 0.145777 seconds

Process-2082, Step 1: Divide with 255, processing time: 0.01041 seconds
Process-2082, Step 2: Resize if scale is not 1: 0.002158 seconds
Process-2082, Step 3: Get HOG channels: 1e-05 seconds
Process-2082, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2082, Step 5: Compute individual channel HOG features for the entire image: 0.109697 seconds
Process-2082, Step 6: Misc initializations: 1e-06 seconds
Process-2082, Step 7: for loop: 0.005335 seconds

Process-2082All steps time: 0.12762 seconds

Process-2082, Find cars processing time: 0.127718 seconds

The times for each task are: [0.30472, 0.262067, 0.224691, 0.225085, 0.188373, 0.174447, 0.19259]

Minimum: 0.127718 Maximum: 0.30472 Average: 0.1856 seconds

Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]
1 processes used for testing 5 window sizes
Processing times for each image [8.3986, 11.0777, 10.3461, 10.9286, 10.5558, 10.2554] with an average of 10.3461

Time elapsed so far... 342.7180000000001
#####

Number of processes used: 2 window size 260
Length of task list: 22

Number of processes used: 2

Process-2083, Step 1: Divide with 255, processing time: 0.034416 seconds
Process-2083, Step 2: Resize if scale is not 1: 0.002497 seconds
Process-2083, Step 3: Get HOG channels: 2.7e-05 seconds
Process-2083, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.255904 seconds
Process-2083, Step 6: Misc initializations: 1e-06 seconds
Process-2083, Step 7: for loop: 0.017297 seconds

Process-2083All steps time: 0.310148 seconds

Process-2083, Find cars processing time: 0.310335 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.01099 seconds
Process-2083, Step 2: Resize if scale is not 1: 0.000917 seconds
Process-2083, Step 3: Get HOG channels: 8e-06 seconds
Process-2083, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.239156 seconds
Process-2083, Step 6: Misc initializations: 1e-06 seconds
Process-2083, Step 7: for loop: 0.013157 seconds

Process-2083All steps time: 0.2642389999999995 seconds

Process-2083, Find cars processing time: 0.264322 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.032254 seconds
Process-2084, Step 2: Resize if scale is not 1: 0.002347 seconds
Process-2084, Step 3: Get HOG channels: 2.4e-05 seconds
Process-2084, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.236443 seconds
Process-2084, Step 6: Misc initializations: 1e-06 seconds
Process-2084, Step 7: for loop: 0.018365 seconds

Process-2084All steps time: 0.28944 seconds

Process-2084, Find cars processing time: 0.289616 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.010384 seconds
Process-2084, Step 2: Resize if scale is not 1: 0.000698 seconds
Process-2084, Step 3: Get HOG channels: 8e-06 seconds
Process-2084, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.201115 seconds
Process-2084, Step 6: Misc initializations: 1e-06 seconds
Process-2084, Step 7: for loop: 0.012192 seconds

Process-2084All steps time: 0.224404 seconds

Process-2084, Find cars processing time: 0.22448 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.01055 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.000896 seconds

Process-2083, Step 3: Get HOG channels: 8e-06 seconds

Process-2083, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.183765 seconds

Process-2083, Step 6: Misc initializations: 0.0 seconds

Process-2083, Step 7: for loop: 0.013961 seconds

Process-2083All steps time: 0.209188 seconds

Process-2083, Find cars processing time: 0.209267 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.008723 seconds

Process-2084, Step 2: Resize if scale is not 1: 0.000726 seconds

Process-2084, Step 3: Get HOG channels: 7e-06 seconds

Process-2084, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.152737 seconds

Process-2084, Step 6: Misc initializations: 0.0 seconds

Process-2084, Step 7: for loop: 0.009533 seconds

Process-2084All steps time: 0.17173300000000002 seconds

Process-2084, Find cars processing time: 0.171798 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.011109 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.001257 seconds

Process-2083, Step 3: Get HOG channels: 1.6e-05 seconds

Process-2083, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.197262 seconds

Process-2083, Step 6: Misc initializations: 0.0 seconds

Process-2083, Step 7: for loop: 0.010363 seconds

Process-2083All steps time: 0.220021 seconds

Process-2083, Find cars processing time: 0.220137 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.009776 seconds

Process-2084, Step 2: Resize if scale is not 1: 0.000686 seconds

Process-2084, Step 3: Get HOG channels: 7e-06 seconds

Process-2084, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.182264 seconds

Process-2084, Step 6: Misc initializations: 0.0 seconds

Process-2084, Step 7: for loop: 0.01072 seconds

Process-2084All steps time: 0.20346000000000003 seconds

Process-2084, Find cars processing time: 0.20354 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.009073 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.00075 seconds

Process-2083, Step 3: Get HOG channels: 7e-06 seconds

Process-2083, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.128556 seconds

Process-2083, Step 6: Misc initializations: 0.0 seconds

Process-2083, Step 7: for loop: 0.007643 seconds

Process-2083All steps time: 0.14603700000000003 seconds

Process-2083, Find cars processing time: 0.146113 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.01062 seconds

Process-2084, Step 2: Resize if scale is not 1: 0.000659 seconds

Process-2084, Step 3: Get HOG channels: 5e-06 seconds

Process-2084, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.145039 seconds

Process-2084, Step 6: Misc initializations: 1e-06 seconds

Process-2084, Step 7: for loop: 0.008454 seconds

Process-2084All steps time: 0.1647829999999999 seconds

Process-2084, Find cars processing time: 0.164854 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.010084 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.000658 seconds

Process-2083, Step 3: Get HOG channels: 6e-06 seconds

Process-2083, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.148477 seconds

Process-2083, Step 6: Misc initializations: 0.0 seconds

Process-2083, Step 7: for loop: 0.008285 seconds

Process-2083All steps time: 0.167516 seconds

Process-2083, Find cars processing time: 0.167587 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.010857 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.00088 seconds

Process-2083, Step 3: Get HOG channels: 9e-06 seconds

Process-2083, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.137855 seconds

Process-2083, Step 6: Misc initializations: 1e-06 seconds

Process-2083, Step 7: for loop: 0.005692 seconds

Process-2083All steps time: 0.155304 seconds

Process-2083, Find cars processing time: 0.155399 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.010957 seconds

Process-2084, Step 2: Resize if scale is not 1: 0.000511 seconds

Process-2084, Step 3: Get HOG channels: 5e-06 seconds

Process-2084, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.166559 seconds

Process-2084, Step 6: Misc initializations: 1e-06 seconds

Process-2084, Step 7: for loop: 0.009727 seconds

Process-2084All steps time: 0.18776500000000002 seconds

Process-2084, Find cars processing time: 0.18782 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.00949 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.000729 seconds

Process-2083, Step 3: Get HOG channels: 8e-06 seconds

Process-2083, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.130605 seconds

Process-2083, Step 6: Misc initializations: 1e-06 seconds

Process-2083, Step 7: for loop: 0.006774 seconds

Process-2083All steps time: 0.147615 seconds

Process-2083, Find cars processing time: 0.14769 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.011339 seconds

Process-2084, Step 2: Resize if scale is not 1: 0.001255 seconds

Process-2084, Step 3: Get HOG channels: 1.2e-05 seconds

Process-2084, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.161304 seconds

Process-2084, Step 6: Misc initializations: 1e-06 seconds

Process-2084, Step 7: for loop: 0.007319 seconds

Process-2084All steps time: 0.1812409999999999 seconds

Process-2084, Find cars processing time: 0.181331 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.011357 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.000552 seconds

Process-2083, Step 3: Get HOG channels: 7e-06 seconds

Process-2083, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.142376 seconds

Process-2083, Step 6: Misc initializations: 0.0 seconds

Process-2083, Step 7: for loop: 0.00724 seconds

Process-2083All steps time: 0.161539 seconds

Process-2083, Find cars processing time: 0.161614 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.008585 seconds

Process-2084, Step 2: Resize if scale is not 1: 0.000582 seconds

Process-2084, Step 3: Get HOG channels: 7e-06 seconds

Process-2084, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.110363 seconds

Process-2084, Step 6: Misc initializations: 1e-06 seconds

Process-2084, Step 7: for loop: 0.005331 seconds

Process-2084All steps time: 0.124878 seconds

Process-2084, Find cars processing time: 0.124952 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.010712 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.001318 seconds

Process-2083, Step 3: Get HOG channels: 8e-06 seconds

Process-2083, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.127974 seconds

Process-2083, Step 6: Misc initializations: 1e-06 seconds

Process-2083, Step 7: for loop: 0.007347 seconds

Process-2083All steps time: 0.147368 seconds

Process-2083, Find cars processing time: 0.147449 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.011209 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.00081 seconds

Process-2083, Step 3: Get HOG channels: 8e-06 seconds

Process-2083, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.144205 seconds

Process-2083, Step 6: Misc initializations: 0.0 seconds

Process-2083, Step 7: for loop: 0.007435 seconds

Process-2083All steps time: 0.163677 seconds

Process-2083, Find cars processing time: 0.16377 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.010573 seconds

Process-2084, Step 2: Resize if scale is not 1: 0.001371 seconds

Process-2084, Step 3: Get HOG channels: 8e-06 seconds

Process-2084, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.127013 seconds

Process-2084, Step 6: Misc initializations: 1e-06 seconds

Process-2084, Step 7: for loop: 0.007241 seconds

Process-2084All steps time: 0.1462149999999998 seconds

Process-2084, Find cars processing time: 0.146284 seconds

Process-2083, Step 1: Divide with 255, processing time: 0.010307 seconds

Process-2083, Step 2: Resize if scale is not 1: 0.000582 seconds

Process-2083, Step 3: Get HOG channels: 7e-06 seconds

Process-2083, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2083, Step 5: Compute individual channel HOG features for the entire image: 0.127186 seconds

Process-2083, Step 6: Misc initializations: 1e-06 seconds

Process-2083, Step 7: for loop: 0.00718 seconds

Process-2083All steps time: 0.1452709999999998 seconds

Process-2083, Find cars processing time: 0.145354 seconds

Process-2084, Step 1: Divide with 255, processing time: 0.01069 seconds

Process-2084, Step 2: Resize if scale is not 1: 0.001774 seconds

Process-2084, Step 3: Get HOG channels: 7e-06 seconds

Process-2084, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2084, Step 5: Compute individual channel HOG features for the entire image: 0.127713 seconds

Process-2084, Step 6: Misc initializations: 1e-06 seconds

Process-2084, Step 7: for loop: 0.007039 seconds

Process-2084All steps time: 0.1472319999999997 seconds

Process-2084, Find cars processing time: 0.147339 seconds

The times for each task are: [0.310335, 0.264322, 0.289616, 0.22448, 0.209267, 0.171798, 0.22013

Minimum: 0.124952 Maximum: 0.310335 Average: 0.1855 seconds

Number of processes used: 2 window size 260

Length of task list: 22

Number of processes used: 2

Process-2085, Step 1: Divide with 255, processing time: 0.031359 seconds

Process-2085, Step 2: Resize if scale is not 1: 0.002398 seconds

Process-2085, Step 3: Get HOG channels: 2.3e-05 seconds

Process-2085, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.250415 seconds

Process-2085, Step 6: Misc initializations: 1e-06 seconds

Process-2085, Step 7: for loop: 0.015302 seconds

Process-2085All steps time: 0.2995049999999997 seconds

Process-2085, Find cars processing time: 0.29966 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.01106 seconds

Process-2085, Step 2: Resize if scale is not 1: 0.000465 seconds
Process-2085, Step 3: Get HOG channels: 5e-06 seconds
Process-2085, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.191523 seconds
Process-2085, Step 6: Misc initializations: 1e-06 seconds
Process-2085, Step 7: for loop: 0.012513 seconds

Process-2085All steps time: 0.21557199999999999 seconds

Process-2085, Find cars processing time: 0.215639 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.030054 seconds
Process-2086, Step 2: Resize if scale is not 1: 0.002202 seconds
Process-2086, Step 3: Get HOG channels: 2.4e-05 seconds
Process-2086, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.241756 seconds
Process-2086, Step 6: Misc initializations: 0.0 seconds
Process-2086, Step 7: for loop: 0.017444 seconds

Process-2086All steps time: 0.291486 seconds

Process-2086, Find cars processing time: 0.291632 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.009982 seconds
Process-2085, Step 2: Resize if scale is not 1: 0.000486 seconds
Process-2085, Step 3: Get HOG channels: 5e-06 seconds
Process-2085, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.18965 seconds
Process-2085, Step 6: Misc initializations: 0.0 seconds
Process-2085, Step 7: for loop: 0.012403 seconds

Process-2085All steps time: 0.212533 seconds

Process-2085, Find cars processing time: 0.21261 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.009712 seconds
Process-2086, Step 2: Resize if scale is not 1: 0.000604 seconds
Process-2086, Step 3: Get HOG channels: 5e-06 seconds
Process-2086, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.200797 seconds
Process-2086, Step 6: Misc initializations: 1e-06 seconds
Process-2086, Step 7: for loop: 0.010221 seconds

Process-2086All steps time: 0.22134500000000001 seconds

Process-2086, Find cars processing time: 0.221405 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.009507 seconds

Process-2086, Step 2: Resize if scale is not 1: 0.000788 seconds
Process-2086, Step 3: Get HOG channels: 7e-06 seconds
Process-2086, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.188531 seconds
Process-2086, Step 6: Misc initializations: 0.0 seconds
Process-2086, Step 7: for loop: 0.009915 seconds

Process-2086All steps time: 0.20875700000000003 seconds

Process-2086, Find cars processing time: 0.208827 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.009835 seconds
Process-2085, Step 2: Resize if scale is not 1: 0.000559 seconds
Process-2085, Step 3: Get HOG channels: 5e-06 seconds
Process-2085, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.185827 seconds
Process-2085, Step 6: Misc initializations: 1e-06 seconds
Process-2085, Step 7: for loop: 0.010231 seconds

Process-2085All steps time: 0.20646399999999998 seconds

Process-2085, Find cars processing time: 0.206533 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.00945 seconds
Process-2086, Step 2: Resize if scale is not 1: 0.000594 seconds
Process-2086, Step 3: Get HOG channels: 6e-06 seconds
Process-2086, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.177496 seconds
Process-2086, Step 6: Misc initializations: 0.0 seconds
Process-2086, Step 7: for loop: 0.010007 seconds

Process-2086All steps time: 0.1975599999999999 seconds

Process-2086, Find cars processing time: 0.197631 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.010829 seconds
Process-2085, Step 2: Resize if scale is not 1: 0.000901 seconds
Process-2085, Step 3: Get HOG channels: 8e-06 seconds
Process-2085, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.140749 seconds
Process-2085, Step 6: Misc initializations: 1e-06 seconds
Process-2085, Step 7: for loop: 0.008021 seconds

Process-2085All steps time: 0.16051700000000002 seconds

Process-2085, Find cars processing time: 0.160584 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.010031 seconds

Process-2086, Step 2: Resize if scale is not 1: 0.000832 seconds
Process-2086, Step 3: Get HOG channels: 9e-06 seconds
Process-2086, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.144959 seconds
Process-2086, Step 6: Misc initializations: 1e-06 seconds
Process-2086, Step 7: for loop: 0.008056 seconds

Process-2086All steps time: 0.1638970000000001 seconds

Process-2086, Find cars processing time: 0.163978 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.009705 seconds
Process-2085, Step 2: Resize if scale is not 1: 0.000617 seconds
Process-2085, Step 3: Get HOG channels: 5e-06 seconds
Process-2085, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.140421 seconds
Process-2085, Step 6: Misc initializations: 0.0 seconds
Process-2085, Step 7: for loop: 0.0076 seconds

Process-2085All steps time: 0.158354 seconds

Process-2085, Find cars processing time: 0.158416 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.009768 seconds
Process-2086, Step 2: Resize if scale is not 1: 0.000445 seconds
Process-2086, Step 3: Get HOG channels: 5e-06 seconds
Process-2086, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.124683 seconds
Process-2086, Step 6: Misc initializations: 0.0 seconds
Process-2086, Step 7: for loop: 0.007421 seconds

Process-2086All steps time: 0.142328 seconds

Process-2086, Find cars processing time: 0.142381 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.010107 seconds
Process-2085, Step 2: Resize if scale is not 1: 0.000597 seconds
Process-2085, Step 3: Get HOG channels: 5e-06 seconds
Process-2085, Step 4: Define blocks and steps as above: 2.3e-05 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.140234 seconds
Process-2085, Step 6: Misc initializations: 0.0 seconds
Process-2085, Step 7: for loop: 0.006 seconds

Process-2085All steps time: 0.156966 seconds

Process-2085, Find cars processing time: 0.157046 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.010657 seconds

Process-2086, Step 2: Resize if scale is not 1: 0.000845 seconds
Process-2086, Step 3: Get HOG channels: 7e-06 seconds
Process-2086, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.130006 seconds
Process-2086, Step 6: Misc initializations: 1e-06 seconds
Process-2086, Step 7: for loop: 0.005895 seconds

Process-2086All steps time: 0.1474180000000002 seconds

Process-2086, Find cars processing time: 0.147491 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.010716 seconds
Process-2086, Step 2: Resize if scale is not 1: 0.000556 seconds
Process-2086, Step 3: Get HOG channels: 6e-06 seconds
Process-2086, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.123232 seconds
Process-2086, Step 6: Misc initializations: 0.0 seconds
Process-2086, Step 7: for loop: 0.006051 seconds

Process-2086All steps time: 0.140567 seconds

Process-2086, Find cars processing time: 0.140642 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.009318 seconds
Process-2085, Step 2: Resize if scale is not 1: 0.000705 seconds
Process-2085, Step 3: Get HOG channels: 7e-06 seconds
Process-2085, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.143908 seconds
Process-2085, Step 6: Misc initializations: 0.0 seconds
Process-2085, Step 7: for loop: 0.00729 seconds

Process-2085All steps time: 0.161235 seconds

Process-2085, Find cars processing time: 0.161304 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.010228 seconds
Process-2086, Step 2: Resize if scale is not 1: 0.000671 seconds
Process-2086, Step 3: Get HOG channels: 8e-06 seconds
Process-2086, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.085913 seconds
Process-2086, Step 6: Misc initializations: 0.0 seconds
Process-2086, Step 7: for loop: 0.00377 seconds

Process-2086All steps time: 0.1005979999999999 seconds

Process-2086, Find cars processing time: 0.100663 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.009211 seconds

Process-2085, Step 2: Resize if scale is not 1: 0.001208 seconds
Process-2085, Step 3: Get HOG channels: 7e-06 seconds
Process-2085, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.127547 seconds
Process-2085, Step 6: Misc initializations: 1e-06 seconds
Process-2085, Step 7: for loop: 0.007097 seconds

Process-2085All steps time: 0.1450779999999998 seconds

Process-2085, Find cars processing time: 0.145138 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.00907 seconds
Process-2086, Step 2: Resize if scale is not 1: 0.001103 seconds
Process-2086, Step 3: Get HOG channels: 1e-05 seconds
Process-2086, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.107995 seconds
Process-2086, Step 6: Misc initializations: 1e-06 seconds
Process-2086, Step 7: for loop: 0.005781 seconds

Process-2086All steps time: 0.1239689999999998 seconds

Process-2086, Find cars processing time: 0.124061 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.011827 seconds
Process-2085, Step 2: Resize if scale is not 1: 0.000815 seconds
Process-2085, Step 3: Get HOG channels: 1.6e-05 seconds
Process-2085, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.128566 seconds
Process-2085, Step 6: Misc initializations: 1e-06 seconds
Process-2085, Step 7: for loop: 0.007046 seconds

Process-2085All steps time: 0.1482780000000002 seconds

Process-2085, Find cars processing time: 0.148345 seconds

Process-2086, Step 1: Divide with 255, processing time: 0.010691 seconds
Process-2086, Step 2: Resize if scale is not 1: 0.000866 seconds
Process-2086, Step 3: Get HOG channels: 1e-05 seconds
Process-2086, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2086, Step 5: Compute individual channel HOG features for the entire image: 0.104062 seconds
Process-2086, Step 6: Misc initializations: 0.0 seconds
Process-2086, Step 7: for loop: 0.005968 seconds

Process-2086All steps time: 0.121607 seconds

Process-2086, Find cars processing time: 0.121703 seconds

Process-2085, Step 1: Divide with 255, processing time: 0.0102 seconds

Process-2085, Step 2: Resize if scale is not 1: 0.001578 seconds
Process-2085, Step 3: Get HOG channels: 8e-06 seconds
Process-2085, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2085, Step 5: Compute individual channel HOG features for the entire image: 0.125227 seconds
Process-2085, Step 6: Misc initializations: 0.0 seconds
Process-2085, Step 7: for loop: 0.006033 seconds

Process-2085All steps time: 0.14305500000000002 seconds

Process-2085, Find cars processing time: 0.143124 seconds

The times for each task are: [0.29966, 0.215639, 0.291632, 0.21261, 0.221405, 0.208827, 0.206533]

Minimum: 0.100663 Maximum: 0.29966 Average: 0.1759 seconds

Number of processes used: 2 window size 260
Length of task list: 22
Number of processes used: 2

Process-2087, Step 1: Divide with 255, processing time: 0.031811 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.002477 seconds
Process-2087, Step 3: Get HOG channels: 2.3e-05 seconds
Process-2087, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.248521 seconds
Process-2087, Step 6: Misc initializations: 0.0 seconds
Process-2087, Step 7: for loop: 0.017238 seconds

Process-2087All steps time: 0.3000759999999995 seconds

Process-2087, Find cars processing time: 0.300238 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.031101 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.002229 seconds
Process-2088, Step 3: Get HOG channels: 1.7e-05 seconds
Process-2088, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.240882 seconds
Process-2088, Step 6: Misc initializations: 1e-06 seconds
Process-2088, Step 7: for loop: 0.013782 seconds

Process-2088All steps time: 0.288017 seconds

Process-2088, Find cars processing time: 0.288147 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.010476 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.000892 seconds
Process-2087, Step 3: Get HOG channels: 8e-06 seconds
Process-2087, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.197101 seconds
Process-2087, Step 6: Misc initializations: 1e-06 seconds
Process-2087, Step 7: for loop: 0.012835 seconds

Process-2087All steps time: 0.22132200000000002 seconds

Process-2087, Find cars processing time: 0.221395 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.01013 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.000849 seconds
Process-2088, Step 3: Get HOG channels: 7e-06 seconds
Process-2088, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.190986 seconds
Process-2088, Step 6: Misc initializations: 0.0 seconds
Process-2088, Step 7: for loop: 0.01252 seconds

Process-2088All steps time: 0.214499 seconds

Process-2088, Find cars processing time: 0.214563 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.00521 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.000469 seconds
Process-2088, Step 3: Get HOG channels: 4e-06 seconds
Process-2088, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.170589 seconds
Process-2088, Step 6: Misc initializations: 0.0 seconds
Process-2088, Step 7: for loop: 0.010685 seconds

Process-2088All steps time: 0.186962 seconds

Process-2088, Find cars processing time: 0.187012 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.008111 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.000705 seconds
Process-2087, Step 3: Get HOG channels: 6e-06 seconds
Process-2087, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.170343 seconds
Process-2087, Step 6: Misc initializations: 2e-06 seconds
Process-2087, Step 7: for loop: 0.010953 seconds

Process-2087All steps time: 0.190126 seconds

Process-2087, Find cars processing time: 0.190234 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.010202 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.000844 seconds
Process-2088, Step 3: Get HOG channels: 9e-06 seconds
Process-2088, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.162892 seconds
Process-2088, Step 6: Misc initializations: 1e-06 seconds
Process-2088, Step 7: for loop: 0.010288 seconds

Process-2088All steps time: 0.184244 seconds

Process-2088, Find cars processing time: 0.184329 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.011099 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.001035 seconds
Process-2087, Step 3: Get HOG channels: 1.9e-05 seconds
Process-2087, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.155687 seconds
Process-2087, Step 6: Misc initializations: 0.0 seconds
Process-2087, Step 7: for loop: 0.009878 seconds

Process-2087All steps time: 0.177725 seconds

Process-2087, Find cars processing time: 0.177827 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.011037 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.000737 seconds
Process-2087, Step 3: Get HOG channels: 6e-06 seconds
Process-2087, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.135112 seconds
Process-2087, Step 6: Misc initializations: 1e-06 seconds
Process-2087, Step 7: for loop: 0.007876 seconds

Process-2087All steps time: 0.154775 seconds

Process-2087, Find cars processing time: 0.154851 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.010172 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.000905 seconds
Process-2088, Step 3: Get HOG channels: 8e-06 seconds
Process-2088, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.137731 seconds
Process-2088, Step 6: Misc initializations: 0.0 seconds
Process-2088, Step 7: for loop: 0.008143 seconds

Process-2088All steps time: 0.156966 seconds

Process-2088, Find cars processing time: 0.157053 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.009216 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.000683 seconds
Process-2087, Step 3: Get HOG channels: 6e-06 seconds
Process-2087, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.141986 seconds
Process-2087, Step 6: Misc initializations: 1e-06 seconds
Process-2087, Step 7: for loop: 0.008442 seconds

Process-2087All steps time: 0.16034 seconds

Process-2087, Find cars processing time: 0.160401 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.022097 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.000532 seconds
Process-2087, Step 3: Get HOG channels: 5e-06 seconds
Process-2087, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.110276 seconds
Process-2087, Step 6: Misc initializations: 0.0 seconds
Process-2087, Step 7: for loop: 0.00664 seconds

Process-2087All steps time: 0.139555 seconds

Process-2087, Find cars processing time: 0.139615 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.011194 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.000726 seconds
Process-2088, Step 3: Get HOG channels: 4e-06 seconds
Process-2088, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.130641 seconds
Process-2088, Step 6: Misc initializations: 0.0 seconds
Process-2088, Step 7: for loop: 0.007797 seconds

Process-2088All steps time: 0.150366 seconds

Process-2088, Find cars processing time: 0.150421 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.022653 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.000425 seconds
Process-2087, Step 3: Get HOG channels: 4e-06 seconds
Process-2087, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.122514 seconds
Process-2087, Step 6: Misc initializations: 1e-06 seconds
Process-2087, Step 7: for loop: 0.008502 seconds

Process-2087All steps time: 0.15410400000000002 seconds

Process-2087, Find cars processing time: 0.154161 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.010642 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.000961 seconds
Process-2087, Step 3: Get HOG channels: 1.1e-05 seconds
Process-2087, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.126589 seconds
Process-2087, Step 6: Misc initializations: 1e-06 seconds
Process-2087, Step 7: for loop: 0.006291 seconds

Process-2087All steps time: 0.144504 seconds

Process-2087, Find cars processing time: 0.144597 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.015639 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.000747 seconds
Process-2088, Step 3: Get HOG channels: 8e-06 seconds
Process-2088, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.148444 seconds
Process-2088, Step 6: Misc initializations: 1e-06 seconds
Process-2088, Step 7: for loop: 0.007996 seconds

Process-2088All steps time: 0.172838 seconds

Process-2088, Find cars processing time: 0.172918 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.014983 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.000833 seconds
Process-2088, Step 3: Get HOG channels: 5e-06 seconds
Process-2088, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.127792 seconds
Process-2088, Step 6: Misc initializations: 0.0 seconds
Process-2088, Step 7: for loop: 0.008893 seconds

Process-2088All steps time: 0.152513 seconds

Process-2088, Find cars processing time: 0.152591 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.010337 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.001817 seconds
Process-2087, Step 3: Get HOG channels: 8e-06 seconds
Process-2087, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.116254 seconds
Process-2087, Step 6: Misc initializations: 1e-06 seconds
Process-2087, Step 7: for loop: 0.005142 seconds

Process-2087All steps time: 0.133567 seconds

Process-2087, Find cars processing time: 0.133643 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.008758 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.000477 seconds
Process-2088, Step 3: Get HOG channels: 5e-06 seconds
Process-2088, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.129684 seconds
Process-2088, Step 6: Misc initializations: 1e-06 seconds
Process-2088, Step 7: for loop: 0.007376 seconds

Process-2088All steps time: 0.146306 seconds

Process-2088, Find cars processing time: 0.146363 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.010247 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.00132 seconds
Process-2087, Step 3: Get HOG channels: 8e-06 seconds
Process-2087, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.12884 seconds
Process-2087, Step 6: Misc initializations: 1e-06 seconds
Process-2087, Step 7: for loop: 0.007446 seconds

Process-2087All steps time: 0.1478700000000003 seconds

Process-2087, Find cars processing time: 0.147947 seconds

Process-2088, Step 1: Divide with 255, processing time: 0.010853 seconds
Process-2088, Step 2: Resize if scale is not 1: 0.001067 seconds
Process-2088, Step 3: Get HOG channels: 1.4e-05 seconds
Process-2088, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-2088, Step 5: Compute individual channel HOG features for the entire image: 0.13429 seconds
Process-2088, Step 6: Misc initializations: 1e-06 seconds
Process-2088, Step 7: for loop: 0.007345 seconds

Process-2088All steps time: 0.153582 seconds

Process-2088, Find cars processing time: 0.153688 seconds

Process-2087, Step 1: Divide with 255, processing time: 0.010005 seconds
Process-2087, Step 2: Resize if scale is not 1: 0.001233 seconds
Process-2087, Step 3: Get HOG channels: 5e-06 seconds
Process-2087, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2087, Step 5: Compute individual channel HOG features for the entire image: 0.111669 seconds
Process-2087, Step 6: Misc initializations: 1e-06 seconds
Process-2087, Step 7: for loop: 0.00624 seconds

Process-2087All steps time: 0.129158 seconds

Process-2087, Find cars processing time: 0.129226 seconds

The times for each task are: [0.300238, 0.288147, 0.221395, 0.214563, 0.187012, 0.190234, 0.1843]

Minimum: 0.129226 Maximum: 0.300238 Average: 0.1755 seconds

```
*****
Number of processes used: 2 window size 260
Length of task list: 22
Number of processes used: 2

Process-2089, Step 1: Divide with 255, processing time: 0.034567 seconds
Process-2089, Step 2: Resize if scale is not 1: 0.002576 seconds
Process-2089, Step 3: Get HOG channels: 2.2e-05 seconds
Process-2089, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.24679 seconds
Process-2089, Step 6: Misc initializations: 2e-06 seconds
Process-2089, Step 7: for loop: 0.016987 seconds

Process-2089All steps time: 0.30095099999999997 seconds

Process-2089, Find cars processing time: 0.301126 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.035536 seconds
Process-2090, Step 2: Resize if scale is not 1: 0.002436 seconds
Process-2090, Step 3: Get HOG channels: 2.5e-05 seconds
Process-2090, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.212876 seconds
Process-2090, Step 6: Misc initializations: 0.0 seconds
Process-2090, Step 7: for loop: 0.014683 seconds

Process-2090All steps time: 0.265564 seconds

Process-2090, Find cars processing time: 0.265832 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.011334 seconds
Process-2089, Step 2: Resize if scale is not 1: 0.001027 seconds
Process-2089, Step 3: Get HOG channels: 8e-06 seconds
Process-2089, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.241781 seconds
Process-2089, Step 6: Misc initializations: 1e-06 seconds
Process-2089, Step 7: for loop: 0.015949 seconds

Process-2089All steps time: 0.270109 seconds

Process-2089, Find cars processing time: 0.270178 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.009145 seconds
Process-2089, Step 2: Resize if scale is not 1: 0.001024 seconds
Process-2089, Step 3: Get HOG channels: 1.3e-05 seconds
Process-2089, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.17342 seconds
Process-2089, Step 6: Misc initializations: 0.0 seconds
Process-2089, Step 7: for loop: 0.012746 seconds
```

Process-2089All steps time: 0.196358 seconds

Process-2089, Find cars processing time: 0.196464 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.010691 seconds

Process-2090, Step 2: Resize if scale is not 1: 0.000948 seconds

Process-2090, Step 3: Get HOG channels: 7e-06 seconds

Process-2090, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.237282 seconds

Process-2090, Step 6: Misc initializations: 1e-06 seconds

Process-2090, Step 7: for loop: 0.015895 seconds

Process-2090All steps time: 0.264831 seconds

Process-2090, Find cars processing time: 0.264902 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.010429 seconds

Process-2090, Step 2: Resize if scale is not 1: 0.000882 seconds

Process-2090, Step 3: Get HOG channels: 8e-06 seconds

Process-2090, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.161328 seconds

Process-2090, Step 6: Misc initializations: 0.0 seconds

Process-2090, Step 7: for loop: 0.009572 seconds

Process-2090All steps time: 0.182228 seconds

Process-2090, Find cars processing time: 0.1823 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.017826 seconds

Process-2089, Step 2: Resize if scale is not 1: 0.000844 seconds

Process-2089, Step 3: Get HOG channels: 6e-06 seconds

Process-2089, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.200301 seconds

Process-2089, Step 6: Misc initializations: 1e-06 seconds

Process-2089, Step 7: for loop: 0.012782 seconds

Process-2089All steps time: 0.231767 seconds

Process-2089, Find cars processing time: 0.231844 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.010307 seconds

Process-2090, Step 2: Resize if scale is not 1: 0.000816 seconds

Process-2090, Step 3: Get HOG channels: 8e-06 seconds

Process-2090, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.20202 seconds

Process-2090, Step 6: Misc initializations: 1e-06 seconds

Process-2090, Step 7: for loop: 0.011378 seconds

Process-2090All steps time: 0.22453700000000001 seconds

Process-2090, Find cars processing time: 0.224606 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.010205 seconds

Process-2089, Step 2: Resize if scale is not 1: 0.000747 seconds

Process-2089, Step 3: Get HOG channels: 5e-06 seconds

Process-2089, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.157216 seconds

Process-2089, Step 6: Misc initializations: 0.0 seconds

Process-2089, Step 7: for loop: 0.007531 seconds

Process-2089All steps time: 0.175709 seconds

Process-2089, Find cars processing time: 0.17577 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.010251 seconds

Process-2089, Step 2: Resize if scale is not 1: 0.000834 seconds

Process-2089, Step 3: Get HOG channels: 8e-06 seconds

Process-2089, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.128379 seconds

Process-2089, Step 6: Misc initializations: 0.0 seconds

Process-2089, Step 7: for loop: 0.007349 seconds

Process-2089All steps time: 0.146831 seconds

Process-2089, Find cars processing time: 0.146898 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.016881 seconds

Process-2090, Step 2: Resize if scale is not 1: 0.000588 seconds

Process-2090, Step 3: Get HOG channels: 6e-06 seconds

Process-2090, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.170849 seconds

Process-2090, Step 6: Misc initializations: 0.0 seconds

Process-2090, Step 7: for loop: 0.007886 seconds

Process-2090All steps time: 0.196217 seconds

Process-2090, Find cars processing time: 0.19628 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.009559 seconds

Process-2090, Step 2: Resize if scale is not 1: 0.000757 seconds

Process-2090, Step 3: Get HOG channels: 8e-06 seconds

Process-2090, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.112995 seconds

Process-2090, Step 6: Misc initializations: 0.0 seconds

Process-2090, Step 7: for loop: 0.005842 seconds

Process-2090All steps time: 0.1291689999999998 seconds

Process-2090, Find cars processing time: 0.129232 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.010558 seconds

Process-2089, Step 2: Resize if scale is not 1: 0.000446 seconds

Process-2089, Step 3: Get HOG channels: 5e-06 seconds

Process-2089, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.134351 seconds

Process-2089, Step 6: Misc initializations: 1e-06 seconds

Process-2089, Step 7: for loop: 0.008206 seconds

Process-2089All steps time: 0.153572 seconds

Process-2089, Find cars processing time: 0.153626 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.010771 seconds

Process-2090, Step 2: Resize if scale is not 1: 0.000599 seconds

Process-2090, Step 3: Get HOG channels: 6e-06 seconds

Process-2090, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.117669 seconds

Process-2090, Step 6: Misc initializations: 0.0 seconds

Process-2090, Step 7: for loop: 0.005865 seconds

Process-2090All steps time: 0.134917 seconds

Process-2090, Find cars processing time: 0.134996 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.007923 seconds

Process-2089, Step 2: Resize if scale is not 1: 0.000645 seconds

Process-2089, Step 3: Get HOG channels: 6e-06 seconds

Process-2089, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.118233 seconds

Process-2089, Step 6: Misc initializations: 1e-06 seconds

Process-2089, Step 7: for loop: 0.005962 seconds

Process-2089All steps time: 0.132776 seconds

Process-2089, Find cars processing time: 0.132843 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.010493 seconds

Process-2089, Step 2: Resize if scale is not 1: 0.000991 seconds

Process-2089, Step 3: Get HOG channels: 9e-06 seconds

Process-2089, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.122386 seconds

Process-2089, Step 6: Misc initializations: 0.0 seconds

Process-2089, Step 7: for loop: 0.005102 seconds

Process-2089All steps time: 0.13899 seconds

Process-2089, Find cars processing time: 0.139069 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.010506 seconds

Process-2090, Step 2: Resize if scale is not 1: 0.000533 seconds

Process-2090, Step 3: Get HOG channels: 6e-06 seconds

Process-2090, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.12295 seconds

Process-2090, Step 6: Misc initializations: 0.0 seconds

Process-2090, Step 7: for loop: 0.00584 seconds

Process-2090All steps time: 0.13984200000000002 seconds

Process-2090, Find cars processing time: 0.139915 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.010562 seconds

Process-2089, Step 2: Resize if scale is not 1: 0.001151 seconds

Process-2089, Step 3: Get HOG channels: 8e-06 seconds

Process-2089, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.125644 seconds

Process-2089, Step 6: Misc initializations: 0.0 seconds

Process-2089, Step 7: for loop: 0.007129 seconds

Process-2089All steps time: 0.144502 seconds

Process-2089, Find cars processing time: 0.144563 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.006094 seconds

Process-2090, Step 2: Resize if scale is not 1: 0.00118 seconds

Process-2090, Step 3: Get HOG channels: 8e-06 seconds

Process-2090, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.106189 seconds

Process-2090, Step 6: Misc initializations: 1e-06 seconds

Process-2090, Step 7: for loop: 0.005912 seconds

Process-2090All steps time: 0.11939200000000001 seconds

Process-2090, Find cars processing time: 0.119456 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.010445 seconds

Process-2089, Step 2: Resize if scale is not 1: 0.000815 seconds

Process-2089, Step 3: Get HOG channels: 7e-06 seconds

Process-2089, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.12704 seconds

Process-2089, Step 6: Misc initializations: 1e-06 seconds

Process-2089, Step 7: for loop: 0.007058 seconds

Process-2089All steps time: 0.145373 seconds

Process-2089, Find cars processing time: 0.14544 seconds

Process-2090, Step 1: Divide with 255, processing time: 0.005975 seconds

Process-2090, Step 2: Resize if scale is not 1: 0.000417 seconds

Process-2090, Step 3: Get HOG channels: 5e-06 seconds

Process-2090, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2090, Step 5: Compute individual channel HOG features for the entire image: 0.100517 seconds

Process-2090, Step 6: Misc initializations: 1e-06 seconds

Process-2090, Step 7: for loop: 0.006312 seconds

Process-2090All steps time: 0.113232 seconds

Process-2090, Find cars processing time: 0.113278 seconds

Process-2089, Step 1: Divide with 255, processing time: 0.010731 seconds

Process-2089, Step 2: Resize if scale is not 1: 0.001436 seconds

Process-2089, Step 3: Get HOG channels: 7e-06 seconds

Process-2089, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2089, Step 5: Compute individual channel HOG features for the entire image: 0.114976 seconds

Process-2089, Step 6: Misc initializations: 1e-06 seconds

Process-2089, Step 7: for loop: 0.006122 seconds

Process-2089All steps time: 0.1332819999999998 seconds

Process-2089, Find cars processing time: 0.133368 seconds

The times for each task are: [0.301126, 0.265832, 0.270178, 0.196464, 0.264902, 0.1823, 0.231844]

Minimum: 0.113278 Maximum: 0.301126 Average: 0.1792 seconds

Number of processes used: 2 window size 260

Length of task list: 22

Number of processes used: 2

Process-2091, Step 1: Divide with 255, processing time: 0.0346 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.002619 seconds

Process-2091, Step 3: Get HOG channels: 3e-05 seconds

Process-2091, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.255298 seconds

Process-2091, Step 6: Misc initializations: 1e-06 seconds

Process-2091, Step 7: for loop: 0.017571 seconds

Process-2091All steps time: 0.310125 seconds

Process-2091, Find cars processing time: 0.310319 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.032897 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.002131 seconds

Process-2092, Step 3: Get HOG channels: 2.7e-05 seconds

Process-2092, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.226181 seconds

Process-2092, Step 6: Misc initializations: 1e-06 seconds

Process-2092, Step 7: for loop: 0.015359 seconds

Process-2092All steps time: 0.2766019999999996 seconds

Process-2092, Find cars processing time: 0.276787 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.00944 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.000847 seconds

Process-2091, Step 3: Get HOG channels: 1.6e-05 seconds

Process-2091, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.234987 seconds

Process-2091, Step 6: Misc initializations: 1e-06 seconds

Process-2091, Step 7: for loop: 0.016136 seconds

Process-2091All steps time: 0.261434 seconds

Process-2091, Find cars processing time: 0.261508 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.012136 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.000683 seconds

Process-2091, Step 3: Get HOG channels: 7e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.189829 seconds

Process-2091, Step 6: Misc initializations: 1e-06 seconds

Process-2091, Step 7: for loop: 0.010003 seconds

Process-2091All steps time: 0.2126660000000002 seconds

Process-2091, Find cars processing time: 0.212742 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.009143 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.000782 seconds

Process-2092, Step 3: Get HOG channels: 8e-06 seconds

Process-2092, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.227991 seconds

Process-2092, Step 6: Misc initializations: 1e-06 seconds

Process-2092, Step 7: for loop: 0.012866 seconds

Process-2092All steps time: 0.250798 seconds

Process-2092, Find cars processing time: 0.250867 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.010893 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.000587 seconds

Process-2091, Step 3: Get HOG channels: 6e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.157419 seconds

Process-2091, Step 6: Misc initializations: 1e-06 seconds

Process-2091, Step 7: for loop: 0.010018 seconds

Process-2091All steps time: 0.178931 seconds

Process-2091, Find cars processing time: 0.179001 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.0102 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.000925 seconds

Process-2092, Step 3: Get HOG channels: 9e-06 seconds

Process-2092, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.190577 seconds

Process-2092, Step 6: Misc initializations: 1e-06 seconds

Process-2092, Step 7: for loop: 0.010692 seconds

Process-2092All steps time: 0.21241200000000002 seconds

Process-2092, Find cars processing time: 0.212489 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.007645 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.000708 seconds

Process-2092, Step 3: Get HOG channels: 1.9e-05 seconds

Process-2092, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.168213 seconds

Process-2092, Step 6: Misc initializations: 1e-06 seconds

Process-2092, Step 7: for loop: 0.008395 seconds

Process-2092All steps time: 0.18499 seconds

Process-2092, Find cars processing time: 0.185071 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.010077 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.000682 seconds

Process-2091, Step 3: Get HOG channels: 6e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.151766 seconds

Process-2091, Step 6: Misc initializations: 0.0 seconds

Process-2091, Step 7: for loop: 0.009417 seconds

Process-2091All steps time: 0.17195500000000002 seconds

Process-2091, Find cars processing time: 0.172017 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.010813 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.000543 seconds

Process-2092, Step 3: Get HOG channels: 8e-06 seconds

Process-2092, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.150975 seconds

Process-2092, Step 6: Misc initializations: 1e-06 seconds

Process-2092, Step 7: for loop: 0.016597 seconds

Process-2092All steps time: 0.178945 seconds

Process-2092, Find cars processing time: 0.179031 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.010073 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.002096 seconds

Process-2091, Step 3: Get HOG channels: 7e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.17371 seconds

Process-2091, Step 6: Misc initializations: 0.0 seconds

Process-2091, Step 7: for loop: 0.009552 seconds

Process-2091All steps time: 0.195444 seconds

Process-2091, Find cars processing time: 0.195527 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.007181 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.000608 seconds

Process-2091, Step 3: Get HOG channels: 6e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.108705 seconds

Process-2091, Step 6: Misc initializations: 1e-06 seconds

Process-2091, Step 7: for loop: 0.005766 seconds

Process-2091All steps time: 0.1222719999999999 seconds

Process-2091, Find cars processing time: 0.122319 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.010177 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.000484 seconds

Process-2092, Step 3: Get HOG channels: 5e-06 seconds

Process-2092, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.162133 seconds

Process-2092, Step 6: Misc initializations: 1e-06 seconds

Process-2092, Step 7: for loop: 0.007892 seconds

Process-2092All steps time: 0.1806970000000002 seconds

Process-2092, Find cars processing time: 0.180752 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.022826 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.000478 seconds

Process-2091, Step 3: Get HOG channels: 6e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.108845 seconds

Process-2091, Step 6: Misc initializations: 1e-06 seconds

Process-2091, Step 7: for loop: 0.005545 seconds

Process-2091All steps time: 0.137707 seconds

Process-2091, Find cars processing time: 0.137774 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.010249 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.000897 seconds

Process-2092, Step 3: Get HOG channels: 8e-06 seconds

Process-2092, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.125252 seconds

Process-2092, Step 6: Misc initializations: 0.0 seconds

Process-2092, Step 7: for loop: 0.006032 seconds

Process-2092All steps time: 0.14244600000000002 seconds

Process-2092, Find cars processing time: 0.142511 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.010837 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.000725 seconds

Process-2091, Step 3: Get HOG channels: 6e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.140176 seconds

Process-2091, Step 6: Misc initializations: 0.0 seconds

Process-2091, Step 7: for loop: 0.006967 seconds

Process-2091All steps time: 0.158717 seconds

Process-2091, Find cars processing time: 0.158783 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.010605 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.000725 seconds

Process-2092, Step 3: Get HOG channels: 9e-06 seconds

Process-2092, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.099253 seconds

Process-2092, Step 6: Misc initializations: 1e-06 seconds

Process-2092, Step 7: for loop: 0.004014 seconds

Process-2092All steps time: 0.114616 seconds

Process-2092, Find cars processing time: 0.114684 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.010043 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.00106 seconds

Process-2091, Step 3: Get HOG channels: 6e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.118067 seconds

Process-2091, Step 6: Misc initializations: 0.0 seconds

Process-2091, Step 7: for loop: 0.006148 seconds

Process-2091All steps time: 0.13533099999999998 seconds

Process-2091, Find cars processing time: 0.135401 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.010103 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.000933 seconds

Process-2092, Step 3: Get HOG channels: 7e-06 seconds

Process-2092, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.104205 seconds

Process-2092, Step 6: Misc initializations: 0.0 seconds

Process-2092, Step 7: for loop: 0.006076 seconds

Process-2092All steps time: 0.12133100000000001 seconds

Process-2092, Find cars processing time: 0.121405 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.010309 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.000571 seconds

Process-2091, Step 3: Get HOG channels: 6e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.103371 seconds

Process-2091, Step 6: Misc initializations: 1e-06 seconds

Process-2091, Step 7: for loop: 0.005661 seconds

Process-2091All steps time: 0.119925 seconds

Process-2091, Find cars processing time: 0.119988 seconds

Process-2091, Step 1: Divide with 255, processing time: 0.012738 seconds

Process-2091, Step 2: Resize if scale is not 1: 0.001217 seconds

Process-2091, Step 3: Get HOG channels: 6e-06 seconds

Process-2091, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2091, Step 5: Compute individual channel HOG features for the entire image: 0.08872 seconds

Process-2091, Step 6: Misc initializations: 0.0 seconds

Process-2091, Step 7: for loop: 0.005017 seconds

Process-2091All steps time: 0.107704 seconds

Process-2091, Find cars processing time: 0.107771 seconds

Process-2092, Step 1: Divide with 255, processing time: 0.010816 seconds

Process-2092, Step 2: Resize if scale is not 1: 0.000831 seconds

Process-2092, Step 3: Get HOG channels: 9e-06 seconds

Process-2092, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2092, Step 5: Compute individual channel HOG features for the entire image: 0.09898 seconds

Process-2092, Step 6: Misc initializations: 1e-06 seconds

Process-2092, Step 7: for loop: 0.005778 seconds

Process-2092All steps time: 0.116425 seconds

Process-2092, Find cars processing time: 0.116491 seconds

The times for each task are: [0.310319, 0.276787, 0.261508, 0.212742, 0.250867, 0.179001, 0.2124

Minimum: 0.107771 Maximum: 0.310319 Average: 0.177 seconds

Number of processes used: 2 window size 260

Length of task list: 22

Number of processes used: 2

Process-2094, Step 1: Divide with 255, processing time: 0.024767 seconds

Process-2094, Step 2: Resize if scale is not 1: 0.002029 seconds

Process-2094, Step 3: Get HOG channels: 2.1e-05 seconds

Process-2094, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.190944 seconds

Process-2094, Step 6: Misc initializations: 1e-06 seconds

Process-2094, Step 7: for loop: 0.014369 seconds

Process-2094All steps time: 0.232136 seconds

Process-2094, Find cars processing time: 0.23227 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.033874 seconds

Process-2093, Step 2: Resize if scale is not 1: 0.002242 seconds

Process-2093, Step 3: Get HOG channels: 2.2e-05 seconds

Process-2093, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.228319 seconds

Process-2093, Step 6: Misc initializations: 1e-06 seconds

Process-2093, Step 7: for loop: 0.014286 seconds

Process-2093All steps time: 0.2787489999999997 seconds

Process-2093, Find cars processing time: 0.278903 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.010313 seconds

Process-2093, Step 2: Resize if scale is not 1: 0.000878 seconds
Process-2093, Step 3: Get HOG channels: 9e-06 seconds
Process-2093, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.182728 seconds
Process-2093, Step 6: Misc initializations: 0.0 seconds
Process-2093, Step 7: for loop: 0.012612 seconds

Process-2093All steps time: 0.206548 seconds

Process-2093, Find cars processing time: 0.206611 seconds

Process-2094, Step 1: Divide with 255, processing time: 0.010618 seconds
Process-2094, Step 2: Resize if scale is not 1: 0.000927 seconds
Process-2094, Step 3: Get HOG channels: 8e-06 seconds
Process-2094, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.228585 seconds
Process-2094, Step 6: Misc initializations: 0.0 seconds
Process-2094, Step 7: for loop: 0.012359 seconds

Process-2094All steps time: 0.252505 seconds

Process-2094, Find cars processing time: 0.252574 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.009949 seconds
Process-2093, Step 2: Resize if scale is not 1: 0.000935 seconds
Process-2093, Step 3: Get HOG channels: 5e-06 seconds
Process-2093, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.15692 seconds
Process-2093, Step 6: Misc initializations: 0.0 seconds
Process-2093, Step 7: for loop: 0.009972 seconds

Process-2093All steps time: 0.177787 seconds

Process-2093, Find cars processing time: 0.177851 seconds

Process-2094, Step 1: Divide with 255, processing time: 0.007096 seconds
Process-2094, Step 2: Resize if scale is not 1: 0.000582 seconds
Process-2094, Step 3: Get HOG channels: 5e-06 seconds
Process-2094, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.198489 seconds
Process-2094, Step 6: Misc initializations: 1e-06 seconds
Process-2094, Step 7: for loop: 0.012363 seconds

Process-2094All steps time: 0.218541 seconds

Process-2094, Find cars processing time: 0.218602 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.010276 seconds

Process-2093, Step 2: Resize if scale is not 1: 0.000669 seconds
Process-2093, Step 3: Get HOG channels: 5e-06 seconds
Process-2093, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.161706 seconds
Process-2093, Step 6: Misc initializations: 0.0 seconds
Process-2093, Step 7: for loop: 0.009792 seconds

Process-2093All steps time: 0.18245399999999998 seconds

Process-2093, Find cars processing time: 0.182528 seconds

Process-2094, Step 1: Divide with 255, processing time: 0.010178 seconds
Process-2094, Step 2: Resize if scale is not 1: 0.000706 seconds
Process-2094, Step 3: Get HOG channels: 6e-06 seconds
Process-2094, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.200092 seconds
Process-2094, Step 6: Misc initializations: 1e-06 seconds
Process-2094, Step 7: for loop: 0.010162 seconds

Process-2094All steps time: 0.221151 seconds

Process-2094, Find cars processing time: 0.221215 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.014522 seconds
Process-2093, Step 2: Resize if scale is not 1: 0.001023 seconds
Process-2093, Step 3: Get HOG channels: 9e-06 seconds
Process-2093, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.155803 seconds
Process-2093, Step 6: Misc initializations: 0.0 seconds
Process-2093, Step 7: for loop: 0.007922 seconds

Process-2093All steps time: 0.179288 seconds

Process-2093, Find cars processing time: 0.179371 seconds

Process-2094, Step 1: Divide with 255, processing time: 0.011055 seconds
Process-2094, Step 2: Resize if scale is not 1: 0.00091 seconds
Process-2094, Step 3: Get HOG channels: 8e-06 seconds
Process-2094, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.185859 seconds
Process-2094, Step 6: Misc initializations: 0.0 seconds
Process-2094, Step 7: for loop: 0.00804 seconds

Process-2094All steps time: 0.20587999999999998 seconds

Process-2094, Find cars processing time: 0.205946 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.009439 seconds

Process-2093, Step 2: Resize if scale is not 1: 0.000482 seconds
Process-2093, Step 3: Get HOG channels: 5e-06 seconds
Process-2093, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.126409 seconds
Process-2093, Step 6: Misc initializations: 0.0 seconds
Process-2093, Step 7: for loop: 0.007453 seconds

Process-2093All steps time: 0.1437939999999998 seconds

Process-2093, Find cars processing time: 0.143846 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.0105 seconds
Process-2093, Step 2: Resize if scale is not 1: 0.000624 seconds
Process-2093, Step 3: Get HOG channels: 6e-06 seconds
Process-2093, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.115598 seconds
Process-2093, Step 6: Misc initializations: 0.0 seconds
Process-2093, Step 7: for loop: 0.006066 seconds

Process-2093All steps time: 0.1328 seconds

Process-2093, Find cars processing time: 0.13286 seconds

Process-2094, Step 1: Divide with 255, processing time: 0.010875 seconds
Process-2094, Step 2: Resize if scale is not 1: 0.000829 seconds
Process-2094, Step 3: Get HOG channels: 8e-06 seconds
Process-2094, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.13582 seconds
Process-2094, Step 6: Misc initializations: 1e-06 seconds
Process-2094, Step 7: for loop: 0.008089 seconds

Process-2094All steps time: 0.1556300000000002 seconds

Process-2094, Find cars processing time: 0.155711 seconds

Process-2094, Step 1: Divide with 255, processing time: 0.009547 seconds
Process-2093, Step 1: Divide with 255, processing time: 0.007515 seconds
Process-2094, Step 2: Resize if scale is not 1: 0.000793 seconds
Process-2093, Step 2: Resize if scale is not 1: 0.000546 seconds
Process-2094, Step 3: Get HOG channels: 8e-06 seconds
Process-2093, Step 3: Get HOG channels: 6e-06 seconds
Process-2094, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2093, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.118149 seconds
Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.126999 seconds
Process-2094, Step 6: Misc initializations: 1e-06 seconds
Process-2093, Step 6: Misc initializations: 1e-06 seconds
Process-2094, Step 7: for loop: 0.006093 seconds

Process-2093, Step 7: for loop: 0.006016 seconds

Process-2094All steps time: 0.134598 seconds

Process-2093All steps time: 0.141089 seconds

Process-2094, Find cars processing time: 0.134664 seconds

Process-2093, Find cars processing time: 0.141116 seconds

Process-2094, Step 1: Divide with 255, processing time: 0.010085 seconds

Process-2094, Step 2: Resize if scale is not 1: 0.000543 seconds

Process-2094, Step 3: Get HOG channels: 5e-06 seconds

Process-2094, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.124465 seconds

Process-2094, Step 6: Misc initializations: 1e-06 seconds

Process-2094, Step 7: for loop: 0.006007 seconds

Process-2094All steps time: 0.14111300000000002 seconds

Process-2094, Find cars processing time: 0.141183 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.009628 seconds

Process-2093, Step 2: Resize if scale is not 1: 0.000438 seconds

Process-2093, Step 3: Get HOG channels: 5e-06 seconds

Process-2093, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.093524 seconds

Process-2093, Step 6: Misc initializations: 1e-06 seconds

Process-2093, Step 7: for loop: 0.004247 seconds

Process-2093All steps time: 0.107848 seconds

Process-2093, Find cars processing time: 0.107901 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.010705 seconds

Process-2093, Step 2: Resize if scale is not 1: 0.001104 seconds

Process-2093, Step 3: Get HOG channels: 5e-06 seconds

Process-2093, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.095533 seconds

Process-2093, Step 6: Misc initializations: 0.0 seconds

Process-2093, Step 7: for loop: 0.005666 seconds

Process-2093All steps time: 0.11301900000000001 seconds

Process-2093, Find cars processing time: 0.113081 seconds

Process-2094, Step 1: Divide with 255, processing time: 0.010779 seconds

Process-2094, Step 2: Resize if scale is not 1: 0.001381 seconds
Process-2094, Step 3: Get HOG channels: 9e-06 seconds
Process-2094, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.107497 seconds
Process-2094, Step 6: Misc initializations: 0.0 seconds
Process-2094, Step 7: for loop: 0.005773 seconds

Process-2094All steps time: 0.125447 seconds

Process-2094, Find cars processing time: 0.125519 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.011448 seconds
Process-2093, Step 2: Resize if scale is not 1: 0.000488 seconds
Process-2093, Step 3: Get HOG channels: 5e-06 seconds
Process-2093, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.102481 seconds
Process-2093, Step 6: Misc initializations: 0.0 seconds
Process-2093, Step 7: for loop: 0.005619 seconds

Process-2093All steps time: 0.120047 seconds

Process-2093, Find cars processing time: 0.120105 seconds

Process-2094, Step 1: Divide with 255, processing time: 0.010273 seconds
Process-2094, Step 2: Resize if scale is not 1: 0.000743 seconds
Process-2094, Step 3: Get HOG channels: 7e-06 seconds
Process-2094, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2094, Step 5: Compute individual channel HOG features for the entire image: 0.098036 seconds
Process-2094, Step 6: Misc initializations: 0.0 seconds
Process-2094, Step 7: for loop: 0.005893 seconds

Process-2094All steps time: 0.1149599999999999 seconds

Process-2094, Find cars processing time: 0.115019 seconds

Process-2093, Step 1: Divide with 255, processing time: 0.008503 seconds
Process-2093, Step 2: Resize if scale is not 1: 0.001571 seconds
Process-2093, Step 3: Get HOG channels: 6e-06 seconds
Process-2093, Step 4: Define blocks and steps as above: 1.8e-05 seconds
Process-2093, Step 5: Compute individual channel HOG features for the entire image: 0.086555 seconds
Process-2093, Step 6: Misc initializations: 0.0 seconds
Process-2093, Step 7: for loop: 0.004529 seconds

Process-2093All steps time: 0.1011820000000001 seconds

Process-2093, Find cars processing time: 0.101233 seconds

The times for each task are: [0.23227, 0.278903, 0.206611, 0.252574, 0.177851, 0.218602, 0.18252]

Minimum: 0.101233 Maximum: 0.278903 Average: 0.1676 seconds

Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]

2 processes used for testing 5 window sizes

Processing times for each image [5.7975, 5.9579, 5.1522, 6.0296, 5.6412, 5.4682] with an average

Time elapsed so far... 376.7646000000001

Number of processes used: 3 window size 260

Length of task list: 22

Number of processes used: 3

Process-2095, Step 1: Divide with 255, processing time: 0.032963 seconds

Process-2095, Step 2: Resize if scale is not 1: 0.002551 seconds

Process-2095, Step 3: Get HOG channels: 2.5e-05 seconds

Process-2095, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2095, Step 5: Compute individual channel HOG features for the entire image: 0.252555 seconds

Process-2095, Step 6: Misc initializations: 1e-06 seconds

Process-2095, Step 7: for loop: 0.018281 seconds

Process-2095All steps time: 0.3063819999999993 seconds

Process-2095, Find cars processing time: 0.306559 seconds

Process-2096, Step 1: Divide with 255, processing time: 0.056618 seconds

Process-2096, Step 2: Resize if scale is not 1: 0.004127 seconds

Process-2096, Step 3: Get HOG channels: 5.4e-05 seconds

Process-2096, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2096, Step 5: Compute individual channel HOG features for the entire image: 0.265142 seconds

Process-2096, Step 6: Misc initializations: 1e-06 seconds

Process-2096, Step 7: for loop: 0.018972 seconds

Process-2096All steps time: 0.3449189999999999 seconds

Process-2096, Find cars processing time: 0.345108 seconds

Process-2097, Step 1: Divide with 255, processing time: 0.028366 seconds

Process-2097, Step 2: Resize if scale is not 1: 0.00198 seconds

Process-2097, Step 3: Get HOG channels: 2.5e-05 seconds

Process-2097, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2097, Step 5: Compute individual channel HOG features for the entire image: 0.249037 seconds

Process-2097, Step 6: Misc initializations: 1e-06 seconds

Process-2097, Step 7: for loop: 0.016963 seconds

Process-2097All steps time: 0.296378 seconds

Process-2097, Find cars processing time: 0.296546 seconds

Process-2095, Step 1: Divide with 255, processing time: 0.007873 seconds

Process-2095, Step 2: Resize if scale is not 1: 0.000762 seconds

Process-2095, Step 3: Get HOG channels: 8e-06 seconds

Process-2095, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2095, Step 5: Compute individual channel HOG features for the entire image: 0.187631 seconds

Process-2095, Step 6: Misc initializations: 1e-06 seconds

Process-2095, Step 7: for loop: 0.012799 seconds

Process-2095All steps time: 0.209081 seconds

Process-2095, Find cars processing time: 0.209136 seconds

Process-2095, Step 1: Divide with 255, processing time: 0.007548 seconds

Process-2095, Step 2: Resize if scale is not 1: 0.000731 seconds

Process-2095, Step 3: Get HOG channels: 6e-06 seconds

Process-2095, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2095, Step 5: Compute individual channel HOG features for the entire image: 0.146771 seconds

Process-2095, Step 6: Misc initializations: 0.0 seconds

Process-2095, Step 7: for loop: 0.0094 seconds

Process-2095All steps time: 0.164463 seconds

Process-2095, Find cars processing time: 0.16451 seconds

Process-2096, Step 1: Divide with 255, processing time: 0.010385 seconds

Process-2096, Step 2: Resize if scale is not 1: 0.000935 seconds

Process-2096, Step 3: Get HOG channels: 6e-06 seconds

Process-2096, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2096, Step 5: Compute individual channel HOG features for the entire image: 0.183154 seconds

Process-2096, Step 6: Misc initializations: 1e-06 seconds

Process-2096, Step 7: for loop: 0.010875 seconds

Process-2096All steps time: 0.2053620000000002 seconds

Process-2096, Find cars processing time: 0.205426 seconds

Process-2097, Step 1: Divide with 255, processing time: 0.010551 seconds

Process-2097, Step 2: Resize if scale is not 1: 0.001354 seconds

Process-2097, Step 3: Get HOG channels: 1.4e-05 seconds

Process-2097, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-2097, Step 5: Compute individual channel HOG features for the entire image: 0.211177 seconds

Process-2097, Step 6: Misc initializations: 1e-06 seconds

Process-2097, Step 7: for loop: 0.013484 seconds

Process-2097All steps time: 0.236592 seconds

Process-2097, Find cars processing time: 0.236701 seconds

Process-2096, Step 1: Divide with 255, processing time: 0.010696 seconds

Process-2096, Step 2: Resize if scale is not 1: 0.000678 seconds

Process-2096, Step 3: Get HOG channels: 6e-06 seconds

Process-2096, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2096, Step 5: Compute individual channel HOG features for the entire image: 0.170203 seconds

Process-2096, Step 6: Misc initializations: 1e-06 seconds

Process-2096, Step 7: for loop: 0.009719 seconds

Process-2096All steps time: 0.191309 seconds

Process-2096, Find cars processing time: 0.191375 seconds

Process-2095, Step 1: Divide with 255, processing time: 0.009864 seconds

Process-2095, Step 2: Resize if scale is not 1: 0.000536 seconds

Process-2095, Step 3: Get HOG channels: 6e-06 seconds

Process-2095, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2095, Step 5: Compute individual channel HOG features for the entire image: 0.194978 seconds

Process-2095, Step 6: Misc initializations: 1e-06 seconds

Process-2095, Step 7: for loop: 0.009982 seconds

Process-2095All steps time: 0.215372 seconds

Process-2095, Find cars processing time: 0.215429 seconds

Process-2097, Step 1: Divide with 255, processing time: 0.010519 seconds

Process-2097, Step 2: Resize if scale is not 1: 0.001044 seconds

Process-2097, Step 3: Get HOG channels: 1.3e-05 seconds

Process-2097, Step 4: Define blocks and steps as above: 1.3e-05 seconds

Process-2097, Step 5: Compute individual channel HOG features for the entire image: 0.176082 seconds

Process-2097, Step 6: Misc initializations: 1e-06 seconds

Process-2097, Step 7: for loop: 0.009754 seconds

Process-2097All steps time: 0.197426 seconds

Process-2097, Find cars processing time: 0.197523 seconds

Process-2096, Step 1: Divide with 255, processing time: 0.010293 seconds

Process-2096, Step 2: Resize if scale is not 1: 0.000492 seconds

Process-2096, Step 3: Get HOG channels: 5e-06 seconds

Process-2096, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2096, Step 5: Compute individual channel HOG features for the entire image: 0.178921 seconds

Process-2096, Step 6: Misc initializations: 1e-06 seconds

Process-2096, Step 7: for loop: 0.009817 seconds

Process-2096All steps time: 0.199535 seconds

Process-2096, Find cars processing time: 0.199625 seconds

Process-2097, Step 1: Divide with 255, processing time: 0.008665 seconds

Process-2097, Step 2: Resize if scale is not 1: 0.000695 seconds

Process-2097, Step 3: Get HOG channels: 7e-06 seconds

Process-2097, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2097, Step 5: Compute individual channel HOG features for the entire image: 0.123537 seconds

Process-2097, Step 6: Misc initializations: 0.0 seconds

Process-2097, Step 7: for loop: 0.006142 seconds

Process-2097All steps time: 0.139053 seconds

Process-2097, Find cars processing time: 0.139136 seconds

Process-2095, Step 1: Divide with 255, processing time: 0.010682 seconds

Process-2095, Step 2: Resize if scale is not 1: 0.000877 seconds

Process-2095, Step 3: Get HOG channels: 8e-06 seconds

Process-2095, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2095, Step 5: Compute individual channel HOG features for the entire image: 0.136522 seconds

Process-2095, Step 6: Misc initializations: 0.0 seconds

Process-2095, Step 7: for loop: 0.007616 seconds

Process-2097, Step 1: Divide with 255, processing time: 0.007432 seconds

Process-2097, Step 2: Resize if scale is not 1: 0.000615 seconds

Process-2095All steps time: 0.15571300000000002 seconds

Process-2097, Step 3: Get HOG channels: 6e-06 seconds

Process-2097, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2095, Find cars processing time: 0.155783 seconds

Process-2097, Step 5: Compute individual channel HOG features for the entire image: 0.143249 seconds

Process-2097, Step 6: Misc initializations: 1e-06 seconds

Process-2097, Step 7: for loop: 0.007261 seconds

Process-2097All steps time: 0.15857 seconds

Process-2097, Find cars processing time: 0.158627 seconds

Process-2096, Step 1: Divide with 255, processing time: 0.006382 seconds

Process-2096, Step 2: Resize if scale is not 1: 0.000568 seconds

Process-2096, Step 3: Get HOG channels: 6e-06 seconds

Process-2096, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2096, Step 5: Compute individual channel HOG features for the entire image: 0.158402 seconds

Process-2096, Step 6: Misc initializations: 1e-06 seconds

Process-2096, Step 7: for loop: 0.007349 seconds

Process-2096All steps time: 0.1727129999999998 seconds

Process-2096, Find cars processing time: 0.172778 seconds

Process-2095, Step 1: Divide with 255, processing time: 0.011337 seconds

Process-2095, Step 2: Resize if scale is not 1: 0.000486 seconds

Process-2095, Step 3: Get HOG channels: 5e-06 seconds

Process-2095, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2095, Step 5: Compute individual channel HOG features for the entire image: 0.116721 seconds

Process-2095, Step 6: Misc initializations: 0.0 seconds

Process-2095, Step 7: for loop: 0.005491 seconds

Process-2095All steps time: 0.134045 seconds

Process-2095, Find cars processing time: 0.134115 seconds

Process-2096, Step 1: Divide with 255, processing time: 0.008317 seconds

Process-2096, Step 2: Resize if scale is not 1: 0.001077 seconds

Process-2096, Step 3: Get HOG channels: 1e-05 seconds

Process-2096, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2096, Step 5: Compute individual channel HOG features for the entire image: 0.130696 seconds

Process-2096, Step 6: Misc initializations: 0.0 seconds

Process-2096, Step 7: for loop: 0.007303 seconds

Process-2096All steps time: 0.14741100000000001 seconds

Process-2096, Find cars processing time: 0.147487 seconds

Process-2097, Step 1: Divide with 255, processing time: 0.010773 seconds

Process-2097, Step 2: Resize if scale is not 1: 0.000912 seconds

Process-2097, Step 3: Get HOG channels: 1.1e-05 seconds

Process-2097, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-2097, Step 5: Compute individual channel HOG features for the entire image: 0.130502 seconds

Process-2097, Step 6: Misc initializations: 1e-06 seconds

Process-2097, Step 7: for loop: 0.005002 seconds

Process-2097All steps time: 0.147212 seconds

Process-2097, Find cars processing time: 0.147292 seconds

Process-2095, Step 1: Divide with 255, processing time: 0.011374 seconds

Process-2095, Step 2: Resize if scale is not 1: 0.001377 seconds

Process-2095, Step 3: Get HOG channels: 9e-06 seconds

Process-2095, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2095, Step 5: Compute individual channel HOG features for the entire image: 0.157918 seconds

Process-2095, Step 6: Misc initializations: 1e-06 seconds

Process-2095, Step 7: for loop: 0.007312 seconds

Process-2095All steps time: 0.17800000000000002 seconds

Process-2095, Find cars processing time: 0.178076 seconds

Process-2096, Step 1: Divide with 255, processing time: 0.026274 seconds

Process-2096, Step 2: Resize if scale is not 1: 0.000502 seconds

Process-2096, Step 3: Get HOG channels: 6e-06 seconds

Process-2096, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2096, Step 5: Compute individual channel HOG features for the entire image: 0.126651 seconds

Process-2096, Step 6: Misc initializations: 0.0 seconds

Process-2096, Step 7: for loop: 0.007179 seconds

Process-2097, Step 1: Divide with 255, processing time: 0.010454 seconds

Process-2096All steps time: 0.160618 seconds

Process-2097, Step 2: Resize if scale is not 1: 0.00133 seconds

Process-2097, Step 3: Get HOG channels: 9e-06 seconds

Process-2096, Find cars processing time: 0.160679 seconds

Process-2097, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2097, Step 5: Compute individual channel HOG features for the entire image: 0.127787 seconds

Process-2097, Step 6: Misc initializations: 1e-06 seconds

Process-2097, Step 7: for loop: 0.007288 seconds

Process-2097All steps time: 0.146876 seconds

Process-2097, Find cars processing time: 0.146945 seconds

Process-2095, Step 1: Divide with 255, processing time: 0.009588 seconds

Process-2095, Step 2: Resize if scale is not 1: 0.001492 seconds

Process-2095, Step 3: Get HOG channels: 1.9e-05 seconds

Process-2095, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2095, Step 5: Compute individual channel HOG features for the entire image: 0.116746 seconds

Process-2095, Step 6: Misc initializations: 1e-06 seconds

Process-2095, Step 7: for loop: 0.006732 seconds

Process-2095All steps time: 0.1345849999999998 seconds

Process-2095, Find cars processing time: 0.134661 seconds

The times for each task are: [0.306559, 0.345108, 0.296546, 0.209136, 0.16451, 0.205426, 0.23670

Minimum: 0.134115 Maximum: 0.345108 Average: 0.1929 seconds

Number of processes used: 3 window size 260

Length of task list: 22

Number of processes used: 3

Process-2098, Step 1: Divide with 255, processing time: 0.02851 seconds
Process-2098, Step 2: Resize if scale is not 1: 0.002055 seconds
Process-2098, Step 3: Get HOG channels: 2e-05 seconds
Process-2098, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2098, Step 5: Compute individual channel HOG features for the entire image: 0.25755 seconds
Process-2098, Step 6: Misc initializations: 1e-06 seconds
Process-2098, Step 7: for loop: 0.014544 seconds

Process-2098All steps time: 0.30268599999999996 seconds

Process-2098, Find cars processing time: 0.302877 seconds

Process-2099, Step 1: Divide with 255, processing time: 0.024838 seconds
Process-2099, Step 2: Resize if scale is not 1: 0.002008 seconds
Process-2099, Step 3: Get HOG channels: 2.1e-05 seconds
Process-2099, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2099, Step 5: Compute individual channel HOG features for the entire image: 0.241033 seconds
Process-2099, Step 6: Misc initializations: 1e-06 seconds
Process-2099, Step 7: for loop: 0.016417 seconds

Process-2099All steps time: 0.2843239999999997 seconds

Process-2099, Find cars processing time: 0.284474 seconds

Process-2098, Step 1: Divide with 255, processing time: 0.008799 seconds
Process-2098, Step 2: Resize if scale is not 1: 0.000558 seconds
Process-2098, Step 3: Get HOG channels: 8e-06 seconds
Process-2098, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2098, Step 5: Compute individual channel HOG features for the entire image: 0.221866 seconds
Process-2098, Step 6: Misc initializations: 0.0 seconds
Process-2098, Step 7: for loop: 0.0126 seconds

Process-2098All steps time: 0.243838 seconds

Process-2098, Find cars processing time: 0.243893 seconds

Process-2100, Step 1: Divide with 255, processing time: 0.029789 seconds
Process-2100, Step 2: Resize if scale is not 1: 0.002058 seconds
Process-2100, Step 3: Get HOG channels: 1.8e-05 seconds
Process-2100, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2100, Step 5: Compute individual channel HOG features for the entire image: 0.229927 seconds
Process-2100, Step 6: Misc initializations: 1e-06 seconds
Process-2100, Step 7: for loop: 0.01339 seconds

Process-2100All steps time: 0.275188 seconds

Process-2100, Find cars processing time: 0.275344 seconds

Process-2099, Step 1: Divide with 255, processing time: 0.009852 seconds
Process-2099, Step 2: Resize if scale is not 1: 0.000861 seconds
Process-2099, Step 3: Get HOG channels: 1e-05 seconds
Process-2099, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2099, Step 5: Compute individual channel HOG features for the entire image: 0.171953 seconds
Process-2099, Step 6: Misc initializations: 1e-06 seconds
Process-2099, Step 7: for loop: 0.009993 seconds

Process-2099All steps time: 0.192679 seconds

Process-2099, Find cars processing time: 0.192758 seconds

Process-2098, Step 1: Divide with 255, processing time: 0.010601 seconds
Process-2098, Step 2: Resize if scale is not 1: 0.000862 seconds
Process-2098, Step 3: Get HOG channels: 8e-06 seconds
Process-2098, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2098, Step 5: Compute individual channel HOG features for the entire image: 0.162101 seconds
Process-2098, Step 6: Misc initializations: 0.0 seconds
Process-2098, Step 7: for loop: 0.010293 seconds

Process-2098All steps time: 0.1838719999999998 seconds

Process-2098, Find cars processing time: 0.183936 seconds

Process-2100, Step 1: Divide with 255, processing time: 0.010389 seconds
Process-2100, Step 2: Resize if scale is not 1: 0.000656 seconds
Process-2100, Step 3: Get HOG channels: 5e-06 seconds
Process-2100, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2100, Step 5: Compute individual channel HOG features for the entire image: 0.161704 seconds
Process-2100, Step 6: Misc initializations: 0.0 seconds
Process-2100, Step 7: for loop: 0.010382 seconds

Process-2100All steps time: 0.18314 seconds

Process-2100, Find cars processing time: 0.183199 seconds

Process-2098, Step 1: Divide with 255, processing time: 0.010199 seconds
Process-2098, Step 2: Resize if scale is not 1: 0.000838 seconds
Process-2098, Step 3: Get HOG channels: 1.8e-05 seconds
Process-2098, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2098, Step 5: Compute individual channel HOG features for the entire image: 0.153388 seconds
Process-2098, Step 6: Misc initializations: 1e-06 seconds
Process-2098, Step 7: for loop: 0.008184 seconds

Process-2098All steps time: 0.1726359999999998 seconds

Process-2098, Find cars processing time: 0.172698 seconds

Process-2099, Step 1: Divide with 255, processing time: 0.005208 seconds
Process-2099, Step 2: Resize if scale is not 1: 0.000437 seconds
Process-2099, Step 3: Get HOG channels: 5e-06 seconds
Process-2099, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2099, Step 5: Compute individual channel HOG features for the entire image: 0.152182 seconds
Process-2099, Step 6: Misc initializations: 0.0 seconds
Process-2099, Step 7: for loop: 0.009708 seconds

Process-2099All steps time: 0.167545 seconds

Process-2099, Find cars processing time: 0.16759 seconds

Process-2100, Step 1: Divide with 255, processing time: 0.009651 seconds
Process-2100, Step 2: Resize if scale is not 1: 0.000807 seconds
Process-2100, Step 3: Get HOG channels: 7e-06 seconds
Process-2100, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2100, Step 5: Compute individual channel HOG features for the entire image: 0.133308 seconds
Process-2100, Step 6: Misc initializations: 1e-06 seconds
Process-2100, Step 7: for loop: 0.007849 seconds

Process-2100All steps time: 0.151629 seconds

Process-2100, Find cars processing time: 0.1517 seconds

Process-2098, Step 1: Divide with 255, processing time: 0.010769 seconds
Process-2098, Step 2: Resize if scale is not 1: 0.000863 seconds
Process-2098, Step 3: Get HOG channels: 9e-06 seconds
Process-2098, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2098, Step 5: Compute individual channel HOG features for the entire image: 0.130175 seconds
Process-2098, Step 6: Misc initializations: 1e-06 seconds
Process-2098, Step 7: for loop: 0.007849 seconds

Process-2098All steps time: 0.149676 seconds

Process-2098, Find cars processing time: 0.149768 seconds

Process-2099, Step 1: Divide with 255, processing time: 0.026142 seconds
Process-2099, Step 2: Resize if scale is not 1: 0.000524 seconds
Process-2099, Step 3: Get HOG channels: 5e-06 seconds
Process-2099, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2099, Step 5: Compute individual channel HOG features for the entire image: 0.16688 seconds
Process-2099, Step 6: Misc initializations: 1e-06 seconds
Process-2099, Step 7: for loop: 0.009688 seconds

Process-2099All steps time: 0.203245 seconds

Process-2099, Find cars processing time: 0.203301 seconds

Process-2100, Step 1: Divide with 255, processing time: 0.010346 seconds
Process-2100, Step 2: Resize if scale is not 1: 0.000839 seconds
Process-2100, Step 3: Get HOG channels: 8e-06 seconds
Process-2100, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2100, Step 5: Compute individual channel HOG features for the entire image: 0.130272 seconds
Process-2100, Step 6: Misc initializations: 0.0 seconds
Process-2100, Step 7: for loop: 0.00611 seconds

Process-2100All steps time: 0.147583 seconds

Process-2100, Find cars processing time: 0.147645 seconds

Process-2098, Step 1: Divide with 255, processing time: 0.010707 seconds
Process-2099, Step 1: Divide with 255, processing time: 0.007763 seconds
Process-2099, Step 2: Resize if scale is not 1: 0.000601 seconds
Process-2099, Step 3: Get HOG channels: 5e-06 seconds
Process-2099, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2099, Step 5: Compute individual channel HOG features for the entire image: 0.141101 seconds
Process-2099, Step 6: Misc initializations: 0.0 seconds
Process-2099, Step 7: for loop: 0.006199 seconds

Process-2099All steps time: 0.155674 seconds

Process-2099, Find cars processing time: 0.155721 seconds

Process-2098, Step 2: Resize if scale is not 1: 0.0007 seconds
Process-2098, Step 3: Get HOG channels: 5e-06 seconds
Process-2098, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2098, Step 5: Compute individual channel HOG features for the entire image: 0.142288 seconds
Process-2098, Step 6: Misc initializations: 0.0 seconds
Process-2098, Step 7: for loop: 0.007189 seconds

Process-2098All steps time: 0.160895 seconds

Process-2098, Find cars processing time: 0.160965 seconds

Process-2099, Step 1: Divide with 255, processing time: 0.007953 seconds
Process-2099, Step 2: Resize if scale is not 1: 0.000558 seconds
Process-2099, Step 3: Get HOG channels: 6e-06 seconds
Process-2099, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2099, Step 5: Compute individual channel HOG features for the entire image: 0.08928 seconds
Process-2099, Step 6: Misc initializations: 1e-06 seconds
Process-2099, Step 7: for loop: 0.004029 seconds

Process-2099All steps time: 0.10183400000000001 seconds

Process-2099, Find cars processing time: 0.101902 seconds

Process-2100, Step 1: Divide with 255, processing time: 0.009621 seconds
Process-2100, Step 2: Resize if scale is not 1: 0.000471 seconds
Process-2100, Step 3: Get HOG channels: 6e-06 seconds
Process-2100, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2100, Step 5: Compute individual channel HOG features for the entire image: 0.109304 seconds
Process-2100, Step 6: Misc initializations: 0.0 seconds
Process-2100, Step 7: for loop: 0.005398 seconds

Process-2100All steps time: 0.124806 seconds

Process-2100, Find cars processing time: 0.124868 seconds

Process-2098, Step 1: Divide with 255, processing time: 0.006556 seconds
Process-2098, Step 2: Resize if scale is not 1: 0.000878 seconds
Process-2098, Step 3: Get HOG channels: 6e-06 seconds
Process-2098, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2098, Step 5: Compute individual channel HOG features for the entire image: 0.127116 seconds
Process-2098, Step 6: Misc initializations: 1e-06 seconds
Process-2098, Step 7: for loop: 0.007078 seconds

Process-2098All steps time: 0.1416420000000002 seconds

Process-2098, Find cars processing time: 0.141705 seconds

Process-2099, Step 1: Divide with 255, processing time: 0.010334 seconds
Process-2099, Step 2: Resize if scale is not 1: 0.001129 seconds
Process-2099, Step 3: Get HOG channels: 5e-06 seconds
Process-2099, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2099, Step 5: Compute individual channel HOG features for the entire image: 0.109275 seconds
Process-2099, Step 6: Misc initializations: 1e-06 seconds
Process-2099, Step 7: for loop: 0.00612 seconds

Process-2099All steps time: 0.1268689999999998 seconds

Process-2099, Find cars processing time: 0.126923 seconds

Process-2100, Step 1: Divide with 255, processing time: 0.008839 seconds
Process-2100, Step 2: Resize if scale is not 1: 0.001197 seconds
Process-2100, Step 3: Get HOG channels: 7e-06 seconds
Process-2100, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2100, Step 5: Compute individual channel HOG features for the entire image: 0.131433 seconds
Process-2100, Step 6: Misc initializations: 1e-06 seconds
Process-2100, Step 7: for loop: 0.008028 seconds

Process-2100All steps time: 0.149513 seconds

Process-2100, Find cars processing time: 0.149591 seconds

Process-2098, Step 1: Divide with 255, processing time: 0.007511 seconds
Process-2098, Step 2: Resize if scale is not 1: 0.000595 seconds
Process-2098, Step 3: Get HOG channels: 6e-06 seconds
Process-2098, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2098, Step 5: Compute individual channel HOG features for the entire image: 0.127341 seconds
Process-2098, Step 6: Misc initializations: 1e-06 seconds
Process-2098, Step 7: for loop: 0.007104 seconds

Process-2098All steps time: 0.14256400000000002 seconds

Process-2098, Find cars processing time: 0.142622 seconds

Process-2099, Step 1: Divide with 255, processing time: 0.010835 seconds
Process-2099, Step 2: Resize if scale is not 1: 0.001163 seconds
Process-2099, Step 3: Get HOG channels: 5e-06 seconds
Process-2099, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2099, Step 5: Compute individual channel HOG features for the entire image: 0.085449 seconds
Process-2099, Step 6: Misc initializations: 0.0 seconds
Process-2099, Step 7: for loop: 0.004932 seconds

Process-2099All steps time: 0.10239 seconds

Process-2099, Find cars processing time: 0.102451 seconds

The times for each task are: [0.302877, 0.284474, 0.243893, 0.275344, 0.192758, 0.183936, 0.183144]

Minimum: 0.101902 Maximum: 0.302877 Average: 0.1757 seconds

Number of processes used: 3 window size 260
Length of task list: 22
Number of processes used: 3

Process-2101, Step 1: Divide with 255, processing time: 0.02941 seconds
Process-2101, Step 2: Resize if scale is not 1: 0.002221 seconds
Process-2101, Step 3: Get HOG channels: 2e-05 seconds
Process-2101, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2101, Step 5: Compute individual channel HOG features for the entire image: 0.273267 seconds
Process-2101, Step 6: Misc initializations: 2e-06 seconds
Process-2101, Step 7: for loop: 0.018475 seconds

Process-2101All steps time: 0.323401 seconds

Process-2101, Find cars processing time: 0.323607 seconds

Process-2102, Step 1: Divide with 255, processing time: 0.027582 seconds
Process-2102, Step 2: Resize if scale is not 1: 0.002024 seconds
Process-2102, Step 3: Get HOG channels: 1.9e-05 seconds

Process-2102, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2102, Step 5: Compute individual channel HOG features for the entire image: 0.250197 seconds
Process-2102, Step 6: Misc initializations: 1e-06 seconds
Process-2102, Step 7: for loop: 0.018391 seconds

Process-2102All steps time: 0.29822 seconds

Process-2102, Find cars processing time: 0.298424 seconds

Process-2101, Step 1: Divide with 255, processing time: 0.010935 seconds
Process-2101, Step 2: Resize if scale is not 1: 0.000943 seconds
Process-2101, Step 3: Get HOG channels: 9e-06 seconds
Process-2101, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2101, Step 5: Compute individual channel HOG features for the entire image: 0.202925 seconds
Process-2101, Step 6: Misc initializations: 1e-06 seconds
Process-2101, Step 7: for loop: 0.013767 seconds

Process-2101All steps time: 0.228589 seconds

Process-2101, Find cars processing time: 0.22868 seconds

Process-2102, Step 1: Divide with 255, processing time: 0.009778 seconds
Process-2102, Step 2: Resize if scale is not 1: 0.000827 seconds
Process-2102, Step 3: Get HOG channels: 7e-06 seconds
Process-2102, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2102, Step 5: Compute individual channel HOG features for the entire image: 0.168449 seconds
Process-2102, Step 6: Misc initializations: 1e-06 seconds
Process-2102, Step 7: for loop: 0.01105 seconds

Process-2102All steps time: 0.1901189999999998 seconds

Process-2102, Find cars processing time: 0.190186 seconds

Process-2103, Step 1: Divide with 255, processing time: 0.028864 seconds
Process-2103, Step 2: Resize if scale is not 1: 0.002197 seconds
Process-2103, Step 3: Get HOG channels: 2.8e-05 seconds
Process-2103, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2103, Step 5: Compute individual channel HOG features for the entire image: 0.262155 seconds
Process-2103, Step 6: Misc initializations: 1e-06 seconds
Process-2103, Step 7: for loop: 0.017296 seconds

Process-2103All steps time: 0.310546 seconds

Process-2103, Find cars processing time: 0.310725 seconds

Process-2102, Step 1: Divide with 255, processing time: 0.010772 seconds
Process-2102, Step 2: Resize if scale is not 1: 0.000988 seconds
Process-2102, Step 3: Get HOG channels: 1e-05 seconds

Process-2102, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2102, Step 5: Compute individual channel HOG features for the entire image: 0.203637 seconds
Process-2102, Step 6: Misc initializations: 1e-06 seconds
Process-2102, Step 7: for loop: 0.012868 seconds

Process-2102All steps time: 0.22828500000000002 seconds

Process-2102, Find cars processing time: 0.228371 seconds

Process-2103, Step 1: Divide with 255, processing time: 0.010238 seconds
Process-2103, Step 2: Resize if scale is not 1: 0.000887 seconds
Process-2103, Step 3: Get HOG channels: 8e-06 seconds
Process-2103, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2103, Step 5: Compute individual channel HOG features for the entire image: 0.221357 seconds
Process-2103, Step 6: Misc initializations: 1e-06 seconds
Process-2103, Step 7: for loop: 0.013168 seconds

Process-2103All steps time: 0.245666 seconds

Process-2103, Find cars processing time: 0.245745 seconds

Process-2101, Step 1: Divide with 255, processing time: 0.010542 seconds
Process-2101, Step 2: Resize if scale is not 1: 0.000871 seconds
Process-2101, Step 3: Get HOG channels: 8e-06 seconds
Process-2101, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2101, Step 5: Compute individual channel HOG features for the entire image: 0.216301 seconds
Process-2101, Step 6: Misc initializations: 1e-06 seconds
Process-2101, Step 7: for loop: 0.012851 seconds

Process-2101All steps time: 0.240581 seconds

Process-2101, Find cars processing time: 0.240648 seconds

Process-2102, Step 1: Divide with 255, processing time: 0.010463 seconds
Process-2102, Step 2: Resize if scale is not 1: 0.000925 seconds
Process-2102, Step 3: Get HOG channels: 8e-06 seconds
Process-2102, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2102, Step 5: Compute individual channel HOG features for the entire image: 0.149093 seconds
Process-2102, Step 6: Misc initializations: 1e-06 seconds
Process-2102, Step 7: for loop: 0.008392 seconds

Process-2102All steps time: 0.16889 seconds

Process-2102, Find cars processing time: 0.168959 seconds

Process-2103, Step 1: Divide with 255, processing time: 0.007002 seconds
Process-2103, Step 2: Resize if scale is not 1: 0.000725 seconds
Process-2103, Step 3: Get HOG channels: 8e-06 seconds

Process-2103, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2103, Step 5: Compute individual channel HOG features for the entire image: 0.175574 seconds

Process-2103, Step 6: Misc initializations: 1e-06 seconds

Process-2103, Step 7: for loop: 0.009709 seconds

Process-2103All steps time: 0.193026 seconds

Process-2103, Find cars processing time: 0.193091 seconds

Process-2101, Step 1: Divide with 255, processing time: 0.010386 seconds

Process-2101, Step 2: Resize if scale is not 1: 0.000622 seconds

Process-2101, Step 3: Get HOG channels: 6e-06 seconds

Process-2101, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2101, Step 5: Compute individual channel HOG features for the entire image: 0.172188 seconds

Process-2101, Step 6: Misc initializations: 1e-06 seconds

Process-2101, Step 7: for loop: 0.010222 seconds

Process-2101All steps time: 0.1934300000000002 seconds

Process-2101, Find cars processing time: 0.193501 seconds

Process-2102, Step 1: Divide with 255, processing time: 0.009298 seconds

Process-2102, Step 2: Resize if scale is not 1: 0.000674 seconds

Process-2102, Step 3: Get HOG channels: 7e-06 seconds

Process-2102, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2102, Step 5: Compute individual channel HOG features for the entire image: 0.172232 seconds

Process-2102, Step 6: Misc initializations: 1e-06 seconds

Process-2102, Step 7: for loop: 0.010517 seconds

Process-2102All steps time: 0.192735 seconds

Process-2102, Find cars processing time: 0.19287 seconds

Process-2102, Step 1: Divide with 255, processing time: 0.011141 seconds

Process-2102, Step 2: Resize if scale is not 1: 0.001065 seconds

Process-2102, Step 3: Get HOG channels: 2.4e-05 seconds

Process-2102, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2102, Step 5: Compute individual channel HOG features for the entire image: 0.145033 seconds

Process-2102, Step 6: Misc initializations: 0.0 seconds

Process-2102, Step 7: for loop: 0.007205 seconds

Process-2102All steps time: 0.1644769999999998 seconds

Process-2102, Find cars processing time: 0.164601 seconds

Process-2103, Step 1: Divide with 255, processing time: 0.011021 seconds

Process-2103, Step 2: Resize if scale is not 1: 0.000588 seconds

Process-2103, Step 3: Get HOG channels: 6e-06 seconds

Process-2103, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2103, Step 5: Compute individual channel HOG features for the entire image: 0.172157 seconds

Process-2103, Step 6: Misc initializations: 1e-06 seconds

Process-2103, Step 7: for loop: 0.008569 seconds

Process-2103All steps time: 0.192348 seconds

Process-2103, Find cars processing time: 0.192459 seconds

Process-2101, Step 1: Divide with 255, processing time: 0.009614 seconds

Process-2101, Step 2: Resize if scale is not 1: 0.000547 seconds

Process-2101, Step 3: Get HOG channels: 7e-06 seconds

Process-2101, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2101, Step 5: Compute individual channel HOG features for the entire image: 0.151479 seconds

Process-2101, Step 6: Misc initializations: 6e-06 seconds

Process-2101, Step 7: for loop: 0.007973 seconds

Process-2101All steps time: 0.169632 seconds

Process-2101, Find cars processing time: 0.169757 seconds

Process-2102, Step 1: Divide with 255, processing time: 0.010391 seconds

Process-2102, Step 2: Resize if scale is not 1: 0.000856 seconds

Process-2102, Step 3: Get HOG channels: 8e-06 seconds

Process-2102, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2102, Step 5: Compute individual channel HOG features for the entire image: 0.139927 seconds

Process-2102, Step 6: Misc initializations: 0.0 seconds

Process-2102, Step 7: for loop: 0.006137 seconds

Process-2102All steps time: 0.157326 seconds

Process-2102, Find cars processing time: 0.15739 seconds

Process-2103, Step 1: Divide with 255, processing time: 0.00895 seconds

Process-2103, Step 2: Resize if scale is not 1: 0.000609 seconds

Process-2103, Step 3: Get HOG channels: 5e-06 seconds

Process-2103, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2103, Step 5: Compute individual channel HOG features for the entire image: 0.107761 seconds

Process-2103, Step 6: Misc initializations: 0.0 seconds

Process-2103, Step 7: for loop: 0.0049 seconds

Process-2103All steps time: 0.1222309999999999 seconds

Process-2103, Find cars processing time: 0.122303 seconds

Process-2101, Step 1: Divide with 255, processing time: 0.010438 seconds

Process-2101, Step 2: Resize if scale is not 1: 0.000966 seconds

Process-2101, Step 3: Get HOG channels: 1.6e-05 seconds

Process-2101, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2101, Step 5: Compute individual channel HOG features for the entire image: 0.126761 seconds

Process-2101, Step 6: Misc initializations: 0.0 seconds

Process-2101, Step 7: for loop: 0.007248 seconds

Process-2101All steps time: 0.145434 seconds

Process-2101, Find cars processing time: 0.14553 seconds

Process-2101, Step 1: Divide with 255, processing time: 0.010979 seconds

Process-2101, Step 2: Resize if scale is not 1: 0.000643 seconds

Process-2101, Step 3: Get HOG channels: 7e-06 seconds

Process-2101, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2101, Step 5: Compute individual channel HOG features for the entire image: 0.127321 seconds

Process-2101, Step 6: Misc initializations: 1e-06 seconds

Process-2101, Step 7: for loop: 0.007263 seconds

Process-2101All steps time: 0.146222 seconds

Process-2101, Find cars processing time: 0.146299 seconds

Process-2102, Step 1: Divide with 255, processing time: 0.010713 seconds

Process-2102, Step 2: Resize if scale is not 1: 0.001486 seconds

Process-2102, Step 3: Get HOG channels: 1e-05 seconds

Process-2102, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2102, Step 5: Compute individual channel HOG features for the entire image: 0.131231 seconds

Process-2102, Step 6: Misc initializations: 1e-06 seconds

Process-2102, Step 7: for loop: 0.008585 seconds

Process-2102All steps time: 0.152034 seconds

Process-2102, Find cars processing time: 0.152114 seconds

Process-2103, Step 1: Divide with 255, processing time: 0.009579 seconds

Process-2103, Step 2: Resize if scale is not 1: 0.000869 seconds

Process-2103, Step 3: Get HOG channels: 5e-06 seconds

Process-2103, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2103, Step 5: Compute individual channel HOG features for the entire image: 0.129711 seconds

Process-2103, Step 6: Misc initializations: 0.0 seconds

Process-2103, Step 7: for loop: 0.008246 seconds

Process-2103All steps time: 0.148417 seconds

Process-2103, Find cars processing time: 0.148479 seconds

Process-2101, Step 1: Divide with 255, processing time: 0.010734 seconds

Process-2101, Step 2: Resize if scale is not 1: 0.00186 seconds

Process-2101, Step 3: Get HOG channels: 7e-06 seconds

Process-2101, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2101, Step 5: Compute individual channel HOG features for the entire image: 0.112511 seconds

Process-2101, Step 6: Misc initializations: 1e-06 seconds

Process-2101, Step 7: for loop: 0.006694 seconds

Process-2101All steps time: 0.13181500000000002 seconds

Process-2101, Find cars processing time: 0.131881 seconds

The times for each task are: [0.323607, 0.298424, 0.22868, 0.190186, 0.310725, 0.228371, 0.24574

Minimum: 0.122303 Maximum: 0.323607 Average: 0.1975 seconds

Number of processes used: 3 window size 260

Length of task list: 22

Number of processes used: 3

Process-2104, Step 1: Divide with 255, processing time: 0.030476 seconds

Process-2104, Step 2: Resize if scale is not 1: 0.002154 seconds

Process-2104, Step 3: Get HOG channels: 3.2e-05 seconds

Process-2104, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2104, Step 5: Compute individual channel HOG features for the entire image: 0.24463 seconds

Process-2104, Step 6: Misc initializations: 0.0 seconds

Process-2104, Step 7: for loop: 0.017265 seconds

Process-2104All steps time: 0.29456299999999996 seconds

Process-2104, Find cars processing time: 0.294751 seconds

Process-2105, Step 1: Divide with 255, processing time: 0.030864 seconds

Process-2105, Step 2: Resize if scale is not 1: 0.002138 seconds

Process-2105, Step 3: Get HOG channels: 2.2e-05 seconds

Process-2105, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2105, Step 5: Compute individual channel HOG features for the entire image: 0.19135 seconds

Process-2105, Step 6: Misc initializations: 0.0 seconds

Process-2105, Step 7: for loop: 0.013556 seconds

Process-2105All steps time: 0.237936 seconds

Process-2105, Find cars processing time: 0.238091 seconds

Process-2106, Step 1: Divide with 255, processing time: 0.050018 seconds

Process-2106, Step 2: Resize if scale is not 1: 0.003275 seconds

Process-2106, Step 3: Get HOG channels: 4.2e-05 seconds

Process-2106, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2106, Step 5: Compute individual channel HOG features for the entire image: 0.231915 seconds

Process-2106, Step 6: Misc initializations: 0.0 seconds

Process-2106, Step 7: for loop: 0.014753 seconds

Process-2106All steps time: 0.300009 seconds

Process-2106, Find cars processing time: 0.300194 seconds

Process-2104, Step 1: Divide with 255, processing time: 0.00969 seconds

Process-2104, Step 2: Resize if scale is not 1: 0.000759 seconds

Process-2104, Step 3: Get HOG channels: 8e-06 seconds

Process-2104, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2104, Step 5: Compute individual channel HOG features for the entire image: 0.241547 seconds

Process-2104, Step 6: Misc initializations: 1e-06 seconds

Process-2104, Step 7: for loop: 0.016132 seconds

Process-2104All steps time: 0.2681439999999994 seconds

Process-2104, Find cars processing time: 0.268219 seconds

Process-2105, Step 1: Divide with 255, processing time: 0.006703 seconds

Process-2105, Step 2: Resize if scale is not 1: 0.000726 seconds

Process-2105, Step 3: Get HOG channels: 7e-06 seconds

Process-2105, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2105, Step 5: Compute individual channel HOG features for the entire image: 0.206299 seconds

Process-2105, Step 6: Misc initializations: 1e-06 seconds

Process-2105, Step 7: for loop: 0.013748 seconds

Process-2105All steps time: 0.2274890000000002 seconds

Process-2105, Find cars processing time: 0.227562 seconds

Process-2106, Step 1: Divide with 255, processing time: 0.010147 seconds

Process-2106, Step 2: Resize if scale is not 1: 0.000931 seconds

Process-2106, Step 3: Get HOG channels: 8e-06 seconds

Process-2106, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2106, Step 5: Compute individual channel HOG features for the entire image: 0.202439 seconds

Process-2106, Step 6: Misc initializations: 0.0 seconds

Process-2106, Step 7: for loop: 0.012524 seconds

Process-2106All steps time: 0.226056 seconds

Process-2106, Find cars processing time: 0.22612 seconds

Process-2104, Step 1: Divide with 255, processing time: 0.010299 seconds

Process-2104, Step 2: Resize if scale is not 1: 0.000826 seconds

Process-2104, Step 3: Get HOG channels: 6e-06 seconds

Process-2104, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2104, Step 5: Compute individual channel HOG features for the entire image: 0.202309 seconds

Process-2104, Step 6: Misc initializations: 1e-06 seconds

Process-2104, Step 7: for loop: 0.011198 seconds

Process-2104All steps time: 0.22464299999999998 seconds

Process-2104, Find cars processing time: 0.2247 seconds

Process-2106, Step 1: Divide with 255, processing time: 0.007859 seconds

Process-2106, Step 2: Resize if scale is not 1: 0.000694 seconds

Process-2106, Step 3: Get HOG channels: 1.8e-05 seconds

Process-2106, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2106, Step 5: Compute individual channel HOG features for the entire image: 0.16987 seconds

Process-2106, Step 6: Misc initializations: 0.0 seconds

Process-2106, Step 7: for loop: 0.009558 seconds

Process-2106All steps time: 0.188006 seconds

Process-2106, Find cars processing time: 0.18807 seconds

Process-2105, Step 1: Divide with 255, processing time: 0.011008 seconds

Process-2105, Step 2: Resize if scale is not 1: 0.000984 seconds

Process-2105, Step 3: Get HOG channels: 8e-06 seconds

Process-2105, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2105, Step 5: Compute individual channel HOG features for the entire image: 0.198145 seconds

Process-2105, Step 6: Misc initializations: 1e-06 seconds

Process-2105, Step 7: for loop: 0.012717 seconds

Process-2105All steps time: 0.2228709999999999 seconds

Process-2105, Find cars processing time: 0.22295 seconds

Process-2104, Step 1: Divide with 255, processing time: 0.010038 seconds

Process-2104, Step 2: Resize if scale is not 1: 0.00076 seconds

Process-2104, Step 3: Get HOG channels: 7e-06 seconds

Process-2104, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2104, Step 5: Compute individual channel HOG features for the entire image: 0.178321 seconds

Process-2104, Step 6: Misc initializations: 1e-06 seconds

Process-2104, Step 7: for loop: 0.009839 seconds

Process-2104All steps time: 0.1989720000000004 seconds

Process-2104, Find cars processing time: 0.199041 seconds

Process-2106, Step 1: Divide with 255, processing time: 0.007312 seconds

Process-2106, Step 2: Resize if scale is not 1: 0.000545 seconds

Process-2106, Step 3: Get HOG channels: 5e-06 seconds

Process-2106, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2106, Step 5: Compute individual channel HOG features for the entire image: 0.170115 seconds

Process-2106, Step 6: Misc initializations: 1e-06 seconds

Process-2106, Step 7: for loop: 0.009656 seconds

Process-2106All steps time: 0.18764 seconds

Process-2106, Find cars processing time: 0.187714 seconds

Process-2105, Step 1: Divide with 255, processing time: 0.010683 seconds

Process-2105, Step 2: Resize if scale is not 1: 0.000953 seconds

Process-2105, Step 3: Get HOG channels: 8e-06 seconds

Process-2105, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2105, Step 5: Compute individual channel HOG features for the entire image: 0.185578 seconds

Process-2105, Step 6: Misc initializations: 1e-06 seconds

Process-2105, Step 7: for loop: 0.010053 seconds

Process-2105All steps time: 0.207283 seconds

Process-2105, Find cars processing time: 0.207362 seconds

Process-2105, Step 1: Divide with 255, processing time: 0.009879 seconds

Process-2105, Step 2: Resize if scale is not 1: 0.00071 seconds

Process-2105, Step 3: Get HOG channels: 7e-06 seconds

Process-2105, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2105, Step 5: Compute individual channel HOG features for the entire image: 0.165544 seconds

Process-2105, Step 6: Misc initializations: 1e-06 seconds

Process-2105, Step 7: for loop: 0.00759 seconds

Process-2105All steps time: 0.183737 seconds

Process-2105, Find cars processing time: 0.183815 seconds

Process-2104, Step 1: Divide with 255, processing time: 0.010745 seconds

Process-2104, Step 2: Resize if scale is not 1: 0.000908 seconds

Process-2104, Step 3: Get HOG channels: 9e-06 seconds

Process-2104, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2104, Step 5: Compute individual channel HOG features for the entire image: 0.146036 seconds

Process-2104, Step 6: Misc initializations: 1e-06 seconds

Process-2104, Step 7: for loop: 0.00724 seconds

Process-2104All steps time: 0.1649479999999998 seconds

Process-2104, Find cars processing time: 0.165024 seconds

Process-2106, Step 1: Divide with 255, processing time: 0.011009 seconds

Process-2106, Step 2: Resize if scale is not 1: 0.000905 seconds

Process-2106, Step 3: Get HOG channels: 8e-06 seconds

Process-2106, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2106, Step 5: Compute individual channel HOG features for the entire image: 0.157221 seconds

Process-2106, Step 6: Misc initializations: 1e-06 seconds

Process-2106, Step 7: for loop: 0.009648 seconds

Process-2106All steps time: 0.1788 seconds

Process-2105, Step 1: Divide with 255, processing time: 0.010182 seconds

Process-2106, Find cars processing time: 0.178907 seconds

Process-2105, Step 2: Resize if scale is not 1: 0.000568 seconds

Process-2105, Step 3: Get HOG channels: 6e-06 seconds

Process-2105, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2105, Step 5: Compute individual channel HOG features for the entire image: 0.147171 seconds

Process-2105, Step 6: Misc initializations: 1e-06 seconds

Process-2105, Step 7: for loop: 0.007467 seconds

Process-2105All steps time: 0.1654 seconds

Process-2105, Find cars processing time: 0.165478 seconds

Process-2106, Step 1: Divide with 255, processing time: 0.010063 seconds

Process-2106, Step 2: Resize if scale is not 1: 0.001158 seconds

Process-2106, Step 3: Get HOG channels: 8e-06 seconds

Process-2106, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2106, Step 5: Compute individual channel HOG features for the entire image: 0.149661 seconds

Process-2106, Step 6: Misc initializations: 0.0 seconds

Process-2106, Step 7: for loop: 0.007674 seconds

Process-2106All steps time: 0.1685709999999997 seconds

Process-2106, Find cars processing time: 0.168647 seconds

Process-2104, Step 1: Divide with 255, processing time: 0.008545 seconds

Process-2104, Step 2: Resize if scale is not 1: 0.000772 seconds

Process-2104, Step 3: Get HOG channels: 1e-05 seconds

Process-2104, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2104, Step 5: Compute individual channel HOG features for the entire image: 0.12257 seconds

Process-2104, Step 6: Misc initializations: 1e-06 seconds

Process-2104, Step 7: for loop: 0.006904 seconds

Process-2104All steps time: 0.13881 seconds

Process-2104, Find cars processing time: 0.138893 seconds

Process-2105, Step 1: Divide with 255, processing time: 0.009099 seconds

Process-2105, Step 2: Resize if scale is not 1: 0.000998 seconds

Process-2105, Step 3: Get HOG channels: 8e-06 seconds

Process-2105, Step 6: Misc initializations: 1e-06 seconds

Process-2105, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2105, Step 5: Compute individual channel HOG features for the entire image: 0.153591 seconds

Process-2105, Step 7: for loop: 0.007411 seconds

Process-2105All steps time: 0.17111600000000002 seconds

Process-2105, Find cars processing time: 0.171201 seconds

Process-2106, Step 1: Divide with 255, processing time: 0.009852 seconds

Process-2105, Step 1: Divide with 255, processing time: 0.010869 seconds

Process-2106, Step 2: Resize if scale is not 1: 0.00064 seconds

Process-2105, Step 2: Resize if scale is not 1: 0.002055 seconds

Process-2106, Step 3: Get HOG channels: 7e-06 seconds

Process-2106, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2105, Step 3: Get HOG channels: 7e-06 seconds

Process-2106, Step 5: Compute individual channel HOG features for the entire image: 0.132869 seconds

Process-2105, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2106, Step 6: Misc initializations: 1e-06 seconds

Process-2106, Step 7: for loop: 0.007204 seconds

Process-2105, Step 5: Compute individual channel HOG features for the entire image: 0.119356 seconds

Process-2106All steps time: 0.1505799999999996 seconds

Process-2105, Step 6: Misc initializations: 1e-06 seconds

Process-2105, Step 7: for loop: 0.006213 seconds

Process-2105All steps time: 0.138508 seconds

Process-2105, Find cars processing time: 0.138593 seconds

Process-2106, Find cars processing time: 0.150653 seconds

Process-2104, Step 1: Divide with 255, processing time: 0.009886 seconds

Process-2104, Step 2: Resize if scale is not 1: 0.001807 seconds

Process-2104, Step 3: Get HOG channels: 9e-06 seconds

Process-2104, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2104, Step 5: Compute individual channel HOG features for the entire image: 0.158243 seconds

Process-2104, Step 6: Misc initializations: 1e-06 seconds

Process-2104, Step 7: for loop: 0.007262 seconds

Process-2104All steps time: 0.1772169999999999 seconds

Process-2104, Find cars processing time: 0.177301 seconds

The times for each task are: [0.294751, 0.238091, 0.300194, 0.268219, 0.227562, 0.22612, 0.2247,

Minimum: 0.138593 Maximum: 0.300194 Average: 0.2011 seconds

Number of processes used: 3 window size 260

Length of task list: 22
Number of processes used: 3

Process-2107, Step 1: Divide with 255, processing time: 0.026901 seconds

Process-2107, Step 2: Resize if scale is not 1: 0.002788 seconds

Process-2107, Step 3: Get HOG channels: 1.9e-05 seconds

Process-2107, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2107, Step 5: Compute individual channel HOG features for the entire image: 0.293 seconds

Process-2107, Step 6: Misc initializations: 1e-06 seconds

Process-2107, Step 7: for loop: 0.018132 seconds

Process-2107All steps time: 0.3408489999999996 seconds

Process-2107, Find cars processing time: 0.341033 seconds

Process-2107, Step 1: Divide with 255, processing time: 0.010159 seconds

Process-2107, Step 2: Resize if scale is not 1: 0.00088 seconds

Process-2107, Step 3: Get HOG channels: 9e-06 seconds

Process-2107, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2107, Step 5: Compute individual channel HOG features for the entire image: 0.223127 seconds

Process-2107, Step 6: Misc initializations: 0.0 seconds

Process-2107, Step 7: for loop: 0.013709 seconds

Process-2107All steps time: 0.247893 seconds

Process-2107, Find cars processing time: 0.247964 seconds

Process-2108, Step 1: Divide with 255, processing time: 0.02444 seconds

Process-2108, Step 2: Resize if scale is not 1: 0.001868 seconds

Process-2108, Step 3: Get HOG channels: 1.6e-05 seconds

Process-2108, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2108, Step 5: Compute individual channel HOG features for the entire image: 0.237237 seconds

Process-2108, Step 6: Misc initializations: 1e-06 seconds

Process-2108, Step 7: for loop: 0.013763 seconds

Process-2108All steps time: 0.27733 seconds

Process-2108, Find cars processing time: 0.277459 seconds

Process-2107, Step 1: Divide with 255, processing time: 0.007762 seconds

Process-2107, Step 2: Resize if scale is not 1: 0.000707 seconds

Process-2107, Step 3: Get HOG channels: 7e-06 seconds

Process-2107, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2107, Step 5: Compute individual channel HOG features for the entire image: 0.188013 seconds

Process-2107, Step 6: Misc initializations: 0.0 seconds

Process-2107, Step 7: for loop: 0.01053 seconds

Process-2107All steps time: 0.2070260000000002 seconds

Process-2107, Find cars processing time: 0.207108 seconds

Process-2109, Step 1: Divide with 255, processing time: 0.028143 seconds

Process-2109, Step 2: Resize if scale is not 1: 0.001899 seconds

Process-2109, Step 5: Compute individual channel HOG features for the entire image: 0.275299 seconds

Process-2109, Step 3: Get HOG channels: 1.7e-05 seconds

Process-2109, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2109, Step 6: Misc initializations: 1e-06 seconds

Process-2109, Step 7: for loop: 0.021243 seconds

Process-2109All steps time: 0.326608 seconds

Process-2109, Find cars processing time: 0.326758 seconds

Process-2108, Step 1: Divide with 255, processing time: 0.009451 seconds

Process-2108, Step 2: Resize if scale is not 1: 0.000737 seconds

Process-2108, Step 3: Get HOG channels: 7e-06 seconds

Process-2108, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2108, Step 5: Compute individual channel HOG features for the entire image: 0.158062 seconds

Process-2108, Step 6: Misc initializations: 0.0 seconds

Process-2108, Step 7: for loop: 0.01 seconds

Process-2108All steps time: 0.178264 seconds

Process-2108, Find cars processing time: 0.178323 seconds

Process-2109, Step 1: Divide with 255, processing time: 0.010406 seconds

Process-2109, Step 2: Resize if scale is not 1: 0.000856 seconds

Process-2109, Step 3: Get HOG channels: 8e-06 seconds

Process-2109, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2109, Step 5: Compute individual channel HOG features for the entire image: 0.197993 seconds

Process-2109, Step 6: Misc initializations: 0.0 seconds

Process-2109, Step 7: for loop: 0.012529 seconds

Process-2109All steps time: 0.2217990000000002 seconds

Process-2109, Find cars processing time: 0.221868 seconds

Process-2108, Step 1: Divide with 255, processing time: 0.008296 seconds

Process-2108, Step 2: Resize if scale is not 1: 0.000713 seconds

Process-2108, Step 3: Get HOG channels: 7e-06 seconds

Process-2108, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2108, Step 5: Compute individual channel HOG features for the entire image: 0.131486 seconds

Process-2108, Step 6: Misc initializations: 0.0 seconds

Process-2108, Step 7: for loop: 0.007454 seconds

Process-2108All steps time: 0.1479629999999998 seconds

Process-2108, Find cars processing time: 0.148021 seconds

Process-2107, Step 1: Divide with 255, processing time: 0.009781 seconds

Process-2107, Step 2: Resize if scale is not 1: 0.000713 seconds

Process-2107, Step 3: Get HOG channels: 8e-06 seconds

Process-2107, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2107, Step 5: Compute individual channel HOG features for the entire image: 0.18246 seconds

Process-2107, Step 6: Misc initializations: 0.0 seconds

Process-2107, Step 7: for loop: 0.010556 seconds

Process-2107All steps time: 0.203525 seconds

Process-2107, Find cars processing time: 0.203589 seconds

Process-2107, Step 1: Divide with 255, processing time: 0.007483 seconds

Process-2107, Step 2: Resize if scale is not 1: 0.000682 seconds

Process-2107, Step 3: Get HOG channels: 7e-06 seconds

Process-2107, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2107, Step 5: Compute individual channel HOG features for the entire image: 0.163175 seconds

Process-2107, Step 6: Misc initializations: 1e-06 seconds

Process-2107, Step 7: for loop: 0.008249 seconds

Process-2107All steps time: 0.1796039999999999 seconds

Process-2107, Find cars processing time: 0.179674 seconds

Process-2109, Step 1: Divide with 255, processing time: 0.010196 seconds

Process-2109, Step 2: Resize if scale is not 1: 0.00083 seconds

Process-2109, Step 3: Get HOG channels: 8e-06 seconds

Process-2109, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2109, Step 5: Compute individual channel HOG features for the entire image: 0.130949 seconds

Process-2109, Step 6: Misc initializations: 1e-06 seconds

Process-2109, Step 7: for loop: 0.008223 seconds

Process-2109All steps time: 0.1502140000000001 seconds

Process-2109, Find cars processing time: 0.15028 seconds

Process-2107, Step 1: Divide with 255, processing time: 0.010357 seconds

Process-2107, Step 2: Resize if scale is not 1: 0.000801 seconds

Process-2108, Step 1: Divide with 255, processing time: 0.010534 seconds

Process-2107, Step 3: Get HOG channels: 5e-06 seconds

Process-2108, Step 2: Resize if scale is not 1: 0.000542 seconds

Process-2108, Step 3: Get HOG channels: 4e-06 seconds

Process-2108, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2107, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2108, Step 5: Compute individual channel HOG features for the entire image: 0.138654 seconds

Process-2107, Step 5: Compute individual channel HOG features for the entire image: 0.146995 seconds
Process-2108, Step 6: Misc initializations: 0.0 seconds
Process-2107, Step 6: Misc initializations: 0.0 seconds
Process-2108, Step 7: for loop: 0.007958 seconds
Process-2107, Step 7: for loop: 0.006351 seconds

Process-2107All steps time: 0.164514 seconds

Process-2107, Find cars processing time: 0.164581 seconds

Process-2108All steps time: 0.157697 seconds

Process-2108, Find cars processing time: 0.157743 seconds
Process-2108, Step 1: Divide with 255, processing time: 0.008394 seconds
Process-2108, Step 2: Resize if scale is not 1: 0.000651 seconds
Process-2108, Step 3: Get HOG channels: 6e-06 seconds
Process-2108, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2108, Step 5: Compute individual channel HOG features for the entire image: 0.115943 seconds
Process-2108, Step 6: Misc initializations: 1e-06 seconds
Process-2108, Step 7: for loop: 0.005954 seconds

Process-2108All steps time: 0.130956 seconds

Process-2108, Find cars processing time: 0.131012 seconds

Process-2107, Step 1: Divide with 255, processing time: 0.009141 seconds
Process-2107, Step 2: Resize if scale is not 1: 0.000794 seconds
Process-2107, Step 3: Get HOG channels: 8e-06 seconds
Process-2107, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2107, Step 5: Compute individual channel HOG features for the entire image: 0.135602 seconds
Process-2107, Step 6: Misc initializations: 1e-06 seconds
Process-2107, Step 7: for loop: 0.006288 seconds

Process-2107All steps time: 0.1518409999999998 seconds

Process-2107, Find cars processing time: 0.151906 seconds

Process-2109, Step 1: Divide with 255, processing time: 0.009168 seconds
Process-2109, Step 2: Resize if scale is not 1: 0.000674 seconds
Process-2109, Step 3: Get HOG channels: 6e-06 seconds
Process-2109, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2109, Step 5: Compute individual channel HOG features for the entire image: 0.114714 seconds
Process-2109, Step 6: Misc initializations: 0.0 seconds
Process-2109, Step 7: for loop: 0.005558 seconds

Process-2109All steps time: 0.130126 seconds

Process-2109, Find cars processing time: 0.130182 seconds

Process-2107, Step 1: Divide with 255, processing time: 0.008996 seconds

Process-2107, Step 2: Resize if scale is not 1: 0.001129 seconds

Process-2107, Step 3: Get HOG channels: 8e-06 seconds

Process-2107, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2107, Step 5: Compute individual channel HOG features for the entire image: 0.128518 seconds

Process-2107, Step 6: Misc initializations: 0.0 seconds

Process-2107, Step 7: for loop: 0.00742 seconds

Process-2107All steps time: 0.14607900000000001 seconds

Process-2107, Find cars processing time: 0.146143 seconds

Process-2108, Step 1: Divide with 255, processing time: 0.010606 seconds

Process-2108, Step 2: Resize if scale is not 1: 0.000506 seconds

Process-2108, Step 3: Get HOG channels: 6e-06 seconds

Process-2108, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2108, Step 5: Compute individual channel HOG features for the entire image: 0.119803 seconds

Process-2108, Step 6: Misc initializations: 0.0 seconds

Process-2108, Step 7: for loop: 0.00507 seconds

Process-2108All steps time: 0.135997 seconds

Process-2108, Find cars processing time: 0.136056 seconds

Process-2109, Step 1: Divide with 255, processing time: 0.010737 seconds

Process-2109, Step 2: Resize if scale is not 1: 0.000799 seconds

Process-2109, Step 3: Get HOG channels: 6e-06 seconds

Process-2109, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2109, Step 5: Compute individual channel HOG features for the entire image: 0.115628 seconds

Process-2109, Step 6: Misc initializations: 0.0 seconds

Process-2109, Step 7: for loop: 0.006209 seconds

Process-2109All steps time: 0.1333849999999998 seconds

Process-2109, Find cars processing time: 0.133451 seconds

Process-2108, Step 1: Divide with 255, processing time: 0.011291 seconds

Process-2108, Step 2: Resize if scale is not 1: 0.001012 seconds

Process-2108, Step 3: Get HOG channels: 6e-06 seconds

Process-2108, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2108, Step 5: Compute individual channel HOG features for the entire image: 0.13018 seconds

Process-2108, Step 6: Misc initializations: 1e-06 seconds

Process-2108, Step 7: for loop: 0.007082 seconds

Process-2108All steps time: 0.149579 seconds

Process-2108, Find cars processing time: 0.149651 seconds

Process-2107, Step 1: Divide with 255, processing time: 0.010817 seconds

Process-2107, Step 2: Resize if scale is not 1: 0.000802 seconds

Process-2107, Step 3: Get HOG channels: 8e-06 seconds

Process-2107, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2107, Step 5: Compute individual channel HOG features for the entire image: 0.142059 seconds

Process-2107, Step 6: Misc initializations: 1e-06 seconds

Process-2107, Step 7: for loop: 0.007179 seconds

Process-2107All steps time: 0.160875 seconds

Process-2107, Find cars processing time: 0.160939 seconds

Process-2109, Step 1: Divide with 255, processing time: 0.010063 seconds

Process-2109, Step 2: Resize if scale is not 1: 0.002398 seconds

Process-2109, Step 3: Get HOG channels: 1.3e-05 seconds

Process-2109, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2109, Step 5: Compute individual channel HOG features for the entire image: 0.135567 seconds

Process-2109, Step 6: Misc initializations: 1e-06 seconds

Process-2109, Step 7: for loop: 0.0064 seconds

Process-2109All steps time: 0.1544519999999998 seconds

Process-2109, Find cars processing time: 0.154548 seconds

The times for each task are: [0.341033, 0.247964, 0.277459, 0.207108, 0.326758, 0.178323, 0.2218

Minimum: 0.130182 Maximum: 0.341033 Average: 0.1863 seconds

Number of processes used: 3 window size 260

Length of task list: 22

Number of processes used: 3

Process-2110, Step 1: Divide with 255, processing time: 0.022291 seconds

Process-2110, Step 2: Resize if scale is not 1: 0.001731 seconds

Process-2110, Step 3: Get HOG channels: 1.7e-05 seconds

Process-2110, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2110, Step 5: Compute individual channel HOG features for the entire image: 0.231879 seconds

Process-2110, Step 6: Misc initializations: 1e-06 seconds

Process-2110, Step 7: for loop: 0.015169 seconds

Process-2110All steps time: 0.2710939999999995 seconds

Process-2110, Find cars processing time: 0.271259 seconds

Process-2110, Step 1: Divide with 255, processing time: 0.007032 seconds
Process-2110, Step 2: Resize if scale is not 1: 0.000637 seconds
Process-2110, Step 3: Get HOG channels: 6e-06 seconds
Process-2110, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2110, Step 5: Compute individual channel HOG features for the entire image: 0.226488 seconds
Process-2110, Step 6: Misc initializations: 1e-06 seconds
Process-2110, Step 7: for loop: 0.014851 seconds

Process-2110All steps time: 0.249022 seconds

Process-2110, Find cars processing time: 0.249115 seconds

Process-2111, Step 1: Divide with 255, processing time: 0.030624 seconds
Process-2111, Step 2: Resize if scale is not 1: 0.002237 seconds
Process-2111, Step 3: Get HOG channels: 2.5e-05 seconds
Process-2111, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2111, Step 5: Compute individual channel HOG features for the entire image: 0.250078 seconds
Process-2111, Step 6: Misc initializations: 1e-06 seconds
Process-2111, Step 7: for loop: 0.017616 seconds

Process-2111All steps time: 0.300587 seconds

Process-2111, Find cars processing time: 0.300788 seconds

Process-2112, Step 1: Divide with 255, processing time: 0.026208 seconds
Process-2112, Step 2: Resize if scale is not 1: 0.00207 seconds
Process-2112, Step 3: Get HOG channels: 2.3e-05 seconds
Process-2112, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2112, Step 5: Compute individual channel HOG features for the entire image: 0.251318 seconds
Process-2112, Step 6: Misc initializations: 2e-06 seconds
Process-2112, Step 7: for loop: 0.018 seconds

Process-2112All steps time: 0.297626 seconds

Process-2112, Find cars processing time: 0.297803 seconds

Process-2110, Step 1: Divide with 255, processing time: 0.008367 seconds
Process-2110, Step 2: Resize if scale is not 1: 0.000805 seconds
Process-2110, Step 3: Get HOG channels: 1.7e-05 seconds
Process-2110, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2110, Step 5: Compute individual channel HOG features for the entire image: 0.200167 seconds
Process-2110, Step 6: Misc initializations: 1e-06 seconds
Process-2110, Step 7: for loop: 0.011094 seconds

Process-2110All steps time: 0.220461 seconds

Process-2110, Find cars processing time: 0.220562 seconds

Process-2111, Step 1: Divide with 255, processing time: 0.009417 seconds
Process-2111, Step 2: Resize if scale is not 1: 0.000881 seconds
Process-2111, Step 3: Get HOG channels: 8e-06 seconds
Process-2111, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2111, Step 5: Compute individual channel HOG features for the entire image: 0.193266 seconds
Process-2111, Step 6: Misc initializations: 0.0 seconds
Process-2111, Step 7: for loop: 0.010188 seconds

Process-2111All steps time: 0.2137679999999999 seconds

Process-2111, Find cars processing time: 0.213847 seconds

Process-2112, Step 1: Divide with 255, processing time: 0.009388 seconds
Process-2112, Step 2: Resize if scale is not 1: 0.000609 seconds
Process-2112, Step 3: Get HOG channels: 6e-06 seconds
Process-2112, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2112, Step 5: Compute individual channel HOG features for the entire image: 0.199594 seconds
Process-2112, Step 6: Misc initializations: 1e-06 seconds
Process-2112, Step 7: for loop: 0.012929 seconds

Process-2112All steps time: 0.2225329999999998 seconds

Process-2112, Find cars processing time: 0.222613 seconds

Process-2111, Step 1: Divide with 255, processing time: 0.010473 seconds
Process-2111, Step 2: Resize if scale is not 1: 0.000578 seconds
Process-2111, Step 3: Get HOG channels: 6e-06 seconds
Process-2111, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2111, Step 5: Compute individual channel HOG features for the entire image: 0.171613 seconds
Process-2111, Step 6: Misc initializations: 1e-06 seconds
Process-2111, Step 7: for loop: 0.009887 seconds

Process-2111All steps time: 0.1925639999999999 seconds

Process-2111, Find cars processing time: 0.192635 seconds

Process-2110, Step 1: Divide with 255, processing time: 0.017037 seconds
Process-2110, Step 2: Resize if scale is not 1: 0.000499 seconds
Process-2110, Step 3: Get HOG channels: 5e-06 seconds
Process-2110, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2110, Step 5: Compute individual channel HOG features for the entire image: 0.198052 seconds
Process-2110, Step 6: Misc initializations: 1e-06 seconds
Process-2110, Step 7: for loop: 0.012492 seconds

Process-2110All steps time: 0.2280920000000002 seconds

Process-2110, Find cars processing time: 0.228165 seconds

Process-2112, Step 1: Divide with 255, processing time: 0.007553 seconds
Process-2112, Step 2: Resize if scale is not 1: 0.000635 seconds
Process-2112, Step 3: Get HOG channels: 7e-06 seconds
Process-2112, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2112, Step 5: Compute individual channel HOG features for the entire image: 0.184307 seconds
Process-2112, Step 6: Misc initializations: 1e-06 seconds
Process-2112, Step 7: for loop: 0.009748 seconds

Process-2112All steps time: 0.20225700000000002 seconds

Process-2112, Find cars processing time: 0.202329 seconds

Process-2111, Step 1: Divide with 255, processing time: 0.010946 seconds
Process-2111, Step 2: Resize if scale is not 1: 0.000875 seconds
Process-2111, Step 3: Get HOG channels: 9e-06 seconds
Process-2111, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2111, Step 5: Compute individual channel HOG features for the entire image: 0.180293 seconds
Process-2111, Step 6: Misc initializations: 1e-06 seconds
Process-2111, Step 7: for loop: 0.009608 seconds

Process-2111All steps time: 0.201741 seconds

Process-2111, Find cars processing time: 0.201827 seconds

Process-2112, Step 1: Divide with 255, processing time: 0.010654 seconds
Process-2112, Step 2: Resize if scale is not 1: 0.000752 seconds
Process-2112, Step 3: Get HOG channels: 5e-06 seconds
Process-2112, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2112, Step 5: Compute individual channel HOG features for the entire image: 0.163235 seconds
Process-2112, Step 6: Misc initializations: 1e-06 seconds
Process-2112, Step 7: for loop: 0.007383 seconds

Process-2112All steps time: 0.182036 seconds

Process-2112, Find cars processing time: 0.182115 seconds

Process-2110, Step 1: Divide with 255, processing time: 0.008892 seconds
Process-2110, Step 2: Resize if scale is not 1: 0.000641 seconds
Process-2110, Step 3: Get HOG channels: 6e-06 seconds
Process-2110, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-2111, Step 1: Divide with 255, processing time: 0.009168 seconds
Process-2110, Step 5: Compute individual channel HOG features for the entire image: 0.207565 seconds
Process-2111, Step 2: Resize if scale is not 1: 0.00062 seconds
Process-2110, Step 6: Misc initializations: 2e-06 seconds
Process-2111, Step 3: Get HOG channels: 6e-06 seconds
Process-2111, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2110, Step 7: for loop: 0.010818 seconds
Process-2111, Step 5: Compute individual channel HOG features for the entire image: 0.162238 seconds

Process-2111, Step 6: Misc initializations: 1e-06 seconds
Process-2111, Step 7: for loop: 0.009404 seconds

Process-2111All steps time: 0.181443 seconds

Process-2110All steps time: 0.227936 seconds
Process-2111, Find cars processing time: 0.181515 seconds

Process-2110, Find cars processing time: 0.228013 seconds

Process-2110, Step 1: Divide with 255, processing time: 0.010642 seconds
Process-2110, Step 2: Resize if scale is not 1: 0.000906 seconds
Process-2110, Step 3: Get HOG channels: 1.5e-05 seconds
Process-2110, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2110, Step 5: Compute individual channel HOG features for the entire image: 0.126088 seconds
Process-2110, Step 6: Misc initializations: 1e-06 seconds
Process-2110, Step 7: for loop: 0.004481 seconds

Process-2110All steps time: 0.14214300000000002 seconds

Process-2110, Find cars processing time: 0.142252 seconds

Process-2112, Step 1: Divide with 255, processing time: 0.011581 seconds
Process-2112, Step 2: Resize if scale is not 1: 0.000569 seconds
Process-2112, Step 3: Get HOG channels: 6e-06 seconds
Process-2112, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2112, Step 5: Compute individual channel HOG features for the entire image: 0.147803 seconds
Process-2112, Step 6: Misc initializations: 1e-06 seconds
Process-2112, Step 7: for loop: 0.007262 seconds

Process-2112All steps time: 0.167228 seconds

Process-2112, Find cars processing time: 0.167291 seconds

Process-2111, Step 1: Divide with 255, processing time: 0.010671 seconds
Process-2111, Step 2: Resize if scale is not 1: 0.000791 seconds
Process-2111, Step 3: Get HOG channels: 8e-06 seconds
Process-2111, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2111, Step 5: Compute individual channel HOG features for the entire image: 0.14891 seconds
Process-2111, Step 6: Misc initializations: 1e-06 seconds
Process-2111, Step 7: for loop: 0.007658 seconds

Process-2111All steps time: 0.1680479999999998 seconds

Process-2111, Find cars processing time: 0.168131 seconds

Process-2110, Step 1: Divide with 255, processing time: 0.01031 seconds
Process-2110, Step 2: Resize if scale is not 1: 0.001087 seconds
Process-2110, Step 3: Get HOG channels: 5e-06 seconds
Process-2110, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2110, Step 5: Compute individual channel HOG features for the entire image: 0.115343 seconds
Process-2110, Step 6: Misc initializations: 0.0 seconds
Process-2110, Step 7: for loop: 0.006187 seconds

Process-2110All steps time: 0.132936 seconds

Process-2110, Find cars processing time: 0.132998 seconds

Process-2112, Step 1: Divide with 255, processing time: 0.007836 seconds
Process-2112, Step 2: Resize if scale is not 1: 0.000889 seconds
Process-2112, Step 3: Get HOG channels: 7e-06 seconds
Process-2112, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2112, Step 5: Compute individual channel HOG features for the entire image: 0.144115 seconds
Process-2112, Step 6: Misc initializations: 1e-06 seconds
Process-2112, Step 7: for loop: 0.008827 seconds

Process-2112All steps time: 0.161683 seconds

Process-2112, Find cars processing time: 0.161781 seconds

Process-2110, Step 1: Divide with 255, processing time: 0.010375 seconds
Process-2110, Step 2: Resize if scale is not 1: 0.00058 seconds
Process-2110, Step 3: Get HOG channels: 8e-06 seconds
Process-2110, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2110, Step 5: Compute individual channel HOG features for the entire image: 0.126922 seconds
Process-2110, Step 6: Misc initializations: 0.0 seconds
Process-2110, Step 7: for loop: 0.006254 seconds

Process-2110All steps time: 0.1441460000000002 seconds

Process-2110, Find cars processing time: 0.144232 seconds

Process-2111, Step 1: Divide with 255, processing time: 0.029805 seconds
Process-2111, Step 2: Resize if scale is not 1: 0.000995 seconds
Process-2111, Step 3: Get HOG channels: 6e-06 seconds
Process-2111, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2111, Step 5: Compute individual channel HOG features for the entire image: 0.1305 seconds
Process-2111, Step 6: Misc initializations: 1e-06 seconds
Process-2111, Step 7: for loop: 0.006987 seconds

Process-2111All steps time: 0.168299 seconds

Process-2111, Find cars processing time: 0.168359 seconds

Process-2112, Step 1: Divide with 255, processing time: 0.0107 seconds
Process-2112, Step 2: Resize if scale is not 1: 0.001359 seconds
Process-2112, Step 3: Get HOG channels: 7e-06 seconds
Process-2112, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2112, Step 5: Compute individual channel HOG features for the entire image: 0.123976 seconds
Process-2112, Step 6: Misc initializations: 1e-06 seconds
Process-2112, Step 7: for loop: 0.006253 seconds

Process-2112All steps time: 0.142303 seconds

Process-2112, Find cars processing time: 0.142389 seconds

The times for each task are: [0.271259, 0.249115, 0.300788, 0.297803, 0.220562, 0.213847, 0.222661]

Minimum: 0.132998 Maximum: 0.300788 Average: 0.2009 seconds

Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]

3 processes used for testing 5 window sizes

Processing times for each image [3.9165, 4.3203, 4.0393, 4.6336, 4.3872, 4.6724] with an average

Time elapsed so far... 402.7339000000000006

#####

Number of processes used: 4 window size 260

Length of task list: 22

Number of processes used: 4

Process-2114, Step 1: Divide with 255, processing time: 0.02932 seconds
Process-2114, Step 2: Resize if scale is not 1: 0.002676 seconds
Process-2114, Step 3: Get HOG channels: 2.6e-05 seconds
Process-2114, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2114, Step 5: Compute individual channel HOG features for the entire image: 0.239983 seconds
Process-2114, Step 6: Misc initializations: 1.7e-05 seconds
Process-2114, Step 7: for loop: 0.014867 seconds

Process-2114All steps time: 0.286896 seconds

Process-2114, Find cars processing time: 0.28706 seconds

Process-2113, Step 1: Divide with 255, processing time: 0.027832 seconds
Process-2113, Step 2: Resize if scale is not 1: 0.002633 seconds
Process-2113, Step 3: Get HOG channels: 2.5e-05 seconds
Process-2113, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2113, Step 5: Compute individual channel HOG features for the entire image: 0.296783 seconds
Process-2113, Step 6: Misc initializations: 1e-06 seconds
Process-2113, Step 7: for loop: 0.017769 seconds

Process-2113All steps time: 0.34505199999999997 seconds

Process-2113, Find cars processing time: 0.345252 seconds

Process-2115, Step 1: Divide with 255, processing time: 0.028298 seconds

Process-2115, Step 2: Resize if scale is not 1: 0.001927 seconds

Process-2115, Step 3: Get HOG channels: 2e-05 seconds

Process-2115, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2115, Step 5: Compute individual channel HOG features for the entire image: 0.255599 seconds

Process-2115, Step 6: Misc initializations: 1e-06 seconds

Process-2115, Step 7: for loop: 0.025312 seconds

Process-2115All steps time: 0.311163 seconds

Process-2115, Find cars processing time: 0.311355 seconds

Process-2114, Step 1: Divide with 255, processing time: 0.011146 seconds

Process-2114, Step 2: Resize if scale is not 1: 0.000954 seconds

Process-2114, Step 3: Get HOG channels: 8e-06 seconds

Process-2114, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2114, Step 5: Compute individual channel HOG features for the entire image: 0.187365 seconds

Process-2114, Step 6: Misc initializations: 0.0 seconds

Process-2114, Step 7: for loop: 0.012578 seconds

Process-2114All steps time: 0.21206 seconds

Process-2114, Find cars processing time: 0.212128 seconds

Process-2116, Step 1: Divide with 255, processing time: 0.02914 seconds

Process-2116, Step 2: Resize if scale is not 1: 0.001969 seconds

Process-2116, Step 3: Get HOG channels: 2.1e-05 seconds

Process-2116, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2116, Step 5: Compute individual channel HOG features for the entire image: 0.199525 seconds

Process-2116, Step 6: Misc initializations: 1e-06 seconds

Process-2116, Step 7: for loop: 0.01308 seconds

Process-2116All steps time: 0.243741 seconds

Process-2116, Find cars processing time: 0.243895 seconds

Process-2114, Step 1: Divide with 255, processing time: 0.010284 seconds

Process-2114, Step 2: Resize if scale is not 1: 0.000898 seconds

Process-2115, Step 1: Divide with 255, processing time: 0.010342 seconds

Process-2115, Step 2: Resize if scale is not 1: 0.000818 seconds

Process-2114, Step 3: Get HOG channels: 8e-06 seconds

Process-2115, Step 3: Get HOG channels: 8e-06 seconds

Process-2115, Step 4: Define blocks and steps as above: 1.7e-05 seconds

Process-2114, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2115, Step 5: Compute individual channel HOG features for the entire image: 0.233324 seconds
Process-2114, Step 5: Compute individual channel HOG features for the entire image: 0.198665 seconds
Process-2115, Step 6: Misc initializations: 1e-06 seconds
Process-2114, Step 6: Misc initializations: 1e-06 seconds
Process-2115, Step 7: for loop: 0.013546 seconds

Process-2114, Step 7: for loop: 0.012597 seconds
Process-2115All steps time: 0.258056 seconds

Process-2114All steps time: 0.222461 seconds
Process-2115, Find cars processing time: 0.258125 seconds

Process-2114, Find cars processing time: 0.222527 seconds

Process-2113, Step 1: Divide with 255, processing time: 0.010234 seconds
Process-2113, Step 2: Resize if scale is not 1: 0.000883 seconds
Process-2113, Step 3: Get HOG channels: 8e-06 seconds
Process-2113, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2113, Step 5: Compute individual channel HOG features for the entire image: 0.199523 seconds
Process-2113, Step 6: Misc initializations: 1e-06 seconds
Process-2113, Step 7: for loop: 0.012736 seconds

Process-2113All steps time: 0.223392 seconds

Process-2113, Find cars processing time: 0.223463 seconds

Process-2116, Step 1: Divide with 255, processing time: 0.009645 seconds
Process-2116, Step 2: Resize if scale is not 1: 0.000599 seconds
Process-2116, Step 3: Get HOG channels: 6e-06 seconds
Process-2116, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2116, Step 5: Compute individual channel HOG features for the entire image: 0.129972 seconds
Process-2116, Step 6: Misc initializations: 1e-06 seconds
Process-2116, Step 7: for loop: 0.007665 seconds

Process-2116All steps time: 0.14789400000000003 seconds

Process-2116, Find cars processing time: 0.147947 seconds

Process-2115, Step 1: Divide with 255, processing time: 0.010691 seconds
Process-2115, Step 2: Resize if scale is not 1: 0.000915 seconds
Process-2115, Step 3: Get HOG channels: 6e-06 seconds
Process-2115, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2115, Step 5: Compute individual channel HOG features for the entire image: 0.191514 seconds
Process-2115, Step 6: Misc initializations: 0.0 seconds
Process-2115, Step 7: for loop: 0.008403 seconds

Process-2115All steps time: 0.211534 seconds

Process-2115, Find cars processing time: 0.211607 seconds

Process-2113, Step 1: Divide with 255, processing time: 0.006511 seconds

Process-2113, Step 2: Resize if scale is not 1: 0.000598 seconds

Process-2113, Step 3: Get HOG channels: 5e-06 seconds

Process-2113, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2113, Step 5: Compute individual channel HOG features for the entire image: 0.143023 seconds

Process-2113, Step 6: Misc initializations: 1e-06 seconds

Process-2113, Step 7: for loop: 0.007866 seconds

Process-2113All steps time: 0.15801 seconds

Process-2113, Find cars processing time: 0.158066 seconds

Process-2116, Step 1: Divide with 255, processing time: 0.010396 seconds

Process-2116, Step 2: Resize if scale is not 1: 0.000526 seconds

Process-2116, Step 3: Get HOG channels: 5e-06 seconds

Process-2116, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2116, Step 5: Compute individual channel HOG features for the entire image: 0.142092 seconds

Process-2116, Step 6: Misc initializations: 0.0 seconds

Process-2116, Step 7: for loop: 0.006978 seconds

Process-2116All steps time: 0.160003 seconds

Process-2116, Find cars processing time: 0.160064 seconds

Process-2114, Step 1: Divide with 255, processing time: 0.012791 seconds

Process-2114, Step 2: Resize if scale is not 1: 0.000469 seconds

Process-2114, Step 3: Get HOG channels: 5e-06 seconds

Process-2114, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2114, Step 5: Compute individual channel HOG features for the entire image: 0.175284 seconds

Process-2114, Step 6: Misc initializations: 1e-06 seconds

Process-2114, Step 7: for loop: 0.00837 seconds

Process-2114All steps time: 0.196925 seconds

Process-2114, Find cars processing time: 0.196982 seconds

Process-2114, Step 1: Divide with 255, processing time: 0.010217 seconds

Process-2114, Step 2: Resize if scale is not 1: 0.000665 seconds

Process-2114, Step 3: Get HOG channels: 8e-06 seconds

Process-2114, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2114, Step 5: Compute individual channel HOG features for the entire image: 0.111791 seconds

Process-2114, Step 6: Misc initializations: 1e-06 seconds

Process-2114, Step 7: for loop: 0.005157 seconds

Process-2114All steps time: 0.12784800000000002 seconds

Process-2114, Find cars processing time: 0.12793 seconds

Process-2115, Step 1: Divide with 255, processing time: 0.010201 seconds

Process-2115, Step 2: Resize if scale is not 1: 0.000773 seconds

Process-2115, Step 3: Get HOG channels: 7e-06 seconds

Process-2115, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2115, Step 5: Compute individual channel HOG features for the entire image: 0.154552 seconds

Process-2115, Step 6: Misc initializations: 1e-06 seconds

Process-2115, Step 7: for loop: 0.007196 seconds

Process-2115All steps time: 0.172738 seconds

Process-2115, Find cars processing time: 0.172812 seconds

Process-2113, Step 1: Divide with 255, processing time: 0.010408 seconds

Process-2113, Step 2: Resize if scale is not 1: 0.000692 seconds

Process-2113, Step 3: Get HOG channels: 5e-06 seconds

Process-2113, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2113, Step 5: Compute individual channel HOG features for the entire image: 0.149666 seconds

Process-2113, Step 6: Misc initializations: 1e-06 seconds

Process-2113, Step 7: for loop: 0.006261 seconds

Process-2116, Step 1: Divide with 255, processing time: 0.010552 seconds

Process-2113All steps time: 0.167038 seconds

Process-2116, Step 2: Resize if scale is not 1: 0.000478 seconds

Process-2116, Step 3: Get HOG channels: 5e-06 seconds

Process-2113, Find cars processing time: 0.167116 seconds

Process-2116, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2116, Step 5: Compute individual channel HOG features for the entire image: 0.152123 seconds

Process-2116, Step 6: Misc initializations: 1e-06 seconds

Process-2116, Step 7: for loop: 0.007488 seconds

Process-2116All steps time: 0.170652 seconds

Process-2116, Find cars processing time: 0.170704 seconds

Process-2116, Step 1: Divide with 255, processing time: 0.006246 seconds

Process-2116, Step 2: Resize if scale is not 1: 0.000876 seconds

Process-2116, Step 3: Get HOG channels: 6e-06 seconds

Process-2116, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2116, Step 5: Compute individual channel HOG features for the entire image: 0.127048 seconds

Process-2116, Step 6: Misc initializations: 1e-06 seconds

Process-2116, Step 7: for loop: 0.006052 seconds

Process-2116All steps time: 0.140235 seconds

Process-2116, Find cars processing time: 0.14029 seconds

Process-2113, Step 1: Divide with 255, processing time: 0.009593 seconds

Process-2113, Step 2: Resize if scale is not 1: 0.001146 seconds

Process-2113, Step 3: Get HOG channels: 7e-06 seconds

Process-2113, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2113, Step 5: Compute individual channel HOG features for the entire image: 0.105103 seconds

Process-2113, Step 6: Misc initializations: 0.0 seconds

Process-2113, Step 7: for loop: 0.006019 seconds

Process-2113All steps time: 0.121875 seconds

Process-2113, Find cars processing time: 0.121938 seconds

Process-2114, Step 1: Divide with 255, processing time: 0.010231 seconds

Process-2114, Step 2: Resize if scale is not 1: 0.001243 seconds

Process-2114, Step 3: Get HOG channels: 8e-06 seconds

Process-2114, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2114, Step 5: Compute individual channel HOG features for the entire image: 0.129333 seconds

Process-2114, Step 6: Misc initializations: 1e-06 seconds

Process-2114, Step 7: for loop: 0.007082 seconds

Process-2114All steps time: 0.147905 seconds

Process-2114, Find cars processing time: 0.14797 seconds

Process-2115, Step 1: Divide with 255, processing time: 0.007543 seconds

Process-2115, Step 2: Resize if scale is not 1: 0.000879 seconds

Process-2115, Step 3: Get HOG channels: 6e-06 seconds

Process-2115, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2115, Step 5: Compute individual channel HOG features for the entire image: 0.131217 seconds

Process-2115, Step 6: Misc initializations: 1e-06 seconds

Process-2115, Step 7: for loop: 0.007106 seconds

Process-2115All steps time: 0.146758 seconds

Process-2115, Find cars processing time: 0.146812 seconds

Process-2116, Step 1: Divide with 255, processing time: 0.010056 seconds

Process-2116, Step 2: Resize if scale is not 1: 0.001801 seconds

Process-2116, Step 3: Get HOG channels: 8e-06 seconds

Process-2116, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2116, Step 5: Compute individual channel HOG features for the entire image: 0.091473 seconds

Process-2116, Step 6: Misc initializations: 0.0 seconds

Process-2116, Step 7: for loop: 0.00507 seconds

Process-2116All steps time: 0.108415 seconds

Process-2116, Find cars processing time: 0.108467 seconds

The times for each task are: [0.28706, 0.345252, 0.311355, 0.212128, 0.243895, 0.258125, 0.22252

Minimum: 0.108467 Maximum: 0.345252 Average: 0.1947 seconds

Number of processes used: 4 window size 260

Length of task list: 22

Number of processes used: 4

Process-2117, Step 1: Divide with 255, processing time: 0.029634 seconds

Process-2117, Step 2: Resize if scale is not 1: 0.002107 seconds

Process-2117, Step 3: Get HOG channels: 1.8e-05 seconds

Process-2117, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2117, Step 5: Compute individual channel HOG features for the entire image: 0.249599 sec

Process-2117, Step 6: Misc initializations: 1e-06 seconds

Process-2117, Step 7: for loop: 0.017187 seconds

Process-2117All steps time: 0.2985509999999996 seconds

Process-2117, Find cars processing time: 0.298722 seconds

Process-2118, Step 1: Divide with 255, processing time: 0.020709 seconds

Process-2118, Step 2: Resize if scale is not 1: 0.001656 seconds

Process-2118, Step 3: Get HOG channels: 1.4e-05 seconds

Process-2118, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2118, Step 5: Compute individual channel HOG features for the entire image: 0.243357 sec

Process-2118, Step 6: Misc initializations: 1e-06 seconds

Process-2118, Step 7: for loop: 0.014744 seconds

Process-2118All steps time: 0.2804869999999993 seconds

Process-2118, Find cars processing time: 0.280626 seconds

Process-2119, Step 1: Divide with 255, processing time: 0.027893 seconds

Process-2119, Step 2: Resize if scale is not 1: 0.001993 seconds

Process-2119, Step 3: Get HOG channels: 1.9e-05 seconds

Process-2119, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2119, Step 5: Compute individual channel HOG features for the entire image: 0.258006 sec

Process-2119, Step 6: Misc initializations: 1e-06 seconds

Process-2119, Step 7: for loop: 0.01793 seconds

Process-2119All steps time: 0.305847 seconds

Process-2119, Find cars processing time: 0.306016 seconds

Process-2117, Step 1: Divide with 255, processing time: 0.010579 seconds
Process-2117, Step 2: Resize if scale is not 1: 0.000768 seconds
Process-2117, Step 3: Get HOG channels: 7e-06 seconds
Process-2117, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2117, Step 5: Compute individual channel HOG features for the entire image: 0.239282 seconds
Process-2117, Step 6: Misc initializations: 1.2e-05 seconds
Process-2117, Step 7: for loop: 0.016114 seconds

Process-2117All steps time: 0.26676900000000003 seconds

Process-2117, Find cars processing time: 0.266861 seconds

Process-2118, Step 1: Divide with 255, processing time: 0.009497 seconds
Process-2118, Step 2: Resize if scale is not 1: 0.00067 seconds
Process-2118, Step 3: Get HOG channels: 6e-06 seconds
Process-2118, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2118, Step 5: Compute individual channel HOG features for the entire image: 0.198559 seconds
Process-2118, Step 6: Misc initializations: 1e-06 seconds
Process-2118, Step 7: for loop: 0.011645 seconds

Process-2118All steps time: 0.220384 seconds

Process-2118, Find cars processing time: 0.220478 seconds

Process-2120, Step 1: Divide with 255, processing time: 0.026178 seconds
Process-2120, Step 2: Resize if scale is not 1: 0.001912 seconds
Process-2120, Step 3: Get HOG channels: 2.4e-05 seconds
Process-2120, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2120, Step 5: Compute individual channel HOG features for the entire image: 0.212593 seconds
Process-2120, Step 6: Misc initializations: 1e-06 seconds
Process-2120, Step 7: for loop: 0.013996 seconds

Process-2120All steps time: 0.254709 seconds

Process-2120, Find cars processing time: 0.254866 seconds

Process-2119, Step 1: Divide with 255, processing time: 0.010692 seconds
Process-2119, Step 2: Resize if scale is not 1: 0.000976 seconds
Process-2119, Step 3: Get HOG channels: 1.5e-05 seconds
Process-2119, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2119, Step 5: Compute individual channel HOG features for the entire image: 0.240079 seconds
Process-2119, Step 6: Misc initializations: 1e-06 seconds
Process-2119, Step 7: for loop: 0.013015 seconds

Process-2119All steps time: 0.2647879999999997 seconds

Process-2119, Find cars processing time: 0.264894 seconds

Process-2117, Step 1: Divide with 255, processing time: 0.01078 seconds
Process-2117, Step 2: Resize if scale is not 1: 0.000652 seconds
Process-2117, Step 3: Get HOG channels: 5e-06 seconds
Process-2117, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2117, Step 5: Compute individual channel HOG features for the entire image: 0.213286 seconds
Process-2117, Step 6: Misc initializations: 1e-06 seconds
Process-2117, Step 7: for loop: 0.012661 seconds

Process-2117All steps time: 0.23739100000000002 seconds

Process-2117, Find cars processing time: 0.237487 seconds

Process-2120, Step 1: Divide with 255, processing time: 0.009486 seconds
Process-2120, Step 2: Resize if scale is not 1: 0.000576 seconds
Process-2120, Step 3: Get HOG channels: 6e-06 seconds
Process-2120, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2120, Step 5: Compute individual channel HOG features for the entire image: 0.177438 seconds
Process-2120, Step 6: Misc initializations: 1e-06 seconds
Process-2120, Step 7: for loop: 0.010145 seconds

Process-2120All steps time: 0.197657 seconds

Process-2120, Find cars processing time: 0.197736 seconds

Process-2118, Step 1: Divide with 255, processing time: 0.010492 seconds
Process-2118, Step 2: Resize if scale is not 1: 0.000791 seconds
Process-2118, Step 3: Get HOG channels: 7e-06 seconds
Process-2118, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2118, Step 5: Compute individual channel HOG features for the entire image: 0.148547 seconds
Process-2118, Step 6: Misc initializations: 1e-06 seconds
Process-2118, Step 7: for loop: 0.008435 seconds

Process-2118All steps time: 0.1682810000000001 seconds

Process-2118, Find cars processing time: 0.168347 seconds

Process-2117, Step 1: Divide with 255, processing time: 0.008626 seconds
Process-2117, Step 2: Resize if scale is not 1: 0.000677 seconds
Process-2117, Step 3: Get HOG channels: 7e-06 seconds
Process-2117, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2117, Step 5: Compute individual channel HOG features for the entire image: 0.135532 seconds
Process-2117, Step 6: Misc initializations: 1e-06 seconds
Process-2117, Step 7: for loop: 0.008121 seconds

Process-2117All steps time: 0.152971 seconds

Process-2117, Find cars processing time: 0.153045 seconds

Process-2120, Step 1: Divide with 255, processing time: 0.010634 seconds
Process-2120, Step 2: Resize if scale is not 1: 0.000835 seconds
Process-2120, Step 3: Get HOG channels: 8e-06 seconds
Process-2120, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2120, Step 5: Compute individual channel HOG features for the entire image: 0.147435 seconds
Process-2120, Step 6: Misc initializations: 1e-06 seconds
Process-2120, Step 7: for loop: 0.009097 seconds

Process-2120All steps time: 0.168017 seconds

Process-2120, Find cars processing time: 0.168104 seconds

Process-2119, Step 1: Divide with 255, processing time: 0.009988 seconds
Process-2119, Step 2: Resize if scale is not 1: 0.000593 seconds
Process-2119, Step 3: Get HOG channels: 6e-06 seconds
Process-2119, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2119, Step 5: Compute individual channel HOG features for the entire image: 0.184291 seconds
Process-2119, Step 6: Misc initializations: 1e-06 seconds
Process-2119, Step 7: for loop: 0.010099 seconds

Process-2119All steps time: 0.204984 seconds

Process-2119, Find cars processing time: 0.205057 seconds

Process-2118, Step 1: Divide with 255, processing time: 0.00962 seconds
Process-2118, Step 2: Resize if scale is not 1: 0.000777 seconds
Process-2118, Step 3: Get HOG channels: 8e-06 seconds
Process-2118, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2118, Step 5: Compute individual channel HOG features for the entire image: 0.185092 seconds
Process-2118, Step 6: Misc initializations: 1e-06 seconds
Process-2118, Step 7: for loop: 0.008944 seconds

Process-2118All steps time: 0.2044510000000002 seconds

Process-2118, Find cars processing time: 0.204545 seconds

Process-2119, Step 1: Divide with 255, processing time: 0.0082 seconds
Process-2119, Step 2: Resize if scale is not 1: 0.000643 seconds
Process-2119, Step 3: Get HOG channels: 8e-06 seconds
Process-2119, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2119, Step 5: Compute individual channel HOG features for the entire image: 0.155197 seconds
Process-2119, Step 6: Misc initializations: 0.0 seconds
Process-2119, Step 7: for loop: 0.00428 seconds

Process-2119All steps time: 0.1683370000000001 seconds

Process-2119, Find cars processing time: 0.168413 seconds

Process-2117, Step 1: Divide with 255, processing time: 0.009699 seconds
Process-2117, Step 2: Resize if scale is not 1: 0.00088 seconds
Process-2117, Step 3: Get HOG channels: 7e-06 seconds
Process-2117, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2120, Step 1: Divide with 255, processing time: 0.010468 seconds
Process-2117, Step 5: Compute individual channel HOG features for the entire image: 0.118149 seconds
Process-2120, Step 2: Resize if scale is not 1: 0.000726 seconds
Process-2117, Step 6: Misc initializations: 1e-06 seconds
Process-2120, Step 3: Get HOG channels: 7e-06 seconds
Process-2117, Step 7: for loop: 0.005872 seconds

Process-2120, Step 4: Define blocks and steps as above: 2e-05 seconds
Process-2117All steps time: 0.1346149999999998 seconds
Process-2120, Step 5: Compute individual channel HOG features for the entire image: 0.159627 seconds

Process-2120, Step 6: Misc initializations: 1e-06 seconds
Process-2117, Find cars processing time: 0.13468 seconds

Process-2120, Step 7: for loop: 0.007255 seconds

Process-2120All steps time: 0.178104 seconds

Process-2120, Find cars processing time: 0.178167 seconds

Process-2119, Step 1: Divide with 255, processing time: 0.008924 seconds
Process-2119, Step 2: Resize if scale is not 1: 0.000998 seconds
Process-2119, Step 3: Get HOG channels: 8e-06 seconds
Process-2119, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2119, Step 5: Compute individual channel HOG features for the entire image: 0.130928 seconds
Process-2119, Step 6: Misc initializations: 2e-06 seconds
Process-2119, Step 7: for loop: 0.010588 seconds

Process-2119All steps time: 0.151455 seconds

Process-2119, Find cars processing time: 0.151536 seconds

Process-2120, Step 1: Divide with 255, processing time: 0.011353 seconds
Process-2120, Step 2: Resize if scale is not 1: 0.000807 seconds
Process-2120, Step 3: Get HOG channels: 5e-06 seconds
Process-2120, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2120, Step 5: Compute individual channel HOG features for the entire image: 0.129343 seconds
Process-2120, Step 6: Misc initializations: 1e-06 seconds
Process-2120, Step 7: for loop: 0.007135 seconds

Process-2120All steps time: 0.148651 seconds

Process-2120, Find cars processing time: 0.148714 seconds

```
Process-2118, Step 1: Divide with 255, processing time: 0.008786 seconds
Process-2118, Step 2: Resize if scale is not 1: 0.000997 seconds
Process-2118, Step 3: Get HOG channels: 8e-06 seconds
Process-2118, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2118, Step 5: Compute individual channel HOG features for the entire image: 0.133192 seconds
Process-2118, Step 6: Misc initializations: 0.0 seconds
Process-2118, Step 7: for loop: 0.006485 seconds

Process-2118All steps time: 0.149474 seconds

Process-2118, Find cars processing time: 0.149539 seconds

Process-2117, Step 1: Divide with 255, processing time: 0.011387 seconds
Process-2117, Step 2: Resize if scale is not 1: 0.001242 seconds
Process-2117, Step 3: Get HOG channels: 8e-06 seconds
Process-2117, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2117, Step 5: Compute individual channel HOG features for the entire image: 0.108896 seconds
Process-2117, Step 6: Misc initializations: 0.0 seconds
Process-2117, Step 7: for loop: 0.022443 seconds

Process-2117All steps time: 0.143984 seconds

Process-2117, Find cars processing time: 0.14408 seconds

Process-2119, Step 1: Divide with 255, processing time: 0.010715 seconds
Process-2119, Step 2: Resize if scale is not 1: 0.00187 seconds
Process-2119, Step 3: Get HOG channels: 6e-06 seconds
Process-2119, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2119, Step 5: Compute individual channel HOG features for the entire image: 0.111608 seconds
Process-2119, Step 6: Misc initializations: 0.0 seconds
Process-2119, Step 7: for loop: 0.006074 seconds

Process-2119All steps time: 0.130279 seconds

Process-2119, Find cars processing time: 0.130377 seconds

The times for each task are: [0.298722, 0.280626, 0.306016, 0.266861, 0.220478, 0.254866, 0.2648
```

Minimum: 0.130377 Maximum: 0.306016 Average: 0.2015 seconds

```
*****
Number of processes used: 4 window size 260
Length of task list: 22
Number of processes used: 4
```

```
Process-2121, Step 1: Divide with 255, processing time: 0.025019 seconds
Process-2121, Step 2: Resize if scale is not 1: 0.001952 seconds
```

Process-2121, Step 3: Get HOG channels: 1.7e-05 seconds
Process-2121, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2121, Step 5: Compute individual channel HOG features for the entire image: 0.243284 seconds
Process-2121, Step 6: Misc initializations: 1e-06 seconds
Process-2121, Step 7: for loop: 0.014979 seconds

Process-2121All steps time: 0.285258 seconds

Process-2121, Find cars processing time: 0.285401 seconds

Process-2122, Step 1: Divide with 255, processing time: 0.029027 seconds
Process-2122, Step 2: Resize if scale is not 1: 0.002173 seconds
Process-2122, Step 3: Get HOG channels: 2.4e-05 seconds
Process-2122, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-2122, Step 5: Compute individual channel HOG features for the entire image: 0.261375 seconds
Process-2122, Step 6: Misc initializations: 1e-06 seconds
Process-2122, Step 7: for loop: 0.017585 seconds

Process-2122All steps time: 0.3101980000000003 seconds

Process-2122, Find cars processing time: 0.310395 seconds

Process-2123, Step 1: Divide with 255, processing time: 0.026781 seconds
Process-2123, Step 2: Resize if scale is not 1: 0.001842 seconds
Process-2123, Step 3: Get HOG channels: 1.9e-05 seconds
Process-2123, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2123, Step 5: Compute individual channel HOG features for the entire image: 0.245517 seconds
Process-2123, Step 6: Misc initializations: 1e-06 seconds
Process-2123, Step 7: for loop: 0.017455 seconds

Process-2123All steps time: 0.2916209999999996 seconds

Process-2123, Find cars processing time: 0.291761 seconds

Process-2122, Step 1: Divide with 255, processing time: 0.009463 seconds
Process-2122, Step 2: Resize if scale is not 1: 0.000818 seconds
Process-2122, Step 3: Get HOG channels: 7e-06 seconds
Process-2122, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2122, Step 5: Compute individual channel HOG features for the entire image: 0.198829 seconds
Process-2122, Step 6: Misc initializations: 1e-06 seconds
Process-2122, Step 7: for loop: 0.012731 seconds

Process-2122All steps time: 0.221856 seconds

Process-2122, Find cars processing time: 0.221936 seconds

Process-2124, Step 1: Divide with 255, processing time: 0.026558 seconds
Process-2124, Step 2: Resize if scale is not 1: 0.00181 seconds

Process-2124, Step 3: Get HOG channels: 1.9e-05 seconds
Process-2124, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2124, Step 5: Compute individual channel HOG features for the entire image: 0.220191 seconds
Process-2124, Step 6: Misc initializations: 1e-06 seconds
Process-2124, Step 7: for loop: 0.013214 seconds

Process-2124All steps time: 0.261798 seconds

Process-2124, Find cars processing time: 0.261937 seconds

Process-2121, Step 1: Divide with 255, processing time: 0.007008 seconds
Process-2121, Step 2: Resize if scale is not 1: 0.000798 seconds
Process-2121, Step 3: Get HOG channels: 7e-06 seconds
Process-2121, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2121, Step 5: Compute individual channel HOG features for the entire image: 0.255965 seconds
Process-2121, Step 6: Misc initializations: 0.0 seconds
Process-2121, Step 7: for loop: 0.01444 seconds

Process-2121All steps time: 0.278224 seconds

Process-2121, Find cars processing time: 0.27829 seconds

Process-2122, Step 1: Divide with 255, processing time: 0.009742 seconds
Process-2122, Step 2: Resize if scale is not 1: 0.000719 seconds
Process-2122, Step 3: Get HOG channels: 5e-06 seconds
Process-2122, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2122, Step 5: Compute individual channel HOG features for the entire image: 0.156888 seconds
Process-2122, Step 6: Misc initializations: 1e-06 seconds
Process-2122, Step 7: for loop: 0.010609 seconds

Process-2122All steps time: 0.17796900000000002 seconds

Process-2122, Find cars processing time: 0.178033 seconds

Process-2123, Step 1: Divide with 255, processing time: 0.010456 seconds
Process-2123, Step 2: Resize if scale is not 1: 0.000818 seconds
Process-2123, Step 3: Get HOG channels: 5e-06 seconds
Process-2123, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2123, Step 5: Compute individual channel HOG features for the entire image: 0.200242 seconds
Process-2123, Step 6: Misc initializations: 0.0 seconds
Process-2123, Step 7: for loop: 0.01023 seconds

Process-2123All steps time: 0.2217559999999998 seconds

Process-2123, Find cars processing time: 0.221827 seconds

Process-2122, Step 1: Divide with 255, processing time: 0.009519 seconds
Process-2122, Step 2: Resize if scale is not 1: 0.000713 seconds

Process-2122, Step 3: Get HOG channels: 7e-06 seconds
Process-2122, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2122, Step 5: Compute individual channel HOG features for the entire image: 0.141238 seconds
Process-2122, Step 6: Misc initializations: 0.0 seconds
Process-2122, Step 7: for loop: 0.008119 seconds

Process-2122All steps time: 0.159602 seconds

Process-2122, Find cars processing time: 0.159666 seconds

Process-2124, Step 1: Divide with 255, processing time: 0.010453 seconds
Process-2124, Step 2: Resize if scale is not 1: 0.000564 seconds
Process-2124, Step 3: Get HOG channels: 5e-06 seconds
Process-2124, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2124, Step 5: Compute individual channel HOG features for the entire image: 0.162168 seconds
Process-2124, Step 6: Misc initializations: 1e-06 seconds
Process-2124, Step 7: for loop: 0.007895 seconds

Process-2124All steps time: 0.181091 seconds

Process-2124, Find cars processing time: 0.181144 seconds

Process-2121, Step 1: Divide with 255, processing time: 0.010476 seconds
Process-2121, Step 2: Resize if scale is not 1: 0.000978 seconds
Process-2121, Step 3: Get HOG channels: 1e-05 seconds
Process-2121, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2121, Step 5: Compute individual channel HOG features for the entire image: 0.160999 seconds
Process-2121, Step 6: Misc initializations: 1e-06 seconds
Process-2121, Step 7: for loop: 0.008232 seconds

Process-2121All steps time: 0.180704 seconds

Process-2121, Find cars processing time: 0.180795 seconds

Process-2124, Step 1: Divide with 255, processing time: 0.010957 seconds
Process-2124, Step 2: Resize if scale is not 1: 0.000863 seconds
Process-2124, Step 3: Get HOG channels: 8e-06 seconds
Process-2122, Step 1: Divide with 255, processing time: 0.010932 seconds
Process-2124, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2122, Step 2: Resize if scale is not 1: 0.000918 seconds
Process-2124, Step 5: Compute individual channel HOG features for the entire image: 0.120858 seconds
Process-2122, Step 3: Get HOG channels: 8e-06 seconds
Process-2124, Step 6: Misc initializations: 1e-06 seconds
Process-2122, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2124, Step 7: for loop: 0.006036 seconds
Process-2122, Step 5: Compute individual channel HOG features for the entire image: 0.133626 seconds

Process-2122, Step 6: Misc initializations: 1e-06 seconds

Process-2124All steps time: 0.13873000000000002 seconds
Process-2122, Step 7: for loop: 0.006393 seconds

Process-2124, Find cars processing time: 0.138789 seconds
Process-2122All steps time: 0.151886 seconds

Process-2122, Find cars processing time: 0.151958 seconds

Process-2123, Step 1: Divide with 255, processing time: 0.010537 seconds
Process-2123, Step 2: Resize if scale is not 1: 0.000871 seconds
Process-2123, Step 3: Get HOG channels: 8e-06 seconds
Process-2123, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2123, Step 5: Compute individual channel HOG features for the entire image: 0.154374 seconds
Process-2123, Step 6: Misc initializations: 1e-06 seconds
Process-2123, Step 7: for loop: 0.007815 seconds

Process-2123All steps time: 0.173613 seconds

Process-2123, Find cars processing time: 0.173678 seconds

Process-2124, Step 1: Divide with 255, processing time: 0.01066 seconds
Process-2124, Step 2: Resize if scale is not 1: 0.000863 seconds
Process-2124, Step 3: Get HOG channels: 8e-06 seconds
Process-2124, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2124, Step 5: Compute individual channel HOG features for the entire image: 0.141707 seconds
Process-2124, Step 6: Misc initializations: 1e-06 seconds
Process-2124, Step 7: for loop: 0.007157 seconds

Process-2124All steps time: 0.160405 seconds

Process-2124, Find cars processing time: 0.160478 seconds

Process-2123, Step 1: Divide with 255, processing time: 0.010849 seconds
Process-2123, Step 2: Resize if scale is not 1: 0.001346 seconds
Process-2123, Step 3: Get HOG channels: 1e-05 seconds
Process-2123, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2123, Step 5: Compute individual channel HOG features for the entire image: 0.117856 seconds
Process-2123, Step 6: Misc initializations: 1e-06 seconds
Process-2123, Step 7: for loop: 0.006343 seconds

Process-2123All steps time: 0.1364139999999998 seconds

Process-2123, Find cars processing time: 0.136496 seconds

Process-2124, Step 1: Divide with 255, processing time: 0.024093 seconds
Process-2124, Step 2: Resize if scale is not 1: 0.000769 seconds

Process-2124, Step 3: Get HOG channels: 6e-06 seconds
Process-2124, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2124, Step 5: Compute individual channel HOG features for the entire image: 0.112964 seconds
Process-2124, Step 6: Misc initializations: 1e-06 seconds
Process-2124, Step 7: for loop: 0.006583 seconds

Process-2124All steps time: 0.144421 seconds

Process-2124, Find cars processing time: 0.144503 seconds

Process-2122, Step 1: Divide with 255, processing time: 0.010765 seconds
Process-2122, Step 2: Resize if scale is not 1: 0.000613 seconds
Process-2122, Step 3: Get HOG channels: 5e-06 seconds
Process-2122, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2122, Step 5: Compute individual channel HOG features for the entire image: 0.09929 seconds
Process-2122, Step 6: Misc initializations: 2e-06 seconds
Process-2122, Step 7: for loop: 0.004228 seconds

Process-2122All steps time: 0.114908 seconds

Process-2122, Find cars processing time: 0.114969 seconds

Process-2121, Step 1: Divide with 255, processing time: 0.011719 seconds
Process-2121, Step 2: Resize if scale is not 1: 0.000837 seconds
Process-2121, Step 3: Get HOG channels: 9e-06 seconds
Process-2121, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2121, Step 5: Compute individual channel HOG features for the entire image: 0.145614 seconds
Process-2121, Step 6: Misc initializations: 0.0 seconds
Process-2121, Step 7: for loop: 0.005783 seconds

Process-2121All steps time: 0.163972 seconds

Process-2121, Find cars processing time: 0.164072 seconds

Process-2124, Step 1: Divide with 255, processing time: 0.010348 seconds
Process-2124, Step 2: Resize if scale is not 1: 0.000771 seconds
Process-2124, Step 3: Get HOG channels: 9e-06 seconds
Process-2124, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2124, Step 5: Compute individual channel HOG features for the entire image: 0.133422 seconds
Process-2124, Step 6: Misc initializations: 1e-06 seconds
Process-2124, Step 7: for loop: 0.007189 seconds

Process-2124All steps time: 0.1517500000000002 seconds

Process-2124, Find cars processing time: 0.151825 seconds

Process-2123, Step 1: Divide with 255, processing time: 0.009998 seconds
Process-2123, Step 2: Resize if scale is not 1: 0.000538 seconds

Process-2123, Step 3: Get HOG channels: 6e-06 seconds
Process-2123, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2123, Step 5: Compute individual channel HOG features for the entire image: 0.103291 seconds
Process-2123, Step 6: Misc initializations: 1e-06 seconds
Process-2123, Step 7: for loop: 0.006241 seconds

Process-2123All steps time: 0.120082 seconds

Process-2123, Find cars processing time: 0.120143 seconds

Process-2122, Step 1: Divide with 255, processing time: 0.010557 seconds
Process-2122, Step 2: Resize if scale is not 1: 0.000975 seconds
Process-2122, Step 3: Get HOG channels: 2.1e-05 seconds
Process-2122, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2122, Step 5: Compute individual channel HOG features for the entire image: 0.088485 seconds
Process-2122, Step 6: Misc initializations: 1e-06 seconds
Process-2122, Step 7: for loop: 0.00476 seconds

Process-2122All steps time: 0.104804 seconds

Process-2122, Find cars processing time: 0.104849 seconds

The times for each task are: [0.285401, 0.310395, 0.291761, 0.221936, 0.261937, 0.27829, 0.17803

Minimum: 0.104849 Maximum: 0.310395 Average: 0.1879 seconds

Number of processes used: 4 window size 260
Length of task list: 22
Number of processes used: 4

Process-2125, Step 1: Divide with 255, processing time: 0.030707 seconds
Process-2125, Step 2: Resize if scale is not 1: 0.002269 seconds
Process-2125, Step 3: Get HOG channels: 2.6e-05 seconds
Process-2125, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2125, Step 5: Compute individual channel HOG features for the entire image: 0.249847 seconds
Process-2125, Step 6: Misc initializations: 1e-06 seconds
Process-2125, Step 7: for loop: 0.0177 seconds

Process-2125All steps time: 0.300556 seconds

Process-2125, Find cars processing time: 0.300744 seconds

Process-2126, Step 1: Divide with 255, processing time: 0.02797 seconds
Process-2126, Step 2: Resize if scale is not 1: 0.002116 seconds
Process-2126, Step 3: Get HOG channels: 2.9e-05 seconds
Process-2126, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2126, Step 5: Compute individual channel HOG features for the entire image: 0.273278 seconds

Process-2126, Step 6: Misc initializations: 1e-06 seconds
Process-2126, Step 7: for loop: 0.017766 seconds

Process-2126All steps time: 0.321167 seconds

Process-2126, Find cars processing time: 0.321331 seconds

Process-2127, Step 1: Divide with 255, processing time: 0.029902 seconds
Process-2127, Step 2: Resize if scale is not 1: 0.002109 seconds
Process-2127, Step 3: Get HOG channels: 2.5e-05 seconds
Process-2127, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2127, Step 5: Compute individual channel HOG features for the entire image: 0.244679 seconds
Process-2127, Step 6: Misc initializations: 1e-06 seconds
Process-2127, Step 7: for loop: 0.016867 seconds

Process-2127All steps time: 0.293588 seconds

Process-2127, Find cars processing time: 0.293757 seconds

Process-2128, Step 1: Divide with 255, processing time: 0.031715 seconds
Process-2128, Step 2: Resize if scale is not 1: 0.002298 seconds
Process-2128, Step 3: Get HOG channels: 2.3e-05 seconds
Process-2128, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2128, Step 5: Compute individual channel HOG features for the entire image: 0.205685 seconds
Process-2128, Step 6: Misc initializations: 1e-06 seconds
Process-2128, Step 7: for loop: 0.013939 seconds

Process-2128All steps time: 0.25366700000000003 seconds

Process-2128, Find cars processing time: 0.253835 seconds

Process-2126, Step 1: Divide with 255, processing time: 0.009359 seconds
Process-2126, Step 2: Resize if scale is not 1: 0.000513 seconds
Process-2126, Step 3: Get HOG channels: 5e-06 seconds
Process-2126, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2126, Step 5: Compute individual channel HOG features for the entire image: 0.174549 seconds
Process-2126, Step 6: Misc initializations: 1e-06 seconds
Process-2126, Step 7: for loop: 0.011119 seconds

Process-2126All steps time: 0.195551 seconds

Process-2126, Find cars processing time: 0.195609 seconds

Process-2127, Step 1: Divide with 255, processing time: 0.006246 seconds
Process-2127, Step 2: Resize if scale is not 1: 0.000556 seconds
Process-2127, Step 3: Get HOG channels: 6e-06 seconds
Process-2127, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2127, Step 5: Compute individual channel HOG features for the entire image: 0.237036 seconds

Process-2127, Step 6: Misc initializations: 0.0 seconds
Process-2127, Step 7: for loop: 0.012345 seconds

Process-2127All steps time: 0.256194 seconds

Process-2127, Find cars processing time: 0.25626 seconds

Process-2125, Step 1: Divide with 255, processing time: 0.006489 seconds
Process-2125, Step 2: Resize if scale is not 1: 0.000748 seconds
Process-2125, Step 3: Get HOG channels: 8e-06 seconds
Process-2125, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2125, Step 5: Compute individual channel HOG features for the entire image: 0.241278 seconds
Process-2125, Step 6: Misc initializations: 1e-06 seconds
Process-2125, Step 7: for loop: 0.01674 seconds

Process-2125All steps time: 0.265271 seconds

Process-2125, Find cars processing time: 0.265341 seconds

Process-2126, Step 1: Divide with 255, processing time: 0.010306 seconds
Process-2126, Step 2: Resize if scale is not 1: 0.00092 seconds
Process-2126, Step 3: Get HOG channels: 8e-06 seconds
Process-2126, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2126, Step 5: Compute individual channel HOG features for the entire image: 0.146937 seconds
Process-2126, Step 6: Misc initializations: 0.0 seconds
Process-2126, Step 7: for loop: 0.008479 seconds

Process-2126All steps time: 0.16665800000000003 seconds

Process-2126, Find cars processing time: 0.16673 seconds

Process-2128, Step 1: Divide with 255, processing time: 0.010355 seconds
Process-2128, Step 2: Resize if scale is not 1: 0.000799 seconds
Process-2128, Step 3: Get HOG channels: 6e-06 seconds
Process-2128, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2128, Step 5: Compute individual channel HOG features for the entire image: 0.213552 seconds
Process-2128, Step 6: Misc initializations: 1e-06 seconds
Process-2128, Step 7: for loop: 0.01297 seconds

Process-2128All steps time: 0.237689 seconds

Process-2128, Find cars processing time: 0.237762 seconds

Process-2127, Step 1: Divide with 255, processing time: 0.009427 seconds
Process-2127, Step 2: Resize if scale is not 1: 0.000713 seconds
Process-2127, Step 3: Get HOG channels: 7e-06 seconds
Process-2127, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2127, Step 5: Compute individual channel HOG features for the entire image: 0.170233 seconds

Process-2127, Step 6: Misc initializations: 0.0 seconds
Process-2127, Step 7: for loop: 0.008286 seconds

Process-2127All steps time: 0.1886719999999998 seconds

Process-2127, Find cars processing time: 0.18873 seconds

Process-2125, Step 1: Divide with 255, processing time: 0.00724 seconds
Process-2125, Step 2: Resize if scale is not 1: 0.000738 seconds
Process-2125, Step 3: Get HOG channels: 6e-06 seconds
Process-2125, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2125, Step 5: Compute individual channel HOG features for the entire image: 0.178441 seconds
Process-2125, Step 6: Misc initializations: 2e-06 seconds
Process-2125, Step 7: for loop: 0.01057 seconds

Process-2125All steps time: 0.1970039999999998 seconds

Process-2125, Find cars processing time: 0.197066 seconds

Process-2127, Step 1: Divide with 255, processing time: 0.009506 seconds
Process-2127, Step 2: Resize if scale is not 1: 0.000766 seconds
Process-2127, Step 3: Get HOG channels: 7e-06 seconds
Process-2127, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2127, Step 5: Compute individual channel HOG features for the entire image: 0.144708 seconds
Process-2127, Step 6: Misc initializations: 1e-06 seconds
Process-2127, Step 7: for loop: 0.008398 seconds

Process-2127All steps time: 0.1633929999999998 seconds

Process-2127, Find cars processing time: 0.163464 seconds

Process-2126, Step 1: Divide with 255, processing time: 0.009772 seconds
Process-2126, Step 2: Resize if scale is not 1: 0.000806 seconds
Process-2126, Step 3: Get HOG channels: 7e-06 seconds
Process-2126, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2126, Step 5: Compute individual channel HOG features for the entire image: 0.172586 seconds
Process-2126, Step 6: Misc initializations: 1e-06 seconds
Process-2126, Step 7: for loop: 0.011439 seconds

Process-2126All steps time: 0.194619 seconds

Process-2126, Find cars processing time: 0.194696 seconds

Process-2128, Step 1: Divide with 255, processing time: 0.008654 seconds
Process-2128, Step 2: Resize if scale is not 1: 0.000977 seconds
Process-2128, Step 3: Get HOG channels: 9e-06 seconds
Process-2128, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2128, Step 5: Compute individual channel HOG features for the entire image: 0.15528 seconds

Process-2128, Step 6: Misc initializations: 1e-06 seconds
Process-2128, Step 7: for loop: 0.010674 seconds

Process-2128All steps time: 0.1756039999999998 seconds

Process-2128, Find cars processing time: 0.175698 seconds

Process-2126, Step 1: Divide with 255, processing time: 0.018314 seconds
Process-2126, Step 2: Resize if scale is not 1: 0.000406 seconds
Process-2126, Step 3: Get HOG channels: 5e-06 seconds
Process-2126, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2126, Step 5: Compute individual channel HOG features for the entire image: 0.11343 seconds
Process-2126, Step 6: Misc initializations: 1e-06 seconds
Process-2126, Step 7: for loop: 0.005161 seconds

Process-2126All steps time: 0.137321 seconds

Process-2126, Find cars processing time: 0.137382 seconds

Process-2125, Step 1: Divide with 255, processing time: 0.02458 seconds

Process-2125, Step 2: Resize if scale is not 1: 0.000573 seconds
Process-2125, Step 3: Get HOG channels: 5e-06 seconds
Process-2125, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2125, Step 5: Compute individual channel HOG features for the entire image: 0.126136 seconds
Process-2125, Step 6: Misc initializations: 1e-06 seconds
Process-2125, Step 7: for loop: 0.008253 seconds

Process-2125All steps time: 0.159554 seconds

Process-2125, Find cars processing time: 0.159641 seconds

Process-2128, Step 1: Divide with 255, processing time: 0.010028 seconds
Process-2128, Step 2: Resize if scale is not 1: 0.000853 seconds
Process-2128, Step 3: Get HOG channels: 2e-05 seconds
Process-2128, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2128, Step 5: Compute individual channel HOG features for the entire image: 0.119383 seconds
Process-2128, Step 6: Misc initializations: 1e-06 seconds
Process-2128, Step 7: for loop: 0.006367 seconds

Process-2128All steps time: 0.136657 seconds

Process-2128, Find cars processing time: 0.136714 seconds

Process-2127, Step 1: Divide with 255, processing time: 0.015752 seconds
Process-2127, Step 2: Resize if scale is not 1: 0.000667 seconds
Process-2127, Step 3: Get HOG channels: 5e-06 seconds
Process-2127, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2127, Step 5: Compute individual channel HOG features for the entire image: 0.165049 seconds

Process-2127, Step 6: Misc initializations: 1e-06 seconds
Process-2127, Step 7: for loop: 0.007439 seconds

Process-2127All steps time: 0.188918 seconds

Process-2127, Find cars processing time: 0.188981 seconds

Process-2126, Step 1: Divide with 255, processing time: 0.009323 seconds
Process-2126, Step 2: Resize if scale is not 1: 0.00123 seconds
Process-2126, Step 3: Get HOG channels: 6e-06 seconds
Process-2126, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2126, Step 5: Compute individual channel HOG features for the entire image: 0.146043 seconds
Process-2126, Step 6: Misc initializations: 1e-06 seconds
Process-2126, Step 7: for loop: 0.007133 seconds

Process-2126All steps time: 0.163741 seconds

Process-2126, Find cars processing time: 0.163804 seconds

Process-2125, Step 1: Divide with 255, processing time: 0.023142 seconds
Process-2125, Step 2: Resize if scale is not 1: 0.000838 seconds
Process-2125, Step 3: Get HOG channels: 6e-06 seconds
Process-2125, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2125, Step 5: Compute individual channel HOG features for the entire image: 0.130574 seconds
Process-2125, Step 6: Misc initializations: 1e-06 seconds
Process-2125, Step 7: for loop: 0.007475 seconds

Process-2125All steps time: 0.1620410000000002 seconds

Process-2125, Find cars processing time: 0.162103 seconds

Process-2127, Step 1: Divide with 255, processing time: 0.010189 seconds
Process-2127, Step 2: Resize if scale is not 1: 0.001166 seconds
Process-2127, Step 3: Get HOG channels: 5e-06 seconds
Process-2127, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2127, Step 5: Compute individual channel HOG features for the entire image: 0.122457 seconds
Process-2127, Step 6: Misc initializations: 1e-06 seconds
Process-2127, Step 7: for loop: 0.006413 seconds

Process-2127All steps time: 0.140236 seconds

Process-2127, Find cars processing time: 0.140299 seconds

Process-2128, Step 1: Divide with 255, processing time: 0.010619 seconds
Process-2128, Step 2: Resize if scale is not 1: 0.001093 seconds
Process-2128, Step 3: Get HOG channels: 1.1e-05 seconds
Process-2128, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2128, Step 5: Compute individual channel HOG features for the entire image: 0.143745 seconds

Process-2128, Step 6: Misc initializations: 1e-06 seconds
Process-2128, Step 7: for loop: 0.009734 seconds

Process-2128All steps time: 0.165213 seconds

Process-2128, Find cars processing time: 0.165313 seconds

The times for each task are: [0.300744, 0.321331, 0.293757, 0.253835, 0.195609, 0.25626, 0.26534

Minimum: 0.136714 Maximum: 0.321331 Average: 0.203 seconds

Number of processes used: 4 window size 260

Length of task list: 22

Number of processes used: 4

Process-2129, Step 1: Divide with 255, processing time: 0.033347 seconds

Process-2129, Step 2: Resize if scale is not 1: 0.002442 seconds

Process-2129, Step 3: Get HOG channels: 2.6e-05 seconds

Process-2129, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2129, Step 5: Compute individual channel HOG features for the entire image: 0.252029 seconds

Process-2129, Step 6: Misc initializations: 1e-06 seconds

Process-2129, Step 7: for loop: 0.017165 seconds

Process-2129All steps time: 0.305017 seconds

Process-2129, Find cars processing time: 0.305207 seconds

Process-2130, Step 1: Divide with 255, processing time: 0.030103 seconds

Process-2130, Step 2: Resize if scale is not 1: 0.002261 seconds

Process-2130, Step 3: Get HOG channels: 2.1e-05 seconds

Process-2130, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2130, Step 5: Compute individual channel HOG features for the entire image: 0.246308 seconds

Process-2130, Step 6: Misc initializations: 1e-06 seconds

Process-2130, Step 7: for loop: 0.018427 seconds

Process-2130All steps time: 0.2971270000000003 seconds

Process-2130, Find cars processing time: 0.297294 seconds

Process-2131, Step 1: Divide with 255, processing time: 0.03195 seconds

Process-2131, Step 2: Resize if scale is not 1: 0.002155 seconds

Process-2131, Step 3: Get HOG channels: 2.9e-05 seconds

Process-2131, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2131, Step 5: Compute individual channel HOG features for the entire image: 0.208386 seconds

Process-2131, Step 6: Misc initializations: 0.0 seconds

Process-2131, Step 7: for loop: 0.015287 seconds

Process-2131All steps time: 0.257813 seconds

Process-2131, Find cars processing time: 0.257982 seconds

Process-2129, Step 1: Divide with 255, processing time: 0.010823 seconds

Process-2129, Step 2: Resize if scale is not 1: 0.00053 seconds

Process-2129, Step 3: Get HOG channels: 5e-06 seconds

Process-2129, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2129, Step 5: Compute individual channel HOG features for the entire image: 0.238521 seconds

Process-2129, Step 6: Misc initializations: 1e-06 seconds

Process-2129, Step 7: for loop: 0.016037 seconds

Process-2129All steps time: 0.265922 seconds

Process-2129, Find cars processing time: 0.265978 seconds

Process-2132, Step 1: Divide with 255, processing time: 0.032393 seconds

Process-2132, Step 2: Resize if scale is not 1: 0.002151 seconds

Process-2132, Step 3: Get HOG channels: 2.2e-05 seconds

Process-2132, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2132, Step 5: Compute individual channel HOG features for the entire image: 0.20132 seconds

Process-2132, Step 6: Misc initializations: 1e-06 seconds

Process-2132, Step 7: for loop: 0.013435 seconds

Process-2132All steps time: 0.249327 seconds

Process-2132, Find cars processing time: 0.2495 seconds

Process-2130, Step 1: Divide with 255, processing time: 0.009957 seconds

Process-2130, Step 2: Resize if scale is not 1: 0.000813 seconds

Process-2130, Step 3: Get HOG channels: 7e-06 seconds

Process-2130, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2130, Step 5: Compute individual channel HOG features for the entire image: 0.201821 seconds

Process-2130, Step 6: Misc initializations: 1e-06 seconds

Process-2130, Step 7: for loop: 0.012687 seconds

Process-2130All steps time: 0.225293 seconds

Process-2130, Find cars processing time: 0.225372 seconds

Process-2131, Step 1: Divide with 255, processing time: 0.010262 seconds

Process-2131, Step 2: Resize if scale is not 1: 0.00087 seconds

Process-2131, Step 3: Get HOG channels: 7e-06 seconds

Process-2131, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2131, Step 5: Compute individual channel HOG features for the entire image: 0.164524 seconds

Process-2131, Step 6: Misc initializations: 1e-06 seconds

Process-2131, Step 7: for loop: 0.01098 seconds

Process-2131All steps time: 0.18665099999999998 seconds

Process-2131, Find cars processing time: 0.186728 seconds

Process-2129, Step 1: Divide with 255, processing time: 0.010355 seconds

Process-2129, Step 2: Resize if scale is not 1: 0.000831 seconds

Process-2129, Step 3: Get HOG channels: 5e-06 seconds

Process-2129, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2129, Step 5: Compute individual channel HOG features for the entire image: 0.200009 seconds

Process-2129, Step 6: Misc initializations: 1e-06 seconds

Process-2129, Step 7: for loop: 0.01232 seconds

Process-2129All steps time: 0.223526 seconds

Process-2129, Find cars processing time: 0.223601 seconds

Process-2132, Step 1: Divide with 255, processing time: 0.009764 seconds

Process-2132, Step 2: Resize if scale is not 1: 0.000804 seconds

Process-2132, Step 3: Get HOG channels: 7e-06 seconds

Process-2132, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2132, Step 5: Compute individual channel HOG features for the entire image: 0.171167 seconds

Process-2132, Step 6: Misc initializations: 1e-06 seconds

Process-2132, Step 7: for loop: 0.009445 seconds

Process-2132All steps time: 0.1911950000000003 seconds

Process-2132, Find cars processing time: 0.191263 seconds

Process-2130, Step 1: Divide with 255, processing time: 0.008761 seconds

Process-2130, Step 2: Resize if scale is not 1: 0.000874 seconds

Process-2130, Step 3: Get HOG channels: 8e-06 seconds

Process-2130, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2130, Step 5: Compute individual channel HOG features for the entire image: 0.170155 seconds

Process-2130, Step 6: Misc initializations: 1e-06 seconds

Process-2130, Step 7: for loop: 0.009901 seconds

Process-2130All steps time: 0.189708 seconds

Process-2130, Find cars processing time: 0.189791 seconds

Process-2131, Step 1: Divide with 255, processing time: 0.010157 seconds

Process-2131, Step 2: Resize if scale is not 1: 0.000853 seconds

Process-2131, Step 3: Get HOG channels: 8e-06 seconds

Process-2131, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2131, Step 5: Compute individual channel HOG features for the entire image: 0.145611 seconds

Process-2131, Step 6: Misc initializations: 0.0 seconds

Process-2131, Step 7: for loop: 0.007858 seconds

Process-2131All steps time: 0.164494 seconds

Process-2131, Find cars processing time: 0.164557 seconds

Process-2129, Step 1: Divide with 255, processing time: 0.009794 seconds

Process-2129, Step 2: Resize if scale is not 1: 0.000687 seconds

Process-2129, Step 3: Get HOG channels: 6e-06 seconds

Process-2129, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2129, Step 5: Compute individual channel HOG features for the entire image: 0.173826 seconds

Process-2129, Step 6: Misc initializations: 1e-06 seconds

Process-2129, Step 7: for loop: 0.010161 seconds

Process-2129All steps time: 0.1944810000000001 seconds

Process-2129, Find cars processing time: 0.194541 seconds

Process-2132, Step 1: Divide with 255, processing time: 0.00928 seconds

Process-2132, Step 2: Resize if scale is not 1: 0.000518 seconds

Process-2132, Step 3: Get HOG channels: 5e-06 seconds

Process-2132, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2132, Step 5: Compute individual channel HOG features for the entire image: 0.14187 seconds

Process-2132, Step 6: Misc initializations: 1e-06 seconds

Process-2132, Step 7: for loop: 0.007071 seconds

Process-2132All steps time: 0.15875 seconds

Process-2132, Find cars processing time: 0.158803 seconds

Process-2130, Step 1: Divide with 255, processing time: 0.009241 seconds

Process-2130, Step 2: Resize if scale is not 1: 0.000515 seconds

Process-2130, Step 3: Get HOG channels: 5e-06 seconds

Process-2130, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2130, Step 5: Compute individual channel HOG features for the entire image: 0.151659 seconds

Process-2130, Step 6: Misc initializations: 1e-06 seconds

Process-2130, Step 7: for loop: 0.007244 seconds

Process-2130All steps time: 0.16867 seconds

Process-2130, Find cars processing time: 0.168741 seconds

Process-2129, Step 1: Divide with 255, processing time: 0.008497 seconds

Process-2129, Step 2: Resize if scale is not 1: 0.000527 seconds

Process-2129, Step 3: Get HOG channels: 5e-06 seconds

Process-2129, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2129, Step 5: Compute individual channel HOG features for the entire image: 0.151816 seconds

Process-2129, Step 6: Misc initializations: 1e-06 seconds

Process-2129, Step 7: for loop: 0.006581 seconds

Process-2129All steps time: 0.167432 seconds

Process-2129, Find cars processing time: 0.167484 seconds

Process-2131, Step 1: Divide with 255, processing time: 0.007769 seconds

Process-2131, Step 2: Resize if scale is not 1: 0.000645 seconds

Process-2131, Step 3: Get HOG channels: 6e-06 seconds

Process-2131, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2131, Step 5: Compute individual channel HOG features for the entire image: 0.152973 seconds

Process-2131, Step 6: Misc initializations: 1e-06 seconds

Process-2131, Step 7: for loop: 0.007542 seconds

Process-2131All steps time: 0.1689419999999998 seconds

Process-2131, Find cars processing time: 0.16901 seconds

Process-2129, Step 1: Divide with 255, processing time: 0.009287 seconds

Process-2129, Step 2: Resize if scale is not 1: 0.000863 seconds

Process-2129, Step 3: Get HOG channels: 6e-06 seconds

Process-2129, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2129, Step 5: Compute individual channel HOG features for the entire image: 0.127747 seconds

Process-2129, Step 6: Misc initializations: 0.0 seconds

Process-2129, Step 7: for loop: 0.007334 seconds

Process-2129All steps time: 0.145243 seconds

Process-2129, Find cars processing time: 0.145304 seconds

Process-2130, Step 1: Divide with 255, processing time: 0.008515 seconds

Process-2130, Step 2: Resize if scale is not 1: 0.000854 seconds

Process-2130, Step 3: Get HOG channels: 6e-06 seconds

Process-2130, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2130, Step 5: Compute individual channel HOG features for the entire image: 0.128666 seconds

Process-2130, Step 6: Misc initializations: 1e-06 seconds

Process-2130, Step 7: for loop: 0.007406 seconds

Process-2130All steps time: 0.145454 seconds

Process-2130, Find cars processing time: 0.145519 seconds

Process-2132, Step 1: Divide with 255, processing time: 0.0096 seconds

Process-2132, Step 2: Resize if scale is not 1: 0.00155 seconds

Process-2132, Step 3: Get HOG channels: 7e-06 seconds

Process-2132, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2132, Step 5: Compute individual channel HOG features for the entire image: 0.123392 seconds

Process-2132, Step 6: Misc initializations: 1e-06 seconds

Process-2132, Step 7: for loop: 0.00503 seconds

Process-2132All steps time: 0.13958700000000002 seconds

Process-2132, Find cars processing time: 0.139648 seconds

Process-2131, Step 1: Divide with 255, processing time: 0.010676 seconds

Process-2131, Step 2: Resize if scale is not 1: 0.001158 seconds

Process-2131, Step 3: Get HOG channels: 8e-06 seconds

Process-2131, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2131, Step 5: Compute individual channel HOG features for the entire image: 0.127085 seconds

Process-2131, Step 6: Misc initializations: 1e-06 seconds

Process-2131, Step 7: for loop: 0.00716 seconds

Process-2131All steps time: 0.146096 seconds

Process-2131, Find cars processing time: 0.146156 seconds

Process-2129, Step 1: Divide with 255, processing time: 0.010959 seconds

Process-2129, Step 2: Resize if scale is not 1: 0.000796 seconds

Process-2129, Step 3: Get HOG channels: 8e-06 seconds

Process-2129, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2129, Step 5: Compute individual channel HOG features for the entire image: 0.128887 seconds

Process-2129, Step 6: Misc initializations: 1e-06 seconds

Process-2129, Step 7: for loop: 0.007463 seconds

Process-2129All steps time: 0.148122 seconds

Process-2129, Find cars processing time: 0.148211 seconds

Process-2130, Step 1: Divide with 255, processing time: 0.010391 seconds

Process-2130, Step 2: Resize if scale is not 1: 0.002032 seconds

Process-2130, Step 3: Get HOG channels: 8e-06 seconds

Process-2130, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2130, Step 5: Compute individual channel HOG features for the entire image: 0.11973 seconds

Process-2130, Step 6: Misc initializations: 1e-06 seconds

Process-2130, Step 7: for loop: 0.006142 seconds

Process-2130All steps time: 0.13831100000000002 seconds

Process-2130, Find cars processing time: 0.138378 seconds

The times for each task are: [0.305207, 0.297294, 0.257982, 0.265978, 0.2495, 0.225372, 0.186728]

Minimum: 0.138378 Maximum: 0.305207 Average: 0.1945 seconds

Number of processes used: 4 window size 260

Length of task list: 22

Number of processes used: 4

Process-2133, Step 1: Divide with 255, processing time: 0.032623 seconds
Process-2133, Step 2: Resize if scale is not 1: 0.002624 seconds
Process-2133, Step 3: Get HOG channels: 2.3e-05 seconds
Process-2133, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2133, Step 5: Compute individual channel HOG features for the entire image: 0.239805 seconds
Process-2133, Step 6: Misc initializations: 1e-06 seconds
Process-2133, Step 7: for loop: 0.015569 seconds

Process-2133All steps time: 0.290651 seconds

Process-2133, Find cars processing time: 0.290822 seconds

Process-2134, Step 1: Divide with 255, processing time: 0.032194 seconds
Process-2134, Step 2: Resize if scale is not 1: 0.002141 seconds
Process-2134, Step 3: Get HOG channels: 1.9e-05 seconds
Process-2134, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2134, Step 5: Compute individual channel HOG features for the entire image: 0.202992 seconds
Process-2134, Step 6: Misc initializations: 1e-06 seconds
Process-2134, Step 7: for loop: 0.014318 seconds

Process-2134All steps time: 0.25167100000000003 seconds

Process-2134, Find cars processing time: 0.251843 seconds

Process-2135, Step 1: Divide with 255, processing time: 0.03452 seconds
Process-2135, Step 2: Resize if scale is not 1: 0.002386 seconds
Process-2135, Step 3: Get HOG channels: 2.4e-05 seconds
Process-2135, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2135, Step 5: Compute individual channel HOG features for the entire image: 0.239737 seconds
Process-2135, Step 6: Misc initializations: 1e-06 seconds
Process-2135, Step 7: for loop: 0.014676 seconds

Process-2135All steps time: 0.29135 seconds

Process-2135, Find cars processing time: 0.29155 seconds

Process-2134, Step 1: Divide with 255, processing time: 0.010166 seconds
Process-2134, Step 2: Resize if scale is not 1: 0.000874 seconds
Process-2134, Step 3: Get HOG channels: 9e-06 seconds
Process-2134, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2134, Step 5: Compute individual channel HOG features for the entire image: 0.190967 seconds
Process-2134, Step 6: Misc initializations: 1e-06 seconds
Process-2134, Step 7: for loop: 0.01111 seconds

Process-2134All steps time: 0.21313500000000002 seconds

Process-2134, Find cars processing time: 0.213208 seconds

Process-2136, Step 1: Divide with 255, processing time: 0.029346 seconds
Process-2136, Step 2: Resize if scale is not 1: 0.002119 seconds
Process-2136, Step 3: Get HOG channels: 1.9e-05 seconds
Process-2136, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2136, Step 5: Compute individual channel HOG features for the entire image: 0.166329 seconds
Process-2136, Step 6: Misc initializations: 0.0 seconds
Process-2136, Step 7: for loop: 0.011088 seconds

Process-2136All steps time: 0.208905 seconds

Process-2136, Find cars processing time: 0.209049 seconds

Process-2133, Step 1: Divide with 255, processing time: 0.010667 seconds
Process-2133, Step 2: Resize if scale is not 1: 0.000947 seconds
Process-2133, Step 3: Get HOG channels: 8e-06 seconds
Process-2133, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2133, Step 5: Compute individual channel HOG features for the entire image: 0.234516 seconds
Process-2133, Step 6: Misc initializations: 1e-06 seconds
Process-2133, Step 7: for loop: 0.01573 seconds

Process-2133All steps time: 0.261876 seconds

Process-2133, Find cars processing time: 0.261961 seconds

Process-2134, Step 1: Divide with 255, processing time: 0.010363 seconds
Process-2134, Step 2: Resize if scale is not 1: 0.000901 seconds
Process-2134, Step 3: Get HOG channels: 1e-05 seconds
Process-2134, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2134, Step 5: Compute individual channel HOG features for the entire image: 0.181135 seconds
Process-2134, Step 6: Misc initializations: 0.0 seconds
Process-2134, Step 7: for loop: 0.01093 seconds

Process-2134All steps time: 0.2033479999999997 seconds

Process-2134, Find cars processing time: 0.20343 seconds

Process-2135, Step 1: Divide with 255, processing time: 0.010614 seconds
Process-2135, Step 2: Resize if scale is not 1: 0.000862 seconds
Process-2135, Step 3: Get HOG channels: 8e-06 seconds
Process-2135, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2135, Step 5: Compute individual channel HOG features for the entire image: 0.155151 seconds
Process-2135, Step 6: Misc initializations: 1e-06 seconds
Process-2135, Step 7: for loop: 0.009997 seconds

Process-2135All steps time: 0.1766420000000002 seconds

Process-2135, Find cars processing time: 0.176703 seconds

Process-2133, Step 1: Divide with 255, processing time: 0.01072 seconds
Process-2133, Step 2: Resize if scale is not 1: 0.000892 seconds
Process-2133, Step 3: Get HOG channels: 8e-06 seconds
Process-2133, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2133, Step 5: Compute individual channel HOG features for the entire image: 0.175113 seconds
Process-2133, Step 6: Misc initializations: 0.0 seconds
Process-2133, Step 7: for loop: 0.009509 seconds

Process-2133All steps time: 0.1962499999999998 seconds

Process-2133, Find cars processing time: 0.196318 seconds

Process-2136, Step 1: Divide with 255, processing time: 0.010234 seconds
Process-2136, Step 2: Resize if scale is not 1: 0.000847 seconds
Process-2136, Step 3: Get HOG channels: 8e-06 seconds
Process-2136, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2136, Step 5: Compute individual channel HOG features for the entire image: 0.144785 seconds
Process-2136, Step 6: Misc initializations: 1e-06 seconds
Process-2136, Step 7: for loop: 0.008746 seconds

Process-2136All steps time: 0.164628 seconds

Process-2136, Find cars processing time: 0.164696 seconds

Process-2133, Step 1: Divide with 255, processing time: 0.026721 seconds
Process-2133, Step 2: Resize if scale is not 1: 0.000518 seconds
Process-2133, Step 3: Get HOG channels: 5e-06 seconds
Process-2133, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2133, Step 5: Compute individual channel HOG features for the entire image: 0.144301 seconds
Process-2133, Step 6: Misc initializations: 0.0 seconds
Process-2133, Step 7: for loop: 0.007106 seconds

Process-2133All steps time: 0.178657 seconds

Process-2133, Find cars processing time: 0.178725 seconds

Process-2134, Step 1: Divide with 255, processing time: 0.010221 seconds
Process-2134, Step 2: Resize if scale is not 1: 0.000822 seconds
Process-2134, Step 3: Get HOG channels: 8e-06 seconds
Process-2134, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2134, Step 5: Compute individual channel HOG features for the entire image: 0.178682 seconds
Process-2134, Step 6: Misc initializations: 1e-06 seconds
Process-2134, Step 7: for loop: 0.008884 seconds

Process-2134All steps time: 0.198626 seconds

Process-2134, Find cars processing time: 0.1987 seconds

Process-2135, Step 1: Divide with 255, processing time: 0.010501 seconds
Process-2135, Step 2: Resize if scale is not 1: 0.000936 seconds
Process-2135, Step 3: Get HOG channels: 8e-06 seconds
Process-2135, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2135, Step 5: Compute individual channel HOG features for the entire image: 0.138759 seconds
Process-2135, Step 6: Misc initializations: 0.0 seconds
Process-2135, Step 7: for loop: 0.008945 seconds
Process-2135All steps time: 0.159157 seconds

Process-2135, Find cars processing time: 0.159218 seconds

Process-2136, Step 1: Divide with 255, processing time: 0.010394 seconds
Process-2136, Step 2: Resize if scale is not 1: 0.000882 seconds
Process-2136, Step 3: Get HOG channels: 8e-06 seconds
Process-2136, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2136, Step 5: Compute individual channel HOG features for the entire image: 0.142867 seconds
Process-2136, Step 6: Misc initializations: 0.0 seconds
Process-2136, Step 7: for loop: 0.007244 seconds

Process-2136All steps time: 0.161404 seconds

Process-2136, Find cars processing time: 0.161481 seconds

Process-2134, Step 1: Divide with 255, processing time: 0.010236 seconds
Process-2134, Step 2: Resize if scale is not 1: 0.000836 seconds
Process-2134, Step 3: Get HOG channels: 9e-06 seconds
Process-2134, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2134, Step 5: Compute individual channel HOG features for the entire image: 0.142975 seconds
Process-2134, Step 6: Misc initializations: 1e-06 seconds
Process-2134, Step 7: for loop: 0.007339 seconds

Process-2134All steps time: 0.161404 seconds

Process-2134, Find cars processing time: 0.161475 seconds

Process-2133, Step 1: Divide with 255, processing time: 0.010454 seconds
Process-2133, Step 2: Resize if scale is not 1: 0.000892 seconds
Process-2133, Step 3: Get HOG channels: 8e-06 seconds
Process-2133, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2133, Step 5: Compute individual channel HOG features for the entire image: 0.143165 seconds
Process-2133, Step 6: Misc initializations: 1e-06 seconds
Process-2133, Step 7: for loop: 0.007299 seconds

Process-2133All steps time: 0.161826 seconds

Process-2133, Find cars processing time: 0.161893 seconds

Process-2135, Step 1: Divide with 255, processing time: 0.01038 seconds
Process-2135, Step 2: Resize if scale is not 1: 0.000695 seconds
Process-2135, Step 3: Get HOG channels: 8e-06 seconds
Process-2135, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2135, Step 5: Compute individual channel HOG features for the entire image: 0.103912 seconds
Process-2135, Step 6: Misc initializations: 1e-06 seconds
Process-2135, Step 7: for loop: 0.004251 seconds

Process-2135All steps time: 0.11925500000000001 seconds

Process-2135, Find cars processing time: 0.119324 seconds

Process-2136, Step 1: Divide with 255, processing time: 0.009251 seconds
Process-2136, Step 2: Resize if scale is not 1: 0.001091 seconds
Process-2136, Step 3: Get HOG channels: 8e-06 seconds
Process-2136, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2136, Step 5: Compute individual channel HOG features for the entire image: 0.139205 seconds
Process-2136, Step 6: Misc initializations: 1e-06 seconds
Process-2136, Step 7: for loop: 0.007115 seconds

Process-2136All steps time: 0.156678 seconds

Process-2136, Find cars processing time: 0.156749 seconds

Process-2136, Step 1: Divide with 255, processing time: 0.009825 seconds
Process-2136, Step 2: Resize if scale is not 1: 0.001104 seconds
Process-2136, Step 3: Get HOG channels: 5e-06 seconds
Process-2136, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2136, Step 5: Compute individual channel HOG features for the entire image: 0.094185 seconds
Process-2136, Step 6: Misc initializations: 1e-06 seconds
Process-2136, Step 7: for loop: 0.005207 seconds

Process-2136All steps time: 0.11033400000000002 seconds

Process-2136, Find cars processing time: 0.11041 seconds

Process-2134, Step 1: Divide with 255, processing time: 0.009689 seconds
Process-2134, Step 2: Resize if scale is not 1: 0.001062 seconds
Process-2134, Step 3: Get HOG channels: 9e-06 seconds
Process-2134, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2134, Step 5: Compute individual channel HOG features for the entire image: 0.14377 seconds
Process-2134, Step 6: Misc initializations: 1e-06 seconds
Process-2134, Step 7: for loop: 0.009939 seconds

Process-2134All steps time: 0.16447900000000001 seconds

Process-2134, Find cars processing time: 0.164552 seconds

```
Process-2133, Step 1: Divide with 255, processing time: 0.010544 seconds
Process-2133, Step 2: Resize if scale is not 1: 0.000736 seconds
Process-2133, Step 3: Get HOG channels: 5e-06 seconds
Process-2133, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2133, Step 5: Compute individual channel HOG features for the entire image: 0.125046 seconds
Process-2133, Step 6: Misc initializations: 1e-06 seconds
Process-2133, Step 7: for loop: 0.007061 seconds

Process-2133All steps time: 0.143398 seconds

Process-2135, Step 1: Divide with 255, processing time: 0.007342 seconds
Process-2133, Find cars processing time: 0.143445 seconds
Process-2135, Step 2: Resize if scale is not 1: 0.000983 seconds

Process-2135, Step 3: Get HOG channels: 7e-06 seconds
Process-2135, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2135, Step 5: Compute individual channel HOG features for the entire image: 0.139493 seconds
Process-2135, Step 6: Misc initializations: 1e-06 seconds
Process-2135, Step 7: for loop: 0.006301 seconds

Process-2135All steps time: 0.154135 seconds

Process-2135, Find cars processing time: 0.154224 seconds

The times for each task are: [0.290822, 0.251843, 0.29155, 0.213208, 0.209049, 0.261961, 0.20343

Minimum: 0.11041 Maximum: 0.29155 Average: 0.1877 seconds

*****
Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]
4 processes used for testing 5 window sizes
Processing times for each image [3.4401, 3.7671, 3.8213, 3.6782, 4.1558, 3.6737] with an average

Time elapsed so far... 425.27010000000007
#####

Number of processes used: 5 window size 260
Length of task list: 22
Number of processes used: 5

Process-2137, Step 1: Divide with 255, processing time: 0.028166 seconds
Process-2137, Step 2: Resize if scale is not 1: 0.002002 seconds
Process-2137, Step 3: Get HOG channels: 1.9e-05 seconds
Process-2137, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2137, Step 5: Compute individual channel HOG features for the entire image: 0.200618 seconds
Process-2137, Step 6: Misc initializations: 1e-06 seconds
Process-2137, Step 7: for loop: 0.014332 seconds
```

Process-2137All steps time: 0.245144 seconds

Process-2137, Find cars processing time: 0.245316 seconds

Process-2138, Step 1: Divide with 255, processing time: 0.032946 seconds

Process-2138, Step 2: Resize if scale is not 1: 0.002432 seconds

Process-2138, Step 3: Get HOG channels: 2.8e-05 seconds

Process-2138, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2138, Step 5: Compute individual channel HOG features for the entire image: 0.259231 seconds

Process-2138, Step 6: Misc initializations: 1e-06 seconds

Process-2138, Step 7: for loop: 0.017939 seconds

Process-2138All steps time: 0.3125829999999994 seconds

Process-2138, Find cars processing time: 0.31278 seconds

Process-2139, Step 1: Divide with 255, processing time: 0.062798 seconds

Process-2139, Step 2: Resize if scale is not 1: 0.004446 seconds

Process-2139, Step 3: Get HOG channels: 5.8e-05 seconds

Process-2139, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2139, Step 5: Compute individual channel HOG features for the entire image: 0.271021 seconds

Process-2139, Step 6: Misc initializations: 1e-06 seconds

Process-2139, Step 7: for loop: 0.015836 seconds

Process-2139All steps time: 0.354165 seconds

Process-2139, Find cars processing time: 0.354378 seconds

Process-2137, Step 1: Divide with 255, processing time: 0.010447 seconds

Process-2137, Step 2: Resize if scale is not 1: 0.00105 seconds

Process-2137, Step 3: Get HOG channels: 1.2e-05 seconds

Process-2137, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2137, Step 5: Compute individual channel HOG features for the entire image: 0.282636 seconds

Process-2137, Step 6: Misc initializations: 1e-06 seconds

Process-2137, Step 7: for loop: 0.014804 seconds

Process-2137All steps time: 0.308957 seconds

Process-2137, Find cars processing time: 0.309045 seconds

Process-2141, Step 1: Divide with 255, processing time: 0.02488 seconds

Process-2141, Step 2: Resize if scale is not 1: 0.001799 seconds

Process-2141, Step 3: Get HOG channels: 1.8e-05 seconds

Process-2141, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2141, Step 5: Compute individual channel HOG features for the entire image: 0.200302 seconds

Process-2141, Step 6: Misc initializations: 1e-06 seconds

Process-2141, Step 7: for loop: 0.011737 seconds

Process-2141All steps time: 0.238743 seconds

Process-2141, Find cars processing time: 0.238875 seconds

Process-2139, Step 1: Divide with 255, processing time: 0.008594 seconds

Process-2139, Step 2: Resize if scale is not 1: 0.000804 seconds

Process-2139, Step 3: Get HOG channels: 8e-06 seconds

Process-2139, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2139, Step 5: Compute individual channel HOG features for the entire image: 0.208587 seconds

Process-2139, Step 6: Misc initializations: 1e-06 seconds

Process-2139, Step 7: for loop: 0.012943 seconds

Process-2139All steps time: 0.230943 seconds

Process-2139, Find cars processing time: 0.231004 seconds

Process-2138, Step 1: Divide with 255, processing time: 0.010297 seconds

Process-2138, Step 2: Resize if scale is not 1: 0.000925 seconds

Process-2138, Step 3: Get HOG channels: 9e-06 seconds

Process-2138, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2138, Step 5: Compute individual channel HOG features for the entire image: 0.170703 seconds

Process-2138, Step 6: Misc initializations: 1e-06 seconds

Process-2138, Step 7: for loop: 0.010433 seconds

Process-2138All steps time: 0.192375 seconds

Process-2138, Find cars processing time: 0.192466 seconds

Process-2140, Step 1: Divide with 255, processing time: 0.026447 seconds

Process-2140, Step 2: Resize if scale is not 1: 0.001599 seconds

Process-2140, Step 3: Get HOG channels: 1.5e-05 seconds

Process-2140, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2140, Step 5: Compute individual channel HOG features for the entire image: 0.166687 seconds

Process-2140, Step 6: Misc initializations: 1e-06 seconds

Process-2140, Step 7: for loop: 0.010374 seconds

Process-2140All steps time: 0.205127 seconds

Process-2140, Find cars processing time: 0.205293 seconds

Process-2137, Step 1: Divide with 255, processing time: 0.009492 seconds

Process-2137, Step 2: Resize if scale is not 1: 0.000715 seconds

Process-2137, Step 3: Get HOG channels: 7e-06 seconds

Process-2137, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2137, Step 5: Compute individual channel HOG features for the entire image: 0.179513 seconds

Process-2137, Step 6: Misc initializations: 0.0 seconds

Process-2137, Step 7: for loop: 0.009516 seconds

Process-2137All steps time: 0.199248 seconds

Process-2137, Find cars processing time: 0.199309 seconds

Process-2139, Step 1: Divide with 255, processing time: 0.0065 seconds

Process-2139, Step 2: Resize if scale is not 1: 0.000542 seconds

Process-2139, Step 3: Get HOG channels: 6e-06 seconds

Process-2139, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2139, Step 5: Compute individual channel HOG features for the entire image: 0.18999 seconds

Process-2139, Step 6: Misc initializations: 1e-06 seconds

Process-2139, Step 7: for loop: 0.010372 seconds

Process-2139All steps time: 0.207416 seconds

Process-2139, Find cars processing time: 0.207478 seconds

Process-2138, Step 1: Divide with 255, processing time: 0.009776 seconds

Process-2138, Step 2: Resize if scale is not 1: 0.000662 seconds

Process-2138, Step 3: Get HOG channels: 5e-06 seconds

Process-2138, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2138, Step 5: Compute individual channel HOG features for the entire image: 0.136148 seconds

Process-2138, Step 6: Misc initializations: 1e-06 seconds

Process-2138, Step 7: for loop: 0.015283 seconds

Process-2138All steps time: 0.16188 seconds

Process-2138, Find cars processing time: 0.161948 seconds

Process-2141, Step 1: Divide with 255, processing time: 0.010595 seconds

Process-2141, Step 2: Resize if scale is not 1: 0.000934 seconds

Process-2141, Step 3: Get HOG channels: 9e-06 seconds

Process-2141, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2141, Step 5: Compute individual channel HOG features for the entire image: 0.172208 seconds

Process-2141, Step 6: Misc initializations: 1e-06 seconds

Process-2141, Step 7: for loop: 0.00978 seconds

Process-2141All steps time: 0.193535 seconds

Process-2141, Find cars processing time: 0.193614 seconds

Process-2137, Step 1: Divide with 255, processing time: 0.008901 seconds

Process-2137, Step 2: Resize if scale is not 1: 0.000486 seconds

Process-2137, Step 3: Get HOG channels: 5e-06 seconds

Process-2137, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2137, Step 5: Compute individual channel HOG features for the entire image: 0.164755 seconds

Process-2137, Step 6: Misc initializations: 1e-06 seconds

Process-2137, Step 7: for loop: 0.007192 seconds

Process-2137All steps time: 0.181345 seconds

Process-2137, Find cars processing time: 0.181417 seconds

Process-2140, Step 1: Divide with 255, processing time: 0.010752 seconds

Process-2140, Step 2: Resize if scale is not 1: 0.000875 seconds

Process-2140, Step 3: Get HOG channels: 8e-06 seconds

Process-2140, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2140, Step 5: Compute individual channel HOG features for the entire image: 0.108815 seconds

Process-2140, Step 6: Misc initializations: 1e-06 seconds

Process-2140, Step 7: for loop: 0.005899 seconds

Process-2140All steps time: 0.126359 seconds

Process-2140, Find cars processing time: 0.126433 seconds

Process-2138, Step 1: Divide with 255, processing time: 0.010994 seconds

Process-2138, Step 2: Resize if scale is not 1: 0.001058 seconds

Process-2138, Step 3: Get HOG channels: 9e-06 seconds

Process-2138, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2138, Step 5: Compute individual channel HOG features for the entire image: 0.144371 seconds

Process-2138, Step 6: Misc initializations: 1e-06 seconds

Process-2138, Step 7: for loop: 0.0072 seconds

Process-2139, Step 1: Divide with 255, processing time: 0.008836 seconds

Process-2139, Step 2: Resize if scale is not 1: 0.000679 seconds

Process-2138All steps time: 0.163641 seconds

Process-2139, Step 3: Get HOG channels: 7e-06 seconds

Process-2139, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2138, Find cars processing time: 0.163727 seconds

Process-2139, Step 5: Compute individual channel HOG features for the entire image: 0.161723 seconds

Process-2139, Step 6: Misc initializations: 1e-06 seconds

Process-2139, Step 7: for loop: 0.007409 seconds

Process-2139All steps time: 0.17866200000000002 seconds

Process-2139, Find cars processing time: 0.178731 seconds

Process-2141, Step 1: Divide with 255, processing time: 0.009434 seconds

Process-2141, Step 2: Resize if scale is not 1: 0.001294 seconds

Process-2141, Step 3: Get HOG channels: 7e-06 seconds

Process-2141, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2141, Step 5: Compute individual channel HOG features for the entire image: 0.127526 seconds

Process-2141, Step 6: Misc initializations: 1e-06 seconds

Process-2141, Step 7: for loop: 0.004953 seconds

Process-2141All steps time: 0.14322200000000002 seconds

Process-2141, Find cars processing time: 0.143287 seconds

Process-2140, Step 1: Divide with 255, processing time: 0.009004 seconds

Process-2140, Step 2: Resize if scale is not 1: 0.000709 seconds

Process-2140, Step 3: Get HOG channels: 5e-06 seconds

Process-2140, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2140, Step 5: Compute individual channel HOG features for the entire image: 0.126037 seconds

Process-2140, Step 6: Misc initializations: 0.0 seconds

Process-2140, Step 7: for loop: 0.007328 seconds

Process-2140All steps time: 0.14308700000000002 seconds

Process-2140, Find cars processing time: 0.143141 seconds

Process-2137, Step 1: Divide with 255, processing time: 0.010389 seconds

Process-2137, Step 2: Resize if scale is not 1: 0.001219 seconds

Process-2137, Step 3: Get HOG channels: 9e-06 seconds

Process-2137, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2137, Step 5: Compute individual channel HOG features for the entire image: 0.128096 seconds

Process-2137, Step 6: Misc initializations: 1e-06 seconds

Process-2137, Step 7: for loop: 0.007308 seconds

Process-2137All steps time: 0.147029 seconds

Process-2137, Find cars processing time: 0.147114 seconds

Process-2138, Step 1: Divide with 255, processing time: 0.010523 seconds

Process-2138, Step 2: Resize if scale is not 1: 0.000949 seconds

Process-2138, Step 3: Get HOG channels: 5e-06 seconds

Process-2138, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2138, Step 5: Compute individual channel HOG features for the entire image: 0.141513 seconds

Process-2138, Step 6: Misc initializations: 1e-06 seconds

Process-2138, Step 7: for loop: 0.007066 seconds

Process-2138All steps time: 0.1600619999999998 seconds

Process-2138, Find cars processing time: 0.160128 seconds

Process-2139, Step 1: Divide with 255, processing time: 0.006452 seconds

Process-2139, Step 2: Resize if scale is not 1: 0.000779 seconds

Process-2139, Step 3: Get HOG channels: 6e-06 seconds

Process-2139, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2139, Step 5: Compute individual channel HOG features for the entire image: 0.137406 seconds

Process-2139, Step 6: Misc initializations: 1e-06 seconds

Process-2139, Step 7: for loop: 0.007319 seconds

Process-2139All steps time: 0.151968 seconds

Process-2139, Find cars processing time: 0.152033 seconds

Process-2141, Step 1: Divide with 255, processing time: 0.010177 seconds

Process-2141, Step 2: Resize if scale is not 1: 0.001852 seconds

Process-2141, Step 3: Get HOG channels: 8e-06 seconds

Process-2141, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2141, Step 5: Compute individual channel HOG features for the entire image: 0.133419 seconds

Process-2141, Step 6: Misc initializations: 0.0 seconds

Process-2141, Step 7: for loop: 0.006154 seconds

Process-2141All steps time: 0.151618 seconds

Process-2141, Find cars processing time: 0.151695 seconds

The times for each task are: [0.245316, 0.31278, 0.354378, 0.309045, 0.238875, 0.231004, 0.19246

Minimum: 0.126433 Maximum: 0.354378 Average: 0.2 seconds

Number of processes used: 5 window size 260

Length of task list: 22

Number of processes used: 5

Process-2142, Step 1: Divide with 255, processing time: 0.023798 seconds

Process-2142, Step 2: Resize if scale is not 1: 0.002061 seconds

Process-2142, Step 3: Get HOG channels: 2.2e-05 seconds

Process-2142, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2142, Step 5: Compute individual channel HOG features for the entire image: 0.259067 seconds

Process-2142, Step 6: Misc initializations: 1e-06 seconds

Process-2142, Step 7: for loop: 0.0174 seconds

Process-2142All steps time: 0.3023559999999996 seconds

Process-2142, Find cars processing time: 0.302541 seconds

Process-2143, Step 1: Divide with 255, processing time: 0.029264 seconds

Process-2143, Step 2: Resize if scale is not 1: 0.002146 seconds

Process-2143, Step 3: Get HOG channels: 2.3e-05 seconds

Process-2143, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2143, Step 5: Compute individual channel HOG features for the entire image: 0.250059 seconds

Process-2143, Step 6: Misc initializations: 1e-06 seconds

Process-2143, Step 7: for loop: 0.017899 seconds

Process-2143All steps time: 0.2993979999999994 seconds

Process-2143, Find cars processing time: 0.299575 seconds

Process-2142, Step 1: Divide with 255, processing time: 0.010881 seconds

Process-2142, Step 2: Resize if scale is not 1: 0.000615 seconds

Process-2142, Step 3: Get HOG channels: 6e-06 seconds

Process-2142, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2142, Step 5: Compute individual channel HOG features for the entire image: 0.267193 seconds

Process-2142, Step 6: Misc initializations: 1e-06 seconds

Process-2142, Step 7: for loop: 0.014426 seconds

Process-2142All steps time: 0.29313 seconds

Process-2142, Find cars processing time: 0.293216 seconds

Process-2144, Step 1: Divide with 255, processing time: 0.030977 seconds

Process-2144, Step 2: Resize if scale is not 1: 0.002682 seconds

Process-2144, Step 3: Get HOG channels: 2.5e-05 seconds

Process-2144, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2144, Step 5: Compute individual channel HOG features for the entire image: 0.297048 seconds

Process-2144, Step 6: Misc initializations: 1e-06 seconds

Process-2144, Step 7: for loop: 0.019303 seconds

Process-2144All steps time: 0.3500439999999997 seconds

Process-2144, Find cars processing time: 0.35024 seconds

Process-2145, Step 1: Divide with 255, processing time: 0.028898 seconds

Process-2145, Step 2: Resize if scale is not 1: 0.002008 seconds

Process-2145, Step 3: Get HOG channels: 2.4e-05 seconds

Process-2145, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2145, Step 5: Compute individual channel HOG features for the entire image: 0.226208 seconds

Process-2145, Step 6: Misc initializations: 0.0 seconds

Process-2145, Step 7: for loop: 0.010999 seconds

Process-2145All steps time: 0.268144 seconds

Process-2145, Find cars processing time: 0.268311 seconds

Process-2146, Step 1: Divide with 255, processing time: 0.030317 seconds

Process-2146, Step 2: Resize if scale is not 1: 0.002516 seconds

Process-2146, Step 3: Get HOG channels: 2.5e-05 seconds

Process-2146, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2146, Step 5: Compute individual channel HOG features for the entire image: 0.230789 seconds

Process-2146, Step 6: Misc initializations: 1e-06 seconds

Process-2146, Step 7: for loop: 0.011801 seconds

Process-2146All steps time: 0.2754569999999995 seconds

Process-2146, Find cars processing time: 0.275619 seconds

Process-2142, Step 1: Divide with 255, processing time: 0.010343 seconds

Process-2142, Step 2: Resize if scale is not 1: 0.00095 seconds

Process-2142, Step 3: Get HOG channels: 8e-06 seconds

Process-2142, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2142, Step 5: Compute individual channel HOG features for the entire image: 0.218329 seconds

Process-2142, Step 6: Misc initializations: 1e-06 seconds

Process-2142, Step 7: for loop: 0.012374 seconds

Process-2142All steps time: 0.2420129999999998 seconds

Process-2142, Find cars processing time: 0.242086 seconds

Process-2144, Step 1: Divide with 255, processing time: 0.010708 seconds

Process-2144, Step 2: Resize if scale is not 1: 0.000932 seconds

Process-2144, Step 3: Get HOG channels: 8e-06 seconds

Process-2144, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2144, Step 5: Compute individual channel HOG features for the entire image: 0.180817 seconds

Process-2144, Step 6: Misc initializations: 1e-06 seconds

Process-2144, Step 7: for loop: 0.00984 seconds

Process-2144All steps time: 0.202315 seconds

Process-2144, Find cars processing time: 0.202406 seconds

Process-2143, Step 1: Divide with 255, processing time: 0.009111 seconds

Process-2143, Step 2: Resize if scale is not 1: 0.000666 seconds

Process-2143, Step 3: Get HOG channels: 6e-06 seconds

Process-2143, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2143, Step 5: Compute individual channel HOG features for the entire image: 0.211426 seconds

Process-2143, Step 6: Misc initializations: 1e-06 seconds

Process-2143, Step 7: for loop: 0.010074 seconds

Process-2143All steps time: 0.231289 seconds

Process-2143, Find cars processing time: 0.231344 seconds

Process-2142, Step 1: Divide with 255, processing time: 0.009872 seconds

Process-2142, Step 2: Resize if scale is not 1: 0.000808 seconds

Process-2142, Step 3: Get HOG channels: 8e-06 seconds

Process-2142, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2142, Step 5: Compute individual channel HOG features for the entire image: 0.131785 seconds

Process-2142, Step 6: Misc initializations: 0.0 seconds

Process-2142, Step 7: for loop: 0.007403 seconds

Process-2142All steps time: 0.1498830000000002 seconds

Process-2142, Find cars processing time: 0.149943 seconds

Process-2143, Step 1: Divide with 255, processing time: 0.009162 seconds

Process-2143, Step 2: Resize if scale is not 1: 0.000782 seconds

Process-2143, Step 3: Get HOG channels: 7e-06 seconds

Process-2143, Step 4: Define blocks and steps as above: 1.4e-05 seconds

Process-2143, Step 5: Compute individual channel HOG features for the entire image: 0.143039 seconds

Process-2143, Step 6: Misc initializations: 1e-06 seconds

Process-2143, Step 7: for loop: 0.007095 seconds

Process-2143All steps time: 0.1601 seconds

Process-2143, Find cars processing time: 0.160166 seconds

Process-2145, Step 1: Divide with 255, processing time: 0.010777 seconds

Process-2145, Step 2: Resize if scale is not 1: 0.000617 seconds

Process-2145, Step 3: Get HOG channels: 6e-06 seconds

Process-2145, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2145, Step 5: Compute individual channel HOG features for the entire image: 0.191276 seconds

Process-2145, Step 6: Misc initializations: 1e-06 seconds

Process-2145, Step 7: for loop: 0.009107 seconds

Process-2145All steps time: 0.211789 seconds

Process-2145, Find cars processing time: 0.211891 seconds

Process-2144, Step 1: Divide with 255, processing time: 0.010135 seconds

Process-2144, Step 2: Resize if scale is not 1: 0.000545 seconds

Process-2144, Step 3: Get HOG channels: 5e-06 seconds

Process-2144, Step 4: Define blocks and steps as above: 4e-06 seconds

Process-2144, Step 5: Compute individual channel HOG features for the entire image: 0.147885 seconds

Process-2144, Step 6: Misc initializations: 1e-06 seconds

Process-2144, Step 7: for loop: 0.007444 seconds

Process-2144All steps time: 0.166019 seconds

Process-2144, Find cars processing time: 0.166085 seconds

Process-2146, Step 1: Divide with 255, processing time: 0.010574 seconds

Process-2146, Step 2: Resize if scale is not 1: 0.000704 seconds

Process-2146, Step 3: Get HOG channels: 7e-06 seconds

Process-2146, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2146, Step 5: Compute individual channel HOG features for the entire image: 0.148812 seconds

Process-2146, Step 6: Misc initializations: 1e-06 seconds

Process-2146, Step 7: for loop: 0.00792 seconds

Process-2146All steps time: 0.168023 seconds

Process-2146, Find cars processing time: 0.168084 seconds

Process-2142, Step 1: Divide with 255, processing time: 0.010358 seconds

Process-2145, Step 1: Divide with 255, processing time: 0.007051 seconds

Process-2142, Step 2: Resize if scale is not 1: 0.000918 seconds

Process-2142, Step 3: Get HOG channels: 9e-06 seconds

Process-2145, Step 2: Resize if scale is not 1: 0.001312 seconds

Process-2145, Step 3: Get HOG channels: 7e-06 seconds

Process-2142, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2145, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2145, Step 5: Compute individual channel HOG features for the entire image: 0.137308 seconds

Process-2145, Step 6: Misc initializations: 1e-06 seconds

Process-2145, Step 7: for loop: 0.004218 seconds

Process-2143, Step 1: Divide with 255, processing time: 0.009817 seconds

Process-2143, Step 2: Resize if scale is not 1: 0.00075 seconds

Process-2145All steps time: 0.149904 seconds

Process-2143, Step 3: Get HOG channels: 7e-06 seconds

Process-2143, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2145, Find cars processing time: 0.149967 seconds

Process-2143, Step 5: Compute individual channel HOG features for the entire image: 0.174694 seconds

Process-2143, Step 6: Misc initializations: 1e-06 seconds

Process-2143, Step 7: for loop: 0.00726 seconds

Process-2143All steps time: 0.1925369999999999 seconds

Process-2142, Step 5: Compute individual channel HOG features for the entire image: 0.115074 seconds

Process-2143, Find cars processing time: 0.192611 seconds

Process-2142, Step 6: Misc initializations: 0.0 seconds

Process-2142, Step 7: for loop: 0.005826 seconds

Process-2142All steps time: 0.132193 seconds

Process-2142, Find cars processing time: 0.132251 seconds

Process-2146, Step 1: Divide with 255, processing time: 0.009765 seconds

Process-2146, Step 2: Resize if scale is not 1: 0.000861 seconds

Process-2143, Step 1: Divide with 255, processing time: 0.007295 seconds

Process-2143, Step 2: Resize if scale is not 1: 0.00099 seconds

Process-2146, Step 3: Get HOG channels: 6e-06 seconds

Process-2143, Step 3: Get HOG channels: 6e-06 seconds

Process-2143, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2146, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2143, Step 5: Compute individual channel HOG features for the entire image: 0.108421 seconds

Process-2143, Step 6: Misc initializations: 0.0 seconds

Process-2146, Step 5: Compute individual channel HOG features for the entire image: 0.127586 seconds
Process-2143, Step 7: for loop: 0.006385 seconds
Process-2146, Step 6: Misc initializations: 0.0 seconds

Process-2146, Step 7: for loop: 0.007003 seconds
Process-2143All steps time: 0.123103 seconds

Process-2143, Find cars processing time: 0.123183 seconds
Process-2146All steps time: 0.14522800000000002 seconds

Process-2146, Find cars processing time: 0.145294 seconds

Process-2142, Step 1: Divide with 255, processing time: 0.010744 seconds
Process-2142, Step 2: Resize if scale is not 1: 0.001187 seconds
Process-2142, Step 3: Get HOG channels: 5e-06 seconds
Process-2142, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2142, Step 5: Compute individual channel HOG features for the entire image: 0.11079 seconds
Process-2142, Step 6: Misc initializations: 1e-06 seconds
Process-2142, Step 7: for loop: 0.006015 seconds

Process-2142All steps time: 0.128747 seconds

Process-2142, Find cars processing time: 0.128807 seconds

Process-2144, Step 1: Divide with 255, processing time: 0.010996 seconds
Process-2144, Step 2: Resize if scale is not 1: 0.001256 seconds
Process-2144, Step 3: Get HOG channels: 9e-06 seconds
Process-2144, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2144, Step 5: Compute individual channel HOG features for the entire image: 0.148244 seconds
Process-2144, Step 6: Misc initializations: 1e-06 seconds
Process-2144, Step 7: for loop: 0.00754 seconds

Process-2144All steps time: 0.1680539999999998 seconds

Process-2144, Find cars processing time: 0.168136 seconds

Process-2145, Step 1: Divide with 255, processing time: 0.006356 seconds
Process-2145, Step 2: Resize if scale is not 1: 0.000791 seconds
Process-2145, Step 3: Get HOG channels: 5e-06 seconds
Process-2145, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2145, Step 5: Compute individual channel HOG features for the entire image: 0.131957 seconds
Process-2145, Step 6: Misc initializations: 1e-06 seconds
Process-2145, Step 7: for loop: 0.007873 seconds

Process-2145All steps time: 0.1469869999999998 seconds

Process-2145, Find cars processing time: 0.147063 seconds

The times for each task are: [0.302541, 0.299575, 0.293216, 0.35024, 0.268311, 0.275619, 0.24208

Minimum: 0.123183 Maximum: 0.35024 Average: 0.2049 seconds

Number of processes used: 5 window size 260

Length of task list: 22

Number of processes used: 5

Process-2147, Step 1: Divide with 255, processing time: 0.028319 seconds

Process-2147, Step 2: Resize if scale is not 1: 0.002118 seconds

Process-2147, Step 3: Get HOG channels: 2.1e-05 seconds

Process-2147, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2147, Step 5: Compute individual channel HOG features for the entire image: 0.243723 sec

Process-2147, Step 6: Misc initializations: 1e-06 seconds

Process-2147, Step 7: for loop: 0.017048 seconds

Process-2147All steps time: 0.291236 seconds

Process-2147, Find cars processing time: 0.29141 seconds

Process-2148, Step 1: Divide with 255, processing time: 0.030644 seconds

Process-2148, Step 2: Resize if scale is not 1: 0.002185 seconds

Process-2148, Step 3: Get HOG channels: 7.9e-05 seconds

Process-2148, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2148, Step 5: Compute individual channel HOG features for the entire image: 0.242504 sec

Process-2148, Step 6: Misc initializations: 1e-06 seconds

Process-2148, Step 7: for loop: 0.015585 seconds

Process-2148All steps time: 0.291005 seconds

Process-2148, Find cars processing time: 0.291193 seconds

Process-2149, Step 1: Divide with 255, processing time: 0.044439 seconds

Process-2149, Step 2: Resize if scale is not 1: 0.002265 seconds

Process-2149, Step 3: Get HOG channels: 3.1e-05 seconds

Process-2149, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2149, Step 5: Compute individual channel HOG features for the entire image: 0.24621 sec

Process-2149, Step 6: Misc initializations: 1.2e-05 seconds

Process-2149, Step 7: for loop: 0.022567 seconds

Process-2149All steps time: 0.315531 seconds

Process-2149, Find cars processing time: 0.315709 seconds

Process-2149, Step 1: Divide with 255, processing time: 0.010131 seconds

Process-2149, Step 2: Resize if scale is not 1: 0.000509 seconds
Process-2149, Step 3: Get HOG channels: 5e-06 seconds
Process-2149, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2149, Step 5: Compute individual channel HOG features for the entire image: 0.161109 seconds
Process-2149, Step 6: Misc initializations: 1e-06 seconds
Process-2149, Step 7: for loop: 0.009565 seconds

Process-2149All steps time: 0.181325 seconds

Process-2149, Find cars processing time: 0.181385 seconds

Process-2147, Step 1: Divide with 255, processing time: 0.010683 seconds
Process-2147, Step 2: Resize if scale is not 1: 0.000624 seconds
Process-2147, Step 3: Get HOG channels: 6e-06 seconds
Process-2147, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2147, Step 5: Compute individual channel HOG features for the entire image: 0.214882 seconds
Process-2147, Step 6: Misc initializations: 1e-06 seconds
Process-2147, Step 7: for loop: 0.013121 seconds

Process-2147All steps time: 0.23932199999999998 seconds

Process-2147, Find cars processing time: 0.239405 seconds

Process-2150, Step 1: Divide with 255, processing time: 0.028439 seconds
Process-2150, Step 2: Resize if scale is not 1: 0.002068 seconds
Process-2150, Step 3: Get HOG channels: 2e-05 seconds
Process-2150, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2150, Step 5: Compute individual channel HOG features for the entire image: 0.237573 seconds
Process-2150, Step 6: Misc initializations: 1e-06 seconds
Process-2150, Step 7: for loop: 0.014309 seconds

Process-2150All steps time: 0.2824149999999997 seconds

Process-2150, Find cars processing time: 0.282556 seconds

Process-2151, Step 1: Divide with 255, processing time: 0.028439 seconds
Process-2151, Step 2: Resize if scale is not 1: 0.002443 seconds
Process-2151, Step 3: Get HOG channels: 2.3e-05 seconds
Process-2151, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2151, Step 5: Compute individual channel HOG features for the entire image: 0.214219 seconds
Process-2151, Step 6: Misc initializations: 1e-06 seconds
Process-2148, Step 1: Divide with 255, processing time: 0.007561 seconds
Process-2151, Step 7: for loop: 0.013276 seconds

Process-2148, Step 2: Resize if scale is not 1: 0.000729 seconds
Process-2151All steps time: 0.2584059999999997 seconds
Process-2148, Step 3: Get HOG channels: 8e-06 seconds

Process-2148, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2151, Find cars processing time: 0.258574 seconds
Process-2148, Step 5: Compute individual channel HOG features for the entire image: 0.229358 seconds

Process-2148, Step 6: Misc initializations: 1e-06 seconds
Process-2148, Step 7: for loop: 0.013626 seconds

Process-2148All steps time: 0.251288 seconds
Process-2147, Step 1: Divide with 255, processing time: 0.008601 seconds

Process-2147, Step 2: Resize if scale is not 1: 0.000654 seconds
Process-2148, Find cars processing time: 0.251358 seconds
Process-2147, Step 3: Get HOG channels: 6e-06 seconds

Process-2147, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2147, Step 5: Compute individual channel HOG features for the entire image: 0.182177 seconds
Process-2147, Step 6: Misc initializations: 1e-06 seconds
Process-2147, Step 7: for loop: 0.011081 seconds

Process-2147All steps time: 0.202525 seconds

Process-2147, Find cars processing time: 0.202599 seconds

Process-2149, Step 1: Divide with 255, processing time: 0.010273 seconds
Process-2149, Step 2: Resize if scale is not 1: 0.000907 seconds
Process-2149, Step 3: Get HOG channels: 8e-06 seconds
Process-2149, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2149, Step 5: Compute individual channel HOG features for the entire image: 0.170418 seconds
Process-2149, Step 6: Misc initializations: 1e-06 seconds
Process-2149, Step 7: for loop: 0.009781 seconds

Process-2149All steps time: 0.191395 seconds

Process-2149, Find cars processing time: 0.191472 seconds

Process-2148, Step 1: Divide with 255, processing time: 0.010509 seconds
Process-2148, Step 2: Resize if scale is not 1: 0.001108 seconds
Process-2148, Step 3: Get HOG channels: 1.5e-05 seconds
Process-2148, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-2148, Step 5: Compute individual channel HOG features for the entire image: 0.159167 seconds
Process-2148, Step 6: Misc initializations: 1e-06 seconds
Process-2148, Step 7: for loop: 0.008645 seconds

Process-2148All steps time: 0.1794549999999998 seconds

Process-2148, Find cars processing time: 0.179551 seconds

Process-2149, Step 1: Divide with 255, processing time: 0.010627 seconds

Process-2149, Step 2: Resize if scale is not 1: 0.000826 seconds
Process-2149, Step 3: Get HOG channels: 8e-06 seconds
Process-2149, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2149, Step 5: Compute individual channel HOG features for the entire image: 0.150249 seconds
Process-2149, Step 6: Misc initializations: 0.0 seconds
Process-2149, Step 7: for loop: 0.007907 seconds

Process-2151, Step 1: Divide with 255, processing time: 0.010885 seconds
Process-2149All steps time: 0.169624 seconds
Process-2151, Step 2: Resize if scale is not 1: 0.000944 seconds

Process-2151, Step 3: Get HOG channels: 9e-06 seconds
Process-2151, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2149, Find cars processing time: 0.169694 seconds
Process-2151, Step 5: Compute individual channel HOG features for the entire image: 0.180944 seconds

Process-2151, Step 6: Misc initializations: 0.0 seconds
Process-2151, Step 7: for loop: 0.009885 seconds

Process-2151All steps time: 0.202676 seconds

Process-2151, Find cars processing time: 0.20276 seconds

Process-2150, Step 1: Divide with 255, processing time: 0.011027 seconds
Process-2150, Step 2: Resize if scale is not 1: 0.000946 seconds
Process-2150, Step 3: Get HOG channels: 8e-06 seconds
Process-2150, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2150, Step 5: Compute individual channel HOG features for the entire image: 0.145442 seconds
Process-2150, Step 6: Misc initializations: 1e-06 seconds
Process-2150, Step 7: for loop: 0.008714 seconds

Process-2150All steps time: 0.166145 seconds

Process-2150, Find cars processing time: 0.16624 seconds

Process-2150, Step 1: Divide with 255, processing time: 0.005843 seconds
Process-2150, Step 2: Resize if scale is not 1: 0.000766 seconds
Process-2150, Step 3: Get HOG channels: 1.5e-05 seconds
Process-2150, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2150, Step 5: Compute individual channel HOG features for the entire image: 0.104496 seconds
Process-2150, Step 6: Misc initializations: 0.0 seconds
Process-2150, Step 7: for loop: 0.005898 seconds

Process-2150All steps time: 0.117023 seconds

Process-2150, Find cars processing time: 0.117093 seconds

Process-2147, Step 1: Divide with 255, processing time: 0.038394 seconds

Process-2147, Step 2: Resize if scale is not 1: 0.001603 seconds
Process-2147, Step 3: Get HOG channels: 7e-06 seconds
Process-2147, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2147, Step 5: Compute individual channel HOG features for the entire image: 0.165606 seconds
Process-2147, Step 6: Misc initializations: 1e-06 seconds
Process-2147, Step 7: for loop: 0.00816 seconds

Process-2147All steps time: 0.213776 seconds

Process-2147, Find cars processing time: 0.21384 seconds

Process-2150, Step 1: Divide with 255, processing time: 0.008788 seconds
Process-2150, Step 2: Resize if scale is not 1: 0.000473 seconds
Process-2150, Step 3: Get HOG channels: 5e-06 seconds
Process-2150, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2150, Step 5: Compute individual channel HOG features for the entire image: 0.12675 seconds
Process-2150, Step 6: Misc initializations: 1e-06 seconds
Process-2150, Step 7: for loop: 0.007453 seconds

Process-2150All steps time: 0.143476 seconds

Process-2150, Find cars processing time: 0.143537 seconds

Process-2149, Step 1: Divide with 255, processing time: 0.01393 seconds
Process-2149, Step 2: Resize if scale is not 1: 0.000521 seconds
Process-2149, Step 3: Get HOG channels: 5e-06 seconds
Process-2149, Step 4: Define blocks and steps as above: 2e-05 seconds
Process-2149, Step 5: Compute individual channel HOG features for the entire image: 0.098387 seconds
Process-2149, Step 6: Misc initializations: 1e-06 seconds
Process-2149, Step 7: for loop: 0.004028 seconds

Process-2149All steps time: 0.11689200000000001 seconds

Process-2149, Find cars processing time: 0.116947 seconds

Process-2151, Step 1: Divide with 255, processing time: 0.009992 seconds
Process-2151, Step 2: Resize if scale is not 1: 0.000912 seconds
Process-2151, Step 3: Get HOG channels: 6e-06 seconds
Process-2151, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2151, Step 5: Compute individual channel HOG features for the entire image: 0.131594 seconds
Process-2151, Step 6: Misc initializations: 1e-06 seconds
Process-2151, Step 7: for loop: 0.007486 seconds

Process-2151All steps time: 0.149995 seconds

Process-2148, Step 1: Divide with 255, processing time: 0.026297 seconds
Process-2151, Find cars processing time: 0.150059 seconds
Process-2148, Step 2: Resize if scale is not 1: 0.001051 seconds

```
Process-2148, Step 3: Get HOG channels: 7e-06 seconds
Process-2148, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2148, Step 5: Compute individual channel HOG features for the entire image: 0.159702 seconds
Process-2148, Step 6: Misc initializations: 1e-06 seconds
Process-2148, Step 7: for loop: 0.007548 seconds

Process-2148All steps time: 0.194612 seconds

Process-2148, Find cars processing time: 0.194681 seconds

Process-2147, Step 1: Divide with 255, processing time: 0.010583 seconds
Process-2147, Step 2: Resize if scale is not 1: 0.002228 seconds
Process-2147, Step 3: Get HOG channels: 8e-06 seconds
Process-2147, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2147, Step 5: Compute individual channel HOG features for the entire image: 0.134794 seconds
Process-2147, Step 6: Misc initializations: 1e-06 seconds
Process-2147, Step 7: for loop: 0.00731 seconds

Process-2147All steps time: 0.15493 seconds

Process-2147, Find cars processing time: 0.155018 seconds

Process-2150, Step 1: Divide with 255, processing time: 0.010397 seconds
Process-2150, Step 2: Resize if scale is not 1: 0.001825 seconds
Process-2150, Step 3: Get HOG channels: 8e-06 seconds
Process-2150, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2150, Step 5: Compute individual channel HOG features for the entire image: 0.127414 seconds
Process-2150, Step 6: Misc initializations: 1e-06 seconds
Process-2150, Step 7: for loop: 0.006719 seconds

Process-2150All steps time: 0.146373 seconds

Process-2150, Find cars processing time: 0.146457 seconds

The times for each task are: [0.29141, 0.291193, 0.315709, 0.181385, 0.239405, 0.282556, 0.258571]

Minimum: 0.116947 Maximum: 0.315709 Average: 0.2028 seconds

*****
Number of processes used: 5 window size 260
Length of task list: 22
Number of processes used: 5

Process-2152, Step 1: Divide with 255, processing time: 0.031129 seconds
Process-2152, Step 2: Resize if scale is not 1: 0.002364 seconds
Process-2152, Step 3: Get HOG channels: 1.7e-05 seconds
Process-2152, Step 4: Define blocks and steps as above: 5e-06 seconds
```

Process-2152, Step 5: Compute individual channel HOG features for the entire image: 0.243572 seconds
Process-2152, Step 6: Misc initializations: 1e-06 seconds
Process-2152, Step 7: for loop: 0.016248 seconds

Process-2152All steps time: 0.293336 seconds

Process-2152, Find cars processing time: 0.293499 seconds

Process-2153, Step 1: Divide with 255, processing time: 0.040139 seconds
Process-2153, Step 2: Resize if scale is not 1: 0.003986 seconds
Process-2153, Step 3: Get HOG channels: 1.9e-05 seconds
Process-2153, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2153, Step 5: Compute individual channel HOG features for the entire image: 0.295271 seconds
Process-2153, Step 6: Misc initializations: 1e-06 seconds
Process-2153, Step 7: for loop: 0.016321 seconds

Process-2153All steps time: 0.3557429999999999 seconds

Process-2153, Find cars processing time: 0.355912 seconds

Process-2154, Step 1: Divide with 255, processing time: 0.02554 seconds
Process-2154, Step 2: Resize if scale is not 1: 0.001799 seconds
Process-2154, Step 3: Get HOG channels: 1.6e-05 seconds
Process-2154, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2154, Step 5: Compute individual channel HOG features for the entire image: 0.198156 seconds
Process-2154, Step 6: Misc initializations: 1e-06 seconds
Process-2154, Step 7: for loop: 0.01449 seconds

Process-2154All steps time: 0.240008 seconds

Process-2154, Find cars processing time: 0.240163 seconds

Process-2152, Step 1: Divide with 255, processing time: 0.011096 seconds
Process-2152, Step 2: Resize if scale is not 1: 0.001807 seconds
Process-2152, Step 3: Get HOG channels: 8e-06 seconds
Process-2152, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2152, Step 5: Compute individual channel HOG features for the entire image: 0.249859 seconds
Process-2152, Step 6: Misc initializations: 1e-06 seconds
Process-2152, Step 7: for loop: 0.016622 seconds

Process-2152All steps time: 0.279399 seconds

Process-2152, Find cars processing time: 0.279487 seconds

Process-2156, Step 1: Divide with 255, processing time: 0.029292 seconds
Process-2156, Step 2: Resize if scale is not 1: 0.001995 seconds
Process-2156, Step 3: Get HOG channels: 2e-05 seconds
Process-2156, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2156, Step 5: Compute individual channel HOG features for the entire image: 0.201102 seconds
Process-2156, Step 6: Misc initializations: 1e-06 seconds
Process-2156, Step 7: for loop: 0.013145 seconds

Process-2156All steps time: 0.245562 seconds

Process-2156, Find cars processing time: 0.245735 seconds

Process-2152, Step 1: Divide with 255, processing time: 0.010645 seconds
Process-2152, Step 2: Resize if scale is not 1: 0.001057 seconds
Process-2152, Step 3: Get HOG channels: 1e-05 seconds
Process-2152, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2152, Step 5: Compute individual channel HOG features for the entire image: 0.181049 seconds
Process-2152, Step 6: Misc initializations: 1e-06 seconds
Process-2152, Step 7: for loop: 0.009968 seconds

Process-2152All steps time: 0.202739 seconds

Process-2152, Find cars processing time: 0.202836 seconds

Process-2155, Step 1: Divide with 255, processing time: 0.02535 seconds
Process-2155, Step 2: Resize if scale is not 1: 0.002009 seconds
Process-2155, Step 3: Get HOG channels: 2.3e-05 seconds
Process-2155, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2155, Step 5: Compute individual channel HOG features for the entire image: 0.169286 seconds
Process-2155, Step 6: Misc initializations: 0.0 seconds
Process-2155, Step 7: for loop: 0.010941 seconds

Process-2155All steps time: 0.207615 seconds

Process-2155, Find cars processing time: 0.207759 seconds

Process-2153, Step 1: Divide with 255, processing time: 0.010683 seconds
Process-2153, Step 2: Resize if scale is not 1: 0.000589 seconds
Process-2153, Step 3: Get HOG channels: 6e-06 seconds
Process-2153, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2153, Step 5: Compute individual channel HOG features for the entire image: 0.203311 seconds
Process-2153, Step 6: Misc initializations: 1e-06 seconds
Process-2153, Step 7: for loop: 0.010759 seconds

Process-2153All steps time: 0.22535399999999997 seconds

Process-2153, Find cars processing time: 0.225407 seconds

Process-2154, Step 1: Divide with 255, processing time: 0.007237 seconds
Process-2154, Step 2: Resize if scale is not 1: 0.000687 seconds
Process-2154, Step 3: Get HOG channels: 7e-06 seconds
Process-2154, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2154, Step 5: Compute individual channel HOG features for the entire image: 0.207614 seconds
Process-2154, Step 6: Misc initializations: 1e-06 seconds
Process-2154, Step 7: for loop: 0.01241 seconds

Process-2154All steps time: 0.227961 seconds

Process-2154, Find cars processing time: 0.228027 seconds

Process-2156, Step 1: Divide with 255, processing time: 0.010431 seconds
Process-2156, Step 2: Resize if scale is not 1: 0.000908 seconds
Process-2156, Step 3: Get HOG channels: 8e-06 seconds
Process-2156, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2156, Step 5: Compute individual channel HOG features for the entire image: 0.164424 seconds
Process-2156, Step 6: Misc initializations: 1e-06 seconds
Process-2156, Step 7: for loop: 0.008122 seconds

Process-2156All steps time: 0.1839009999999998 seconds

Process-2156, Find cars processing time: 0.183984 seconds

Process-2155, Step 1: Divide with 255, processing time: 0.009459 seconds
Process-2155, Step 2: Resize if scale is not 1: 0.000799 seconds
Process-2155, Step 3: Get HOG channels: 8e-06 seconds
Process-2155, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2155, Step 5: Compute individual channel HOG features for the entire image: 0.198359 seconds
Process-2155, Step 6: Misc initializations: 1e-06 seconds
Process-2155, Step 7: for loop: 0.010072 seconds

Process-2155All steps time: 0.218705 seconds

Process-2155, Find cars processing time: 0.218781 seconds

Process-2153, Step 1: Divide with 255, processing time: 0.010204 seconds
Process-2153, Step 2: Resize if scale is not 1: 0.000847 seconds
Process-2153, Step 3: Get HOG channels: 7e-06 seconds
Process-2153, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2153, Step 5: Compute individual channel HOG features for the entire image: 0.122765 seconds
Process-2153, Step 6: Misc initializations: 1e-06 seconds
Process-2153, Step 7: for loop: 0.006552 seconds

Process-2153All steps time: 0.140383 seconds

Process-2153, Find cars processing time: 0.140447 seconds

Process-2154, Step 1: Divide with 255, processing time: 0.008968 seconds
Process-2154, Step 2: Resize if scale is not 1: 0.000632 seconds
Process-2154, Step 3: Get HOG channels: 6e-06 seconds
Process-2154, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2154, Step 5: Compute individual channel HOG features for the entire image: 0.122919 seconds
Process-2154, Step 6: Misc initializations: 1e-06 seconds
Process-2154, Step 7: for loop: 0.00612 seconds

Process-2154All steps time: 0.138652 seconds

Process-2154, Find cars processing time: 0.138707 seconds

Process-2152, Step 1: Divide with 255, processing time: 0.008187 seconds
Process-2152, Step 2: Resize if scale is not 1: 0.000585 seconds
Process-2152, Step 3: Get HOG channels: 7e-06 seconds
Process-2152, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2152, Step 5: Compute individual channel HOG features for the entire image: 0.185169 seconds
Process-2152, Step 6: Misc initializations: 1e-06 seconds
Process-2152, Step 7: for loop: 0.010015 seconds

Process-2152All steps time: 0.20397 seconds

Process-2152, Find cars processing time: 0.204041 seconds

Process-2154, Step 1: Divide with 255, processing time: 0.006418 seconds
Process-2154, Step 2: Resize if scale is not 1: 0.000809 seconds
Process-2154, Step 3: Get HOG channels: 5e-06 seconds
Process-2154, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2154, Step 5: Compute individual channel HOG features for the entire image: 0.112842 seconds
Process-2154, Step 6: Misc initializations: 0.0 seconds
Process-2154, Step 7: for loop: 0.006098 seconds

Process-2154All steps time: 0.1261779999999998 seconds

Process-2154, Find cars processing time: 0.126229 seconds

Process-2156, Step 1: Divide with 255, processing time: 0.010913 seconds
Process-2156, Step 2: Resize if scale is not 1: 0.000893 seconds
Process-2156, Step 3: Get HOG channels: 7e-06 seconds
Process-2156, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2156, Step 5: Compute individual channel HOG features for the entire image: 0.119588 seconds
Process-2156, Step 6: Misc initializations: 0.0 seconds
Process-2156, Step 7: for loop: 0.005895 seconds

Process-2156All steps time: 0.137304 seconds

Process-2156, Find cars processing time: 0.137371 seconds

Process-2155, Step 1: Divide with 255, processing time: 0.010778 seconds
Process-2155, Step 2: Resize if scale is not 1: 0.000813 seconds
Process-2155, Step 3: Get HOG channels: 7e-06 seconds
Process-2155, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2155, Step 5: Compute individual channel HOG features for the entire image: 0.141998 seconds
Process-2155, Step 6: Misc initializations: 1e-06 seconds
Process-2155, Step 7: for loop: 0.007276 seconds

Process-2155All steps time: 0.16088000000000002 seconds

Process-2155, Find cars processing time: 0.160963 seconds

Process-2152, Step 1: Divide with 255, processing time: 0.006665 seconds
Process-2152, Step 2: Resize if scale is not 1: 0.000813 seconds
Process-2152, Step 3: Get HOG channels: 7e-06 seconds
Process-2152, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2152, Step 5: Compute individual channel HOG features for the entire image: 0.130157 seconds
Process-2152, Step 6: Misc initializations: 1e-06 seconds
Process-2152, Step 7: for loop: 0.007425 seconds

Process-2152All steps time: 0.14507299999999998 seconds

Process-2152, Find cars processing time: 0.145136 seconds

Process-2154, Step 1: Divide with 255, processing time: 0.006563 seconds
Process-2154, Step 2: Resize if scale is not 1: 0.000517 seconds
Process-2154, Step 3: Get HOG channels: 5e-06 seconds
Process-2154, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2154, Step 5: Compute individual channel HOG features for the entire image: 0.156571 seconds
Process-2154, Step 6: Misc initializations: 1e-06 seconds
Process-2154, Step 7: for loop: 0.007571 seconds

Process-2154All steps time: 0.171233 seconds

Process-2154, Find cars processing time: 0.171307 seconds

Process-2153, Step 1: Divide with 255, processing time: 0.006121 seconds
Process-2153, Step 2: Resize if scale is not 1: 0.000662 seconds
Process-2153, Step 3: Get HOG channels: 5e-06 seconds
Process-2153, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2153, Step 5: Compute individual channel HOG features for the entire image: 0.097687 seconds
Process-2153, Step 6: Misc initializations: 1e-06 seconds
Process-2153, Step 7: for loop: 0.004392 seconds

Process-2153All steps time: 0.108873 seconds

Process-2153, Find cars processing time: 0.108924 seconds

Process-2155, Step 1: Divide with 255, processing time: 0.010702 seconds
Process-2155, Step 2: Resize if scale is not 1: 0.001866 seconds
Process-2155, Step 3: Get HOG channels: 7e-06 seconds
Process-2155, Step 4: Define blocks and steps as above: 8e-06 seconds

```
Process-2155, Step 5: Compute individual channel HOG features for the entire image: 0.089646 seconds
Process-2155, Step 6: Misc initializations: 1e-06 seconds
Process-2155, Step 7: for loop: 0.005118 seconds

Process-2155All steps time: 0.107348 seconds

Process-2155, Find cars processing time: 0.107411 seconds

Process-2156, Step 1: Divide with 255, processing time: 0.010566 seconds
Process-2156, Step 2: Resize if scale is not 1: 0.001225 seconds
Process-2156, Step 3: Get HOG channels: 8e-06 seconds
Process-2156, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-2156, Step 5: Compute individual channel HOG features for the entire image: 0.13588 seconds
Process-2156, Step 6: Misc initializations: 1e-06 seconds
Process-2156, Step 7: for loop: 0.007116 seconds

Process-2156All steps time: 0.1548030000000002 seconds

Process-2156, Find cars processing time: 0.154875 seconds

The times for each task are: [0.293499, 0.355912, 0.240163, 0.279487, 0.245735, 0.202836, 0.2077

Minimum: 0.107411 Maximum: 0.355912 Average: 0.1944 seconds

*****
Number of processes used: 5 window size 260
Length of task list: 22
Number of processes used: 5

Process-2158, Step 1: Divide with 255, processing time: 0.026339 seconds
Process-2157, Step 1: Divide with 255, processing time: 0.027807 seconds
Process-2158, Step 2: Resize if scale is not 1: 0.001981 seconds
Process-2157, Step 2: Resize if scale is not 1: 0.001778 seconds
Process-2157, Step 3: Get HOG channels: 1.7e-05 seconds
Process-2158, Step 3: Get HOG channels: 1.7e-05 seconds
Process-2157, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2158, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2159, Step 1: Divide with 255, processing time: 0.024734 seconds
Process-2157, Step 5: Compute individual channel HOG features for the entire image: 0.249084 seconds
Process-2158, Step 5: Compute individual channel HOG features for the entire image: 0.246916 seconds
Process-2157, Step 6: Misc initializations: 1e-06 seconds
Process-2159, Step 2: Resize if scale is not 1: 0.001754 seconds
Process-2158, Step 6: Misc initializations: 1e-06 seconds
Process-2157, Step 7: for loop: 0.017671 seconds
Process-2159, Step 3: Get HOG channels: 1.8e-05 seconds
Process-2158, Step 7: for loop: 0.016504 seconds
```

Process-2159, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2157All steps time: 0.2963619999999996 seconds
Process-2159, Step 5: Compute individual channel HOG features for the entire image: 0.234615 seconds
Process-2158All steps time: 0.2917639999999997 seconds

Process-2159, Step 6: Misc initializations: 1e-06 seconds

Process-2157, Find cars processing time: 0.296508 seconds
Process-2159, Step 7: for loop: 0.017505 seconds
Process-2158, Find cars processing time: 0.291933 seconds

Process-2159All steps time: 0.2786319999999994 seconds

Process-2159, Find cars processing time: 0.278795 seconds

Process-2160, Step 1: Divide with 255, processing time: 0.028235 seconds
Process-2160, Step 2: Resize if scale is not 1: 0.001984 seconds
Process-2160, Step 3: Get HOG channels: 1.8e-05 seconds
Process-2160, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2160, Step 5: Compute individual channel HOG features for the entire image: 0.258385 seconds
Process-2160, Step 6: Misc initializations: 1e-06 seconds
Process-2160, Step 7: for loop: 0.018716 seconds

Process-2160All steps time: 0.307345 seconds

Process-2160, Find cars processing time: 0.307503 seconds

Process-2161, Step 1: Divide with 255, processing time: 0.02899 seconds
Process-2161, Step 2: Resize if scale is not 1: 0.002158 seconds
Process-2161, Step 3: Get HOG channels: 2.2e-05 seconds
Process-2161, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2161, Step 5: Compute individual channel HOG features for the entire image: 0.206402 seconds
Process-2161, Step 6: Misc initializations: 1e-06 seconds
Process-2161, Step 7: for loop: 0.013516 seconds

Process-2161All steps time: 0.251095 seconds

Process-2161, Find cars processing time: 0.251247 seconds

Process-2158, Step 1: Divide with 255, processing time: 0.01088 seconds
Process-2158, Step 2: Resize if scale is not 1: 0.000576 seconds
Process-2158, Step 3: Get HOG channels: 5e-06 seconds
Process-2158, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-2158, Step 5: Compute individual channel HOG features for the entire image: 0.206049 seconds
Process-2158, Step 6: Misc initializations: 1e-06 seconds
Process-2158, Step 7: for loop: 0.012863 seconds

Process-2158All steps time: 0.230379 seconds

Process-2158, Find cars processing time: 0.230449 seconds

Process-2159, Step 1: Divide with 255, processing time: 0.010783 seconds

Process-2159, Step 2: Resize if scale is not 1: 0.000884 seconds

Process-2159, Step 3: Get HOG channels: 9e-06 seconds

Process-2159, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2159, Step 5: Compute individual channel HOG features for the entire image: 0.221389 seconds

Process-2159, Step 6: Misc initializations: 0.0 seconds

Process-2159, Step 7: for loop: 0.011263 seconds

Process-2159All steps time: 0.244335 seconds

Process-2159, Find cars processing time: 0.244408 seconds

Process-2157, Step 1: Divide with 255, processing time: 0.010631 seconds

Process-2157, Step 2: Resize if scale is not 1: 0.000877 seconds

Process-2157, Step 3: Get HOG channels: 8e-06 seconds

Process-2157, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2157, Step 5: Compute individual channel HOG features for the entire image: 0.207137 seconds

Process-2157, Step 6: Misc initializations: 1e-06 seconds

Process-2157, Step 7: for loop: 0.011041 seconds

Process-2157All steps time: 0.229702 seconds

Process-2157, Find cars processing time: 0.229769 seconds

Process-2158, Step 1: Divide with 255, processing time: 0.009829 seconds

Process-2158, Step 2: Resize if scale is not 1: 0.000832 seconds

Process-2158, Step 3: Get HOG channels: 7e-06 seconds

Process-2158, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2158, Step 5: Compute individual channel HOG features for the entire image: 0.18573 seconds

Process-2158, Step 6: Misc initializations: 1e-06 seconds

Process-2158, Step 7: for loop: 0.009916 seconds

Process-2158All steps time: 0.206322 seconds

Process-2158, Find cars processing time: 0.206387 seconds

Process-2160, Step 1: Divide with 255, processing time: 0.010684 seconds

Process-2160, Step 2: Resize if scale is not 1: 0.000974 seconds

Process-2160, Step 3: Get HOG channels: 8e-06 seconds

Process-2161, Step 1: Divide with 255, processing time: 0.011279 seconds

Process-2160, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-2161, Step 2: Resize if scale is not 1: 0.000942 seconds

Process-2160, Step 5: Compute individual channel HOG features for the entire image: 0.171472 seconds

Process-2161, Step 3: Get HOG channels: 8e-06 seconds
Process-2160, Step 6: Misc initializations: 0.0 seconds
Process-2161, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2160, Step 7: for loop: 0.009334 seconds
Process-2161, Step 5: Compute individual channel HOG features for the entire image: 0.168079 seconds

Process-2161, Step 6: Misc initializations: 0.0 seconds
Process-2160All steps time: 0.192481 seconds
Process-2161, Step 7: for loop: 0.009406 seconds

Process-2160, Find cars processing time: 0.192553 seconds
Process-2161All steps time: 0.189722 seconds

Process-2161, Find cars processing time: 0.189785 seconds

Process-2159, Step 1: Divide with 255, processing time: 0.007243 seconds
Process-2159, Step 2: Resize if scale is not 1: 0.00081 seconds
Process-2159, Step 3: Get HOG channels: 8e-06 seconds
Process-2159, Step 4: Define blocks and steps as above: 2.3e-05 seconds
Process-2159, Step 5: Compute individual channel HOG features for the entire image: 0.172983 seconds
Process-2159, Step 6: Misc initializations: 1e-06 seconds
Process-2159, Step 7: for loop: 0.008128 seconds

Process-2159All steps time: 0.189196 seconds

Process-2159, Find cars processing time: 0.189277 seconds

Process-2157, Step 1: Divide with 255, processing time: 0.010742 seconds
Process-2157, Step 2: Resize if scale is not 1: 0.000547 seconds
Process-2157, Step 3: Get HOG channels: 5e-06 seconds
Process-2157, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2157, Step 5: Compute individual channel HOG features for the entire image: 0.112193 seconds
Process-2157, Step 6: Misc initializations: 1e-06 seconds
Process-2157, Step 7: for loop: 0.024883 seconds

Process-2157All steps time: 0.148375 seconds

Process-2157, Find cars processing time: 0.14845 seconds

Process-2158, Step 1: Divide with 255, processing time: 0.010432 seconds
Process-2158, Step 2: Resize if scale is not 1: 0.000797 seconds
Process-2158, Step 3: Get HOG channels: 1e-05 seconds
Process-2158, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2158, Step 5: Compute individual channel HOG features for the entire image: 0.142987 seconds
Process-2158, Step 6: Misc initializations: 1e-06 seconds
Process-2158, Step 7: for loop: 0.00726 seconds

Process-2158All steps time: 0.161496 seconds

Process-2158, Find cars processing time: 0.161563 seconds

Process-2157, Step 1: Divide with 255, processing time: 0.007826 seconds

Process-2157, Step 2: Resize if scale is not 1: 0.000918 seconds

Process-2157, Step 3: Get HOG channels: 7e-06 seconds

Process-2157, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2157, Step 5: Compute individual channel HOG features for the entire image: 0.106394 seconds

Process-2157, Step 6: Misc initializations: 0.0 seconds

Process-2157, Step 7: for loop: 0.00634 seconds

Process-2157All steps time: 0.12149 seconds

Process-2157, Find cars processing time: 0.121544 seconds

Process-2161, Step 1: Divide with 255, processing time: 0.010862 seconds

Process-2161, Step 2: Resize if scale is not 1: 0.000558 seconds

Process-2161, Step 3: Get HOG channels: 5e-06 seconds

Process-2161, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2161, Step 5: Compute individual channel HOG features for the entire image: 0.149282 seconds

Process-2161, Step 6: Misc initializations: 1e-06 seconds

Process-2161, Step 7: for loop: 0.007254 seconds

Process-2161All steps time: 0.167967 seconds

Process-2161, Find cars processing time: 0.168035 seconds

Process-2160, Step 1: Divide with 255, processing time: 0.009823 seconds

Process-2160, Step 2: Resize if scale is not 1: 0.000821 seconds

Process-2160, Step 3: Get HOG channels: 7e-06 seconds

Process-2160, Step 4: Define blocks and steps as above: 1.5e-05 seconds

Process-2160, Step 5: Compute individual channel HOG features for the entire image: 0.144084 seconds

Process-2160, Step 6: Misc initializations: 1e-06 seconds

Process-2160, Step 7: for loop: 0.007396 seconds

Process-2160All steps time: 0.1621469999999999 seconds

Process-2160, Find cars processing time: 0.16222 seconds

Process-2159, Step 1: Divide with 255, processing time: 0.010669 seconds

Process-2159, Step 2: Resize if scale is not 1: 0.000693 seconds

Process-2159, Step 3: Get HOG channels: 8e-06 seconds

Process-2159, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2159, Step 5: Compute individual channel HOG features for the entire image: 0.109437 seconds

Process-2158, Step 1: Divide with 255, processing time: 0.009945 seconds

Process-2159, Step 6: Misc initializations: 1e-06 seconds

Process-2158, Step 2: Resize if scale is not 1: 0.000819 seconds
Process-2159, Step 7: for loop: 0.00493 seconds
Process-2158, Step 3: Get HOG channels: 6e-06 seconds

Process-2158, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2159All steps time: 0.125745 seconds

Process-2158, Step 5: Compute individual channel HOG features for the entire image: 0.139261 seconds

Process-2158, Step 6: Misc initializations: 1e-06 seconds
Process-2158, Step 7: for loop: 0.007412 seconds
Process-2159, Find cars processing time: 0.125816 seconds
Process-2161, Step 1: Divide with 255, processing time: 0.007743 seconds

Process-2161, Step 2: Resize if scale is not 1: 0.000862 seconds
Process-2161, Step 3: Get HOG channels: 5e-06 seconds
Process-2158All steps time: 0.15745 seconds
Process-2161, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2161, Step 5: Compute individual channel HOG features for the entire image: 0.129318 seconds
Process-2158, Find cars processing time: 0.157514 seconds
Process-2161, Step 6: Misc initializations: 1e-06 seconds

Process-2161, Step 7: for loop: 0.007545 seconds

Process-2161All steps time: 0.145479 seconds

Process-2161, Find cars processing time: 0.145524 seconds

Process-2157, Step 1: Divide with 255, processing time: 0.009926 seconds
Process-2157, Step 2: Resize if scale is not 1: 0.000437 seconds
Process-2157, Step 3: Get HOG channels: 4e-06 seconds
Process-2157, Step 4: Define blocks and steps as above: 4e-06 seconds
Process-2157, Step 5: Compute individual channel HOG features for the entire image: 0.103115 seconds
Process-2157, Step 6: Misc initializations: 1e-06 seconds
Process-2157, Step 7: for loop: 0.00598 seconds

Process-2157All steps time: 0.119467 seconds

Process-2157, Find cars processing time: 0.119515 seconds

Process-2160, Step 1: Divide with 255, processing time: 0.010208 seconds
Process-2160, Step 2: Resize if scale is not 1: 0.002094 seconds
Process-2160, Step 3: Get HOG channels: 1.3e-05 seconds
Process-2160, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-2160, Step 5: Compute individual channel HOG features for the entire image: 0.088265 seconds
Process-2160, Step 6: Misc initializations: 0.0 seconds
Process-2160, Step 7: for loop: 0.00477 seconds

Process-2160All steps time: 0.105359 seconds

Process-2160, Find cars processing time: 0.105443 seconds

The times for each task are: [0.296508, 0.291933, 0.278795, 0.307503, 0.251247, 0.230449, 0.2444

Minimum: 0.105443 Maximum: 0.307503 Average: 0.1966 seconds

Number of processes used: 5 window size 260

Length of task list: 22

Number of processes used: 5

Process-2162, Step 1: Divide with 255, processing time: 0.027123 seconds

Process-2163, Step 1: Divide with 255, processing time: 0.024704 seconds

Process-2162, Step 2: Resize if scale is not 1: 0.002009 seconds

Process-2163, Step 2: Resize if scale is not 1: 0.001705 seconds

Process-2162, Step 3: Get HOG channels: 1.8e-05 seconds

Process-2163, Step 3: Get HOG channels: 1.7e-05 seconds

Process-2162, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2163, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2162, Step 5: Compute individual channel HOG features for the entire image: 0.231298 sec

Process-2163, Step 5: Compute individual channel HOG features for the entire image: 0.196947 sec

Process-2162, Step 6: Misc initializations: 2e-06 seconds

Process-2163, Step 6: Misc initializations: 1e-06 seconds

Process-2162, Step 7: for loop: 0.014732 seconds

Process-2163, Step 7: for loop: 0.014495 seconds

Process-2162All steps time: 0.2751880000000004 seconds

Process-2163All steps time: 0.2378740000000003 seconds

Process-2162, Find cars processing time: 0.275347 seconds

Process-2163, Find cars processing time: 0.238029 seconds

Process-2164, Step 1: Divide with 255, processing time: 0.026356 seconds

Process-2164, Step 2: Resize if scale is not 1: 0.00174 seconds

Process-2164, Step 3: Get HOG channels: 1.8e-05 seconds

Process-2164, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2164, Step 5: Compute individual channel HOG features for the entire image: 0.245912 sec

Process-2164, Step 6: Misc initializations: 1e-06 seconds

Process-2164, Step 7: for loop: 0.017621 seconds

Process-2164All steps time: 0.2916539999999997 seconds

Process-2164, Find cars processing time: 0.291784 seconds

Process-2165, Step 1: Divide with 255, processing time: 0.027714 seconds

Process-2165, Step 2: Resize if scale is not 1: 0.002036 seconds

Process-2165, Step 3: Get HOG channels: 1.8e-05 seconds

Process-2165, Step 4: Define blocks and steps as above: 6e-06 seconds

Process-2165, Step 5: Compute individual channel HOG features for the entire image: 0.236914 seconds

Process-2165, Step 6: Misc initializations: 1e-06 seconds

Process-2165, Step 7: for loop: 0.02053 seconds

Process-2165All steps time: 0.287219 seconds

Process-2165, Find cars processing time: 0.287367 seconds

Process-2163, Step 1: Divide with 255, processing time: 0.011069 seconds

Process-2163, Step 2: Resize if scale is not 1: 0.000901 seconds

Process-2163, Step 3: Get HOG channels: 9e-06 seconds

Process-2163, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2163, Step 5: Compute individual channel HOG features for the entire image: 0.189665 seconds

Process-2163, Step 6: Misc initializations: 1e-06 seconds

Process-2162, Step 1: Divide with 255, processing time: 0.009699 seconds

Process-2163, Step 7: for loop: 0.01045 seconds

Process-2162, Step 2: Resize if scale is not 1: 0.000565 seconds

Process-2162, Step 3: Get HOG channels: 5e-06 seconds

Process-2163All steps time: 0.2121029999999999 seconds

Process-2162, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2162, Step 5: Compute individual channel HOG features for the entire image: 0.217608 seconds

Process-2163, Find cars processing time: 0.212172 seconds

Process-2162, Step 6: Misc initializations: 1e-06 seconds

Process-2162, Step 7: for loop: 0.013094 seconds

Process-2162All steps time: 0.240977 seconds

Process-2162, Find cars processing time: 0.241049 seconds

Process-2164, Step 1: Divide with 255, processing time: 0.011111 seconds

Process-2166, Step 1: Divide with 255, processing time: 0.025844 seconds

Process-2164, Step 2: Resize if scale is not 1: 0.000912 seconds

Process-2166, Step 2: Resize if scale is not 1: 0.001754 seconds

Process-2164, Step 3: Get HOG channels: 8e-06 seconds

Process-2166, Step 3: Get HOG channels: 2.1e-05 seconds

Process-2164, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2166, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2164, Step 5: Compute individual channel HOG features for the entire image: 0.216839 seconds

Process-2166, Step 5: Compute individual channel HOG features for the entire image: 0.196532 seconds

Process-2164, Step 6: Misc initializations: 1e-06 seconds
Process-2166, Step 6: Misc initializations: 1e-06 seconds
Process-2164, Step 7: for loop: 0.012635 seconds
Process-2166, Step 7: for loop: 0.010948 seconds

Process-2164All steps time: 0.241514 seconds
Process-2166All steps time: 0.235105 seconds

Process-2164, Find cars processing time: 0.241602 seconds
Process-2166, Find cars processing time: 0.235234 seconds

Process-2165, Step 1: Divide with 255, processing time: 0.009593 seconds
Process-2165, Step 2: Resize if scale is not 1: 0.000893 seconds
Process-2165, Step 3: Get HOG channels: 8e-06 seconds
Process-2165, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2165, Step 5: Compute individual channel HOG features for the entire image: 0.19974 seconds
Process-2165, Step 6: Misc initializations: 0.0 seconds
Process-2165, Step 7: for loop: 0.010078 seconds

Process-2165All steps time: 0.220318 seconds

Process-2165, Find cars processing time: 0.220392 seconds

Process-2163, Step 1: Divide with 255, processing time: 0.009196 seconds
Process-2163, Step 2: Resize if scale is not 1: 0.0008 seconds
Process-2163, Step 3: Get HOG channels: 9e-06 seconds
Process-2163, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-2163, Step 5: Compute individual channel HOG features for the entire image: 0.14103 seconds
Process-2163, Step 6: Misc initializations: 1e-06 seconds
Process-2163, Step 7: for loop: 0.008013 seconds

Process-2163All steps time: 0.1590569999999998 seconds

Process-2163, Find cars processing time: 0.159121 seconds

Process-2165, Step 1: Divide with 255, processing time: 0.008277 seconds
Process-2165, Step 2: Resize if scale is not 1: 0.000746 seconds
Process-2165, Step 3: Get HOG channels: 7e-06 seconds
Process-2165, Step 4: Define blocks and steps as above: 6e-06 seconds
Process-2165, Step 5: Compute individual channel HOG features for the entire image: 0.168026 seconds
Process-2165, Step 6: Misc initializations: 1e-06 seconds
Process-2165, Step 7: for loop: 0.006323 seconds

Process-2165All steps time: 0.183386 seconds

Process-2165, Find cars processing time: 0.18347 seconds

Process-2162, Step 1: Divide with 255, processing time: 0.010492 seconds

Process-2162, Step 2: Resize if scale is not 1: 0.000601 seconds

Process-2162, Step 3: Get HOG channels: 6e-06 seconds

Process-2162, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2162, Step 5: Compute individual channel HOG features for the entire image: 0.185502 seconds

Process-2162, Step 6: Misc initializations: 1e-06 seconds

Process-2162, Step 7: for loop: 0.010122 seconds

Process-2162All steps time: 0.206729 seconds

Process-2162, Find cars processing time: 0.206791 seconds

Process-2163, Step 1: Divide with 255, processing time: 0.009486 seconds

Process-2163, Step 2: Resize if scale is not 1: 0.000756 seconds

Process-2163, Step 3: Get HOG channels: 7e-06 seconds

Process-2163, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2163, Step 5: Compute individual channel HOG features for the entire image: 0.131787 seconds

Process-2163, Step 6: Misc initializations: 0.0 seconds

Process-2163, Step 7: for loop: 0.005703 seconds

Process-2163All steps time: 0.147746 seconds

Process-2163, Find cars processing time: 0.14781 seconds

Process-2165, Step 1: Divide with 255, processing time: 0.009259 seconds

Process-2165, Step 2: Resize if scale is not 1: 0.000756 seconds

Process-2165, Step 3: Get HOG channels: 8e-06 seconds

Process-2165, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2165, Step 5: Compute individual channel HOG features for the entire image: 0.134908 seconds

Process-2165, Step 6: Misc initializations: 1e-06 seconds

Process-2165, Step 7: for loop: 0.00661 seconds

Process-2165All steps time: 0.15155 seconds

Process-2165, Find cars processing time: 0.151618 seconds

Process-2166, Step 1: Divide with 255, processing time: 0.01077 seconds

Process-2166, Step 2: Resize if scale is not 1: 0.000668 seconds

Process-2166, Step 3: Get HOG channels: 5e-06 seconds

Process-2166, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2163, Step 1: Divide with 255, processing time: 0.009985 seconds

Process-2166, Step 5: Compute individual channel HOG features for the entire image: 0.12017 seconds

Process-2163, Step 2: Resize if scale is not 1: 0.001189 seconds

Process-2166, Step 6: Misc initializations: 0.0 seconds

Process-2163, Step 3: Get HOG channels: 9e-06 seconds

Process-2166, Step 7: for loop: 0.006089 seconds

Process-2163, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2163, Step 5: Compute individual channel HOG features for the entire image: 0.108136 seconds

Process-2166All steps time: 0.13770700000000002 seconds

Process-2163, Step 6: Misc initializations: 1e-06 seconds

Process-2163, Step 7: for loop: 0.006545 seconds

Process-2166, Find cars processing time: 0.137767 seconds

Process-2162, Step 1: Divide with 255, processing time: 0.010676 seconds

Process-2162, Step 2: Resize if scale is not 1: 0.000645 seconds

Process-2163All steps time: 0.125872 seconds

Process-2162, Step 3: Get HOG channels: 7e-06 seconds

Process-2162, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2163, Find cars processing time: 0.125946 seconds

Process-2162, Step 5: Compute individual channel HOG features for the entire image: 0.114614 seconds

Process-2162, Step 6: Misc initializations: 1e-06 seconds

Process-2162All steps time: 0.131812 seconds

Process-2162, Step 7: for loop: 0.005862 seconds

Process-2162, Find cars processing time: 0.131882 seconds

Process-2164, Step 1: Divide with 255, processing time: 0.010546 seconds

Process-2164, Step 2: Resize if scale is not 1: 0.000897 seconds

Process-2164, Step 3: Get HOG channels: 8e-06 seconds

Process-2164, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-2164, Step 5: Compute individual channel HOG features for the entire image: 0.177023 seconds

Process-2164, Step 6: Misc initializations: 1e-06 seconds

Process-2164, Step 7: for loop: 0.008264 seconds

Process-2164All steps time: 0.196746 seconds

Process-2164, Find cars processing time: 0.196819 seconds

Process-2165, Step 1: Divide with 255, processing time: 0.009531 seconds

Process-2165, Step 2: Resize if scale is not 1: 0.000895 seconds

Process-2165, Step 3: Get HOG channels: 6e-06 seconds

Process-2165, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2165, Step 5: Compute individual channel HOG features for the entire image: 0.144686 seconds

Process-2165, Step 6: Misc initializations: 1e-06 seconds

Process-2165, Step 7: for loop: 0.007118 seconds

Process-2165All steps time: 0.16224200000000003 seconds

Process-2165, Find cars processing time: 0.162324 seconds

Process-2166, Step 1: Divide with 255, processing time: 0.006481 seconds

Process-2166, Step 2: Resize if scale is not 1: 0.000723 seconds

Process-2166, Step 3: Get HOG channels: 5e-06 seconds

Process-2166, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-2166, Step 5: Compute individual channel HOG features for the entire image: 0.111014 seconds

Process-2166, Step 6: Misc initializations: 1e-06 seconds

Process-2166, Step 7: for loop: 0.006125 seconds

Process-2166All steps time: 0.124354 seconds

Process-2163, Step 1: Divide with 255, processing time: 0.011689 seconds

Process-2163, Step 2: Resize if scale is not 1: 0.000746 seconds

Process-2163, Step 3: Get HOG channels: 1e-05 seconds

Process-2166, Find cars processing time: 0.124431 seconds

Process-2163, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-2163, Step 5: Compute individual channel HOG features for the entire image: 0.144244 seconds

Process-2163, Step 6: Misc initializations: 1e-06 seconds

Process-2163, Step 7: for loop: 0.007738 seconds

Process-2163All steps time: 0.164438 seconds

Process-2163, Find cars processing time: 0.16454 seconds

Process-2162, Step 1: Divide with 255, processing time: 0.010993 seconds

Process-2162, Step 2: Resize if scale is not 1: 0.001914 seconds

Process-2162, Step 3: Get HOG channels: 9e-06 seconds

Process-2162, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-2162, Step 5: Compute individual channel HOG features for the entire image: 0.105378 seconds

Process-2162, Step 6: Misc initializations: 1e-06 seconds

Process-2162, Step 7: for loop: 0.005223 seconds

Process-2162All steps time: 0.123526 seconds

Process-2162, Find cars processing time: 0.123599 seconds

The times for each task are: [0.275347, 0.238029, 0.291784, 0.287367, 0.212172, 0.241049, 0.241611]

Minimum: 0.123599 Maximum: 0.291784 Average: 0.1936 seconds

Window sizes used: [1.15, 1.25, 1.35, 1.45, 1.55]

5 processes used for testing 5 window sizes

Processing times for each image [3.6875, 3.5488, 4.0575, 3.5108, 3.5504, 3.4376] with an average

Time elapsed so far... 447.06270000000006

```
#####
=====
In [120]: use_sub_windows = 1
           use_default_number_of_processes = 0
           print_find_car_step_times = 0
           find_car_hog_parallelism_enabled = 0
           xwinstarts, xwinstops = calculate_patches(x_start, x_stop, window, scale, step, cells_
           ywinstarts, ywinstops = calculate_patches(y_start, y_stop, window, scale, step, cells_
           step = 260
           print_task_times = 1

           time_elapsed = 0

           for i in range(1,2):
               totalTasks = len(scaleslist[i]) * len(xwinstarts) * len(ywinstarts)
               print('Number of Tasks for scale and step', totalTasks)
               for j in range(1, int(totalTasks/2)+2):
                   time_for_all_images = []
                   test_images_with_boxes = []
                   number_of_processes_used = j
                   print('Number of processes used: ', number_of_processes_used, 'window size', s
                   print()
                   for image in test_images:
                       img = mpimg.imread(image)
                       freeze_support()
                       t0 = time.time()
                       r = []
                       r = parallelize_find_cars(img, number_of_processes_used, i)
                       t1 = time.time()
                       processing_time = round(t1-t0, 4)
                       print('Time taken for processing each image', processing_time)
                       time_for_all_images.append(processing_time)
                       img_with_boxes = np.copy(img)
                       for box in r:
                           cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
                       #plt.figure(figsize=(16,9))
                       #plt.imshow(img_with_boxes)
                       test_images_with_boxes.append(img_with_boxes)
                   print('*****')
                   print('Window sizes used:', scaleslist[i])
                   print(number_of_processes_used, 'processes used for testing', len(scaleslist[i])
                   print('Processing times for each image', time_for_all_images, \
                         'with an average of', round(sum(time_for_all_images)/len(time_for_all_im
```

```

    print()
    time_elapsed += sum(time_for_all_images)
    print('Time elapsed so far...', time_elapsed)
    print('#####')
    print('#####')

    print()
    print()
    print('=====')
    print()
    print()

Number of Tasks for scale and step 12
Number of processes used: 1 window size 260

Length of task list: 13
Number of processes used: 1

The times for each task are: [0.275101, 0.208479, 0.208347, 0.210472, 0.150878, 0.145408, 0.1581

Minimum: 0.111073 Maximum: 0.275101 Average: 0.165 seconds

Time taken for processing each image 6.0213
*****
Length of task list: 13
Number of processes used: 1

The times for each task are: [0.264052, 0.199001, 0.199136, 0.207918, 0.147452, 0.169336, 0.1407

Minimum: 0.111869 Maximum: 0.264052 Average: 0.16 seconds

Time taken for processing each image 5.2771
*****
Length of task list: 13
Number of processes used: 1

The times for each task are: [0.302624, 0.325507, 0.253233, 0.258277, 0.202801, 0.147929, 0.2008

Minimum: 0.111056 Maximum: 0.325507 Average: 0.1953 seconds

Time taken for processing each image 6.2828
*****
Length of task list: 13
Number of processes used: 1

The times for each task are: [0.280364, 0.207886, 0.205261, 0.20743, 0.150178, 0.152644, 0.14125

Minimum: 0.101878 Maximum: 0.280364 Average: 0.1587 seconds

```

Time taken for processing each image 5.0843

Length of task list: 13

Number of processes used: 1

The times for each task are: [0.252216, 0.25571, 0.210566, 0.218079, 0.164901, 0.143012, 0.15808

Minimum: 0.099896 Maximum: 0.25571 Average: 0.1636 seconds

Time taken for processing each image 5.9952

Length of task list: 13

Number of processes used: 1

The times for each task are: [0.291491, 0.200126, 0.200883, 0.269857, 0.212641, 0.188935, 0.1887

Minimum: 0.114339 Maximum: 0.291491 Average: 0.1821 seconds

Time taken for processing each image 5.5438

Window sizes used: [1.15, 1.35, 1.55]

1 processes used for testing 3 window sizes

Processing times for each image [6.0213, 5.2771, 6.2828, 5.0843, 5.9952, 5.5438] with an average

Time elapsed so far... 34.2045

#####
#####

Number of processes used: 2 window size 260

Length of task list: 13

Number of processes used: 2

The times for each task are: [0.28233, 0.275224, 0.197493, 0.210052, 0.151867, 0.147919, 0.14399

Minimum: 0.106808 Maximum: 0.28233 Average: 0.1672 seconds

Time taken for processing each image 3.3234

Length of task list: 13

Number of processes used: 2

The times for each task are: [0.2985, 0.213568, 0.317483, 0.23633, 0.174457, 0.161119, 0.152521,

Minimum: 0.099019 Maximum: 0.317483 Average: 0.1766 seconds

Time taken for processing each image 3.5311

Length of task list: 13

Number of processes used: 2

The times for each task are: [0.28358, 0.230728, 0.2101, 0.219151, 0.150691, 0.149436, 0.160176,

Minimum: 0.108533 Maximum: 0.28358 Average: 0.1656 seconds

Time taken for processing each image 2.8611

Length of task list: 13

Number of processes used: 2

The times for each task are: [0.226785, 0.211085, 0.306894, 0.205638, 0.146741, 0.164658, 0.1688

Minimum: 0.107543 Maximum: 0.306894 Average: 0.1776 seconds

Time taken for processing each image 3.8512

Length of task list: 13

Number of processes used: 2

The times for each task are: [0.248145, 0.300341, 0.204399, 0.20345, 0.14955, 0.1486, 0.158486,

Minimum: 0.098823 Maximum: 0.300341 Average: 0.1659 seconds

Time taken for processing each image 3.0783

Length of task list: 13

Number of processes used: 2

The times for each task are: [0.324143, 0.314744, 0.202731, 0.157489, 0.273015, 0.177897, 0.2115

Minimum: 0.101988 Maximum: 0.324143 Average: 0.1902 seconds

Time taken for processing each image 3.3945

Window sizes used: [1.15, 1.35, 1.55]

2 processes used for testing 3 window sizes

Processing times for each image [3.3234, 3.5311, 2.8611, 3.8512, 3.0783, 3.3945] with an average

Time elapsed so far... 54.2441

#####

#####

Number of processes used: 3 window size 260

Length of task list: 13
Number of processes used: 3

The times for each task are: [0.281097, 0.280528, 0.158005, 0.218219, 0.27141, 0.157843, 0.15559

Minimum: 0.105139 Maximum: 0.281097 Average: 0.1741 seconds

Time taken for processing each image 2.7441

Length of task list: 13
Number of processes used: 3

The times for each task are: [0.29322, 0.24382, 0.239113, 0.237603, 0.161211, 0.158549, 0.1573,

Minimum: 0.113331 Maximum: 0.29322 Average: 0.1724 seconds

Time taken for processing each image 3.4237

Length of task list: 13
Number of processes used: 3

The times for each task are: [0.293891, 0.235554, 0.255317, 0.247224, 0.163417, 0.146885, 0.1540

Minimum: 0.103609 Maximum: 0.293891 Average: 0.1745 seconds

Time taken for processing each image 2.9148

Length of task list: 13
Number of processes used: 3

The times for each task are: [0.309055, 0.237023, 0.290608, 0.157716, 0.29682, 0.152718, 0.15599

Minimum: 0.097193 Maximum: 0.309055 Average: 0.1831 seconds

Time taken for processing each image 2.6141

Length of task list: 13
Number of processes used: 3

The times for each task are: [0.28614, 0.246554, 0.235493, 0.236714, 0.155885, 0.181205, 0.15435

Minimum: 0.114517 Maximum: 0.28614 Average: 0.1775 seconds

Time taken for processing each image 2.5888

Length of task list: 13
Number of processes used: 3

The times for each task are: [0.262132, 0.36158, 0.361363, 0.178127, 0.294949, 0.188594, 0.14647

Minimum: 0.105138 Maximum: 0.36158 Average: 0.2042 seconds

Time taken for processing each image 2.8591

Length of task list: 13

Number of processes used: 4

The times for each task are: [0.303087, 0.277174, 0.267526, 0.18654, 0.159533, 0.2911, 0.124407,

Minimum: 0.11578 Maximum: 0.303087 Average: 0.1982 seconds

Time taken for processing each image 2.7363

Length of task list: 13

Number of processes used: 4

The times for each task are: [0.30326, 0.300843, 0.232297, 0.322457, 0.197032, 0.260972, 0.16215

Minimum: 0.110674 Maximum: 0.322457 Average: 0.2052 seconds

Time taken for processing each image 2.8419

Window sizes used: [1.15, 1.35, 1.55]

4 processes used for testing 3 window sizes

Processing times for each image [2.6768, 2.8643, 2.796, 2.8591, 2.7363, 2.8419] with an average

Time elapsed so far... 88.4104

#####

#####

Number of processes used: 5 window size 260

Length of task list: 13

Number of processes used: 5

The times for each task are: [0.270407, 0.240097, 0.276602, 0.242045, 0.208991, 0.239115, 0.1343

Minimum: 0.109094 Maximum: 0.276602 Average: 0.1942 seconds

Time taken for processing each image 2.8464

Length of task list: 13

Number of processes used: 5

The times for each task are: [0.302581, 0.296134, 0.2599, 0.299489, 0.199826, 0.190685, 0.128608

Minimum: 0.117639 Maximum: 0.302581 Average: 0.201 seconds

Time taken for processing each image 2.6082

Length of task list: 13

The times for each task are [0.840818, 0.888486, 0.888185, 0.815338, 0.106705, 0.806135, 0.144911]

Minimum: 0.122147 Maximum: 0.288185 Average: 0.1897 seconds

Time taken for processing each image 3.773

* * * * *

Length of task list: 13

Time taken for processing

Number of processes used: 3

The times for each task are: [0.245404, 0.270021, 0.210001,

WERNER WERNER WERNER WERNER WERNER WERNER

Time taken for processing

* *

Number of processes used: 5

The times for each task are: [0.324969, 0.237644, 0.269975,

Minimum: 0.115994 Maximum: 0.324969 Average

time taken for processing each image 2.9214

5 processes used for testing 5 window size
Processing times for each image [3.8464

Processing times for each image [2.54]

Time elapsed so far 104 720499999999999

Time elapsed so far... 184.7204555555555

Number of processes used: 6 window size 260

Length of task list: 13

Number of processes used: 6

The times for each task are: [0.300639, 0.289116, 0.301077, 0.309188, 0.200872, 0.212689, 0.1429

Minimum: 0.129931 Maximum: 0.309188 Average: 0.2149 seconds

Time taken for processing each image 2.8746

Length of task list: 13

Number of processes used: 6

The times for each task are: [0.272083, 0.236108, 0.274545, 0.242382, 0.183349, 0.149148, 0.1196

Minimum: 0.111405 Maximum: 0.274545 Average: 0.1837 seconds

Time taken for processing each image 3.0472

Length of task list: 13

Number of processes used: 6

The times for each task are: [0.29178, 0.243792, 0.274738, 0.219524, 0.197042, 0.213705, 0.16393

Minimum: 0.132022 Maximum: 0.29178 Average: 0.1947 seconds

Time taken for processing each image 2.8745

Length of task list: 13

Number of processes used: 6

The times for each task are: [0.290104, 0.244333, 0.271598, 0.31113, 0.173778, 0.130468, 0.20836

Minimum: 0.126891 Maximum: 0.31113 Average: 0.2028 seconds

Time taken for processing each image 2.9977

Length of task list: 13

Number of processes used: 6

The times for each task are: [0.288979, 0.285554, 0.315901, 0.217183, 0.157351, 0.124741, 0.2124

Minimum: 0.106884 Maximum: 0.315901 Average: 0.1997 seconds

Time taken for processing each image 2.9104

Length of task list: 13
Number of processes used: 7

The times for each task are: [0.296372, 0.247855, 0.25304, 0.324561, 0.223241, 0.199409, 0.18183]

Minimum: 0.13579 Maximum: 0.324561 Average: 0.2104 seconds

Time taken for processing each image 3.0218

Length of task list: 13

Number of processes used: 7

The times for each task are: [0.260579, 0.240083, 0.305225, 0.27938, 0.240013, 0.207318, 0.16894]

Minimum: 0.105338 Maximum: 0.305225 Average: 0.21 seconds

Time taken for processing each image 3.0575

Length of task list: 13

Number of processes used: 7

The times for each task are: [0.241586, 0.240058, 0.284474, 0.296316, 0.156921, 0.213923, 0.1429]

Minimum: 0.115448 Maximum: 0.296316 Average: 0.195 seconds

Time taken for processing each image 3.0977

Window sizes used: [1, 15, 1, 35, 1, 55]

7 processes used for testing 3 window sizes

Processing times for each image [2.933 3.2439 2.8901 3.0218 3.0575 3.0977] with an average

Time elapsed so far... 140.4683

#####

```
In [121]: use_sub_windows = 1
          use_default_number_of_processes = 0
          print_find_car_step_times = 0
          find_car_hog_parallelism_enabled = 0
          step = 200

          print('Cells per step', cells_per_step)
```

```

xwinstarts, xwinstops = calculate_patches(x_start, x_stop, window, scale, step, cells_
ywinstarts, ywinstops = calculate_patches(y_start, y_stop, window, scale, step, cells_
print_task_times = 1

time_elapsed = 0

for i in range(1,2):
    totalTasks = len(scaleslist[i]) * len(xwinstarts) * len(ywinstarts)
    print('Number of Tasks for scale and step', totalTasks)
    for j in range(1, int(totalTasks/2)+2):
        time_for_all_images = []
        test_images_with_boxes = []
        number_of_processes_used = j
        print('Number of processes used: ', number_of_processes_used, 'window size', s
        print()
        for image in test_images:
            img = mpimg.imread(image)
            freeze_support()
            t0 = time.time()
            r = []
            r = parallelize_find_cars(img, number_of_processes_used, i)
            t1 = time.time()
            processing_time = round(t1-t0, 4)
            print('Time taken for processing each image', processing_time)
            time_for_all_images.append(processing_time)
            img_with_boxes = np.copy(img)
            for box in r:
                cv2.rectangle(img_with_boxes, box[0],box[1],(0,0,255),2)
            #plt.figure(figsize=(16,9))
            #plt.imshow(img_with_boxes)
            test_images_with_boxes.append(img_with_boxes)
        print('*****')
        print('Window sizes used:', scaleslist[i])
        print(number_of_processes_used, 'processes used for testing', len(scaleslist[i]
        print('Processing times for each image', time_for_all_images, \
              'with an average of', round(sum(time_for_all_images)/len(time_for_all_im
        print()
        time_elapsed += sum(time_for_all_images)
        print('Time elapsed so far...', time_elapsed)
        print('#####')
        print('#####')

        print()
        print()
        print('=====')
        print()
        print()

```

Cells per step 4

Number of Tasks for scale and step 18

Number of processes used: 1 window size 200

Length of task list: 36

Number of processes used: 1

The times for each task are: [0.176368, 0.14394, 0.120711, 0.122155, 0.133849, 0.122242, 0.08860

Minimum: 0.056791 Maximum: 0.176368 Average: 0.1004 seconds

Time taken for processing each image 13.6518

Length of task list: 36

Number of processes used: 1

The times for each task are: [0.176066, 0.151076, 0.152519, 0.156013, 0.132267, 0.111387, 0.0883

Minimum: 0.057818 Maximum: 0.176066 Average: 0.0942 seconds

Time taken for processing each image 11.8848

Length of task list: 36

Number of processes used: 1

The times for each task are: [0.176694, 0.145205, 0.125299, 0.127435, 0.133337, 0.125333, 0.0990

Minimum: 0.061361 Maximum: 0.176694 Average: 0.0947 seconds

Time taken for processing each image 13.2039

Length of task list: 36

Number of processes used: 1

The times for each task are: [0.184459, 0.18129, 0.163865, 0.128773, 0.141194, 0.156929, 0.11091

Minimum: 0.04697 Maximum: 0.184459 Average: 0.1016 seconds

Time taken for processing each image 13.4793

Length of task list: 36

Number of processes used: 1

The times for each task are: [0.174201, 0.152881, 0.123515, 0.121564, 0.1374, 0.118989, 0.084152

Minimum: 0.060263 Maximum: 0.174201 Average: 0.0954 seconds

Time taken for processing each image 13.0366


```
Length of task list: 36  
Number of processes used: 2
```

The times for each task are: [0.191322, 0.153961, 0.17845, 0.15186, 0.150544, 0.113928, 0.155384]

Minimum: 0.060166 Maximum: 0.191322 Average: 0.1031 seconds

Time taken for processing each image 6.8818

Length of task list: 36

Number of processes used: 2

The times for each task are: [0.195104, 0.174019, 0.176994, 0.149283, 0.165847, 0.150167, 0.1050]

Minimum: 0.066557 Maximum: 0.195104 Average: 0.1017 seconds

Time taken for processing each image 7.2673

Length of task list: 36

Number of processes used: 2

The times for each task are: [0.178344, 0.124556, 0.181758, 0.148465, 0.137404, 0.122508, 0.072256]

Minimum: 0.054825 Maximum: 0.181758 Average: 0.0908 seconds

Time taken for processing each image 7.1406

Window sizes used: [1.15, 1.35, 1.55]

2 processes used for testing 3 window sizes

Processing times for each image [6.7893, 6.7331, 7.2013, 6.8818, 7.2673, 7.1406] with an average

Time elapsed so far... 121.7502

#####

#####
#####

Number of processes used: 3 window size 200

Length of task list: 36

Number of processes used: 3

The times for each task are: [0.185312, 0.158229, 0.124153, 0.15082, 0.180812, 0.113175, 0.12637]

Minimum: 0.054065 Maximum: 0.185312 Average: 0.0943 seconds

Time taken for processing each image 4.9563

Length of task list: 36

Number of processes used: 3

The times for each task are: [0.184008, 0.180015, 0.211119, 0.154168, 0.12561, 0.072282, 0.15340

Minimum: 0.059998 Maximum: 0.211119 Average: 0.1024 seconds

Time taken for processing each image 5.1839

Length of task list: 36

Number of processes used: 3

The times for each task are: [0.148982, 0.188812, 0.123403, 0.18087, 0.11705, 0.172701, 0.07217,

Minimum: 0.055755 Maximum: 0.188812 Average: 0.0976 seconds

Time taken for processing each image 4.8838

Length of task list: 36

Number of processes used: 3

The times for each task are: [0.146213, 0.177799, 0.124349, 0.185506, 0.125601, 0.176117, 0.1129

Minimum: 0.056724 Maximum: 0.185506 Average: 0.0947 seconds

Time taken for processing each image 5.6151

Length of task list: 36

Number of processes used: 3

The times for each task are: [0.210481, 0.187767, 0.175485, 0.221641, 0.152739, 0.170139, 0.0849

Minimum: 0.057936 Maximum: 0.221641 Average: 0.105 seconds

Time taken for processing each image 5.0544

Length of task list: 36

Number of processes used: 3

The times for each task are: [0.177296, 0.140135, 0.169233, 0.121438, 0.179315, 0.084942, 0.1629

Minimum: 0.053028 Maximum: 0.179315 Average: 0.1033 seconds

Time taken for processing each image 4.9435

Window sizes used: [1.15, 1.35, 1.55]

3 processes used for testing 3 window sizes

Processing times for each image [4.9563, 5.1839, 4.8838, 5.6151, 5.0544, 4.9435] with an average

Time elapsed so far... 152.3872

#####
#####

Number of processes used: 4 window size 200

Length of task list: 36

Number of processes used: 4

The times for each task are: [0.170767, 0.174922, 0.124868, 0.122274, 0.090164, 0.148705, 0.177611]

Minimum: 0.055723 Maximum: 0.17764 Average: 0.091 seconds

Time taken for processing each image 4.3809

Length of task list: 36

Number of processes used: 4

The times for each task are: [0.14698, 0.193358, 0.155429, 0.156265, 0.185044, 0.199875, 0.12641]

Minimum: 0.055915 Maximum: 0.199875 Average: 0.1033 seconds

Time taken for processing each image 4.4533

Length of task list: 36

Number of processes used: 4

The times for each task are: [0.177511, 0.175125, 0.120445, 0.115131, 0.135012, 0.144302, 0.1725]

Minimum: 0.053494 Maximum: 0.177511 Average: 0.0946 seconds

Time taken for processing each image 4.5173

Length of task list: 36

Number of processes used: 4

The times for each task are: [0.180446 0.1489 0.156686 0.166166 0.153977 0.177666 0.140012]

Minimum: 0.043568 Maximum: 0.180446 Average: 0.1016 seconds

Time taken for processing each image 4.5231

Time taken for processing each image: 1.0201

Length of task list: 36

Length of task list: 30
Number of processes used: 4

The times for each task are: [0.153047 0.132377 0.175844 0.148584 0.15065 0.162826 0.18007]

Minimum: 0.057999 Maximum: 0.180073 Average: 0.0988 seconds

Time taken for processing each image 4.3553

Length of task list: 36

Number of processes used: 4

The times for each task are: [0.181594, 0.168073, 0.139225, 0.130129, 0.128263, 0.117978, 0.1777

Minimum: 0.057068 Maximum: 0.181594 Average: 0.0987 seconds

Time taken for processing each image 4.2218

Window sizes used: [1.15, 1.35, 1.55]

4 processes used for testing 3 window sizes

Processing times for each image [4.3809, 4.4533, 4.5173, 4.5231, 4.3553, 4.2218] with an average

Time elapsed so far... 178.8389

#####

Number of processes used: 5 window size 200

Length of task list: 36

Number of processes used: 5

The times for each task are: [0.195014, 0.151098, 0.143948, 0.185588, 0.137868, 0.145013, 0.1808

Minimum: 0.061257 Maximum: 0.195014 Average: 0.1073 seconds

Time taken for processing each image 4.3128

Length of task list: 36

Number of processes used: 5

The times for each task are: [0.178057, 0.150997, 0.154808, 0.199824, 0.217251, 0.090188, 0.1213

Minimum: 0.065827 Maximum: 0.217251 Average: 0.1049 seconds

Time taken for processing each image 4.4781

Length of task list: 36

Number of processes used: 5

The times for each task are: [0.179183, 0.192849, 0.180423, 0.176107, 0.16674, 0.121453, 0.11325

Minimum: 0.058183 Maximum: 0.192849 Average: 0.1069 seconds

Time taken for processing each image 4.2415

Length of task list: 36

Number of processes used: 5

The times for each task are: [0.192445, 0.179579, 0.154271, 0.176769, 0.147012, 0.174423, 0.0972

Minimum: 0.058285 Maximum: 0.192445 Average: 0.1009 seconds

Time taken for processing each image 3.9335

Length of task list: 36

Number of processes used: 5

The times for each task are: [0.145264, 0.189862, 0.151713, 0.082526, 0.152606, 0.203505, 0.1107

Minimum: 0.060639 Maximum: 0.203505 Average: 0.0995 seconds

Time taken for processing each image 4.2386

Length of task list: 36

Number of processes used: 5

The times for each task are: [0.175256, 0.135002, 0.183339, 0.167657, 0.104338, 0.171861, 0.1131

Minimum: 0.06865 Maximum: 0.183339 Average: 0.1025 seconds

Time taken for processing each image 4.2028

Window sizes used: [1.15, 1.35, 1.55]

5 processes used for testing 3 window sizes

Processing times for each image [4.3128, 4.4781, 4.2415, 3.9335, 4.2386, 4.2028] with an average

Time elapsed so far... 204.2462

#####

#####

Number of processes used: 6 window size 200

Length of task list: 36

Number of processes used: 6

The times for each task are: [0.208207, 0.157779, 0.213211, 0.157464, 0.140797, 0.072379, 0.0974

Minimum: 0.056584 Maximum: 0.213211 Average: 0.1028 seconds

Time taken for processing each image 4.1238

Length of task list: 36

Number of processes used: 6

The times for each task are: [0.181327, 0.181009, 0.157396, 0.17591, 0.13252, 0.160565, 0.12996,

Minimum: 0.056651 Maximum: 0.181327 Average: 0.1039 seconds

Time taken for processing each image 3.9108

Length of task list: 36

Number of processes used: 6

The times for each task are: [0.189116, 0.140877, 0.163326, 0.185531, 0.12797, 0.112119, 0.14496,

Minimum: 0.053829 Maximum: 0.189116 Average: 0.0998 seconds

Time taken for processing each image 3.984

Length of task list: 36

Number of processes used: 6

The times for each task are: [0.179028, 0.180248, 0.152805, 0.154688, 0.17297, 0.140271, 0.0709,

Minimum: 0.05781 Maximum: 0.180248 Average: 0.103 seconds

Time taken for processing each image 4.3121

Length of task list: 36

Number of processes used: 6

The times for each task are: [0.18417, 0.177192, 0.154193, 0.181062, 0.151361, 0.119312, 0.11691,

Minimum: 0.046874 Maximum: 0.18417 Average: 0.1042 seconds

Time taken for processing each image 4.1316

Length of task list: 36

Number of processes used: 6

The times for each task are: [0.186596, 0.192044, 0.159067, 0.153632, 0.128695, 0.104311, 0.2038,

Minimum: 0.049975 Maximum: 0.203882 Average: 0.1017 seconds

Time taken for processing each image 4.0216

Window sizes used: [1.15, 1.35, 1.55]

6 processes used for testing 3 window sizes

Processing times for each image [4.1238, 3.9108, 3.984, 4.3121, 4.1316, 4.0216] with an average

Time elapsed so far... 228.7301

#####
#####

Number of processes used: 7 window size 200

Length of task list: 36

Number of processes used: 7

The times for each task are: [0.188061, 0.195913, 0.160872, 0.184183, 0.163522, 0.109954, 0.1089]

Minimum: 0.059464 Maximum: 0.195913 Average: 0.1067 seconds

Time taken for processing each image 3.9358

Length of task list: 36

Number of processes used: 7

The times for each task are: [0.183035, 0.131421, 0.148618, 0.165249, 0.132141, 0.152565, 0.0872]

Minimum: 0.049213 Maximum: 0.183035 Average: 0.1008 seconds

Time taken for processing each image 4.2153

Length of task list: 36

Number of processes used: 7

The times for each task are: [0.176509, 0.153584, 0.182304, 0.077485, 0.190375, 0.195714, 0.096031]

Minimum: 0.062537 Maximum: 0.195714 Average: 0.1092 seconds

Time taken for processing each image 4.5928

Length of task list: 36

Number of processes used: 7

The times for each task are: [0.152735, 0.182606, 0.145205, 0.174624, 0.176973, 0.15305, 0.11605]

Minimum: 0.057287 Maximum: 0.182606 Average: 0.1077 seconds

Time taken for processing each image 4.0812

Length of task list: 36

Number of processes used: 7

Minimum: 0.059089 Maximum: 0.185624 Average: 0.1058 seconds

Time taken for processing each image 4.5782

Length of task list: 36

Number of processes used: 9

The times for each task are: [0.185225, 0.17987, 0.153692, 0.173905, 0.154171, 0.15145, 0.136456]

Minimum: 0.059019 Maximum: 0.185225 Average: 0.1041 seconds

Time taken for processing each image 4.4778

Length of task list: 36

Number of processes used: 9

The times for each task are: [0.183949, 0.141542, 0.144106, 0.155248, 0.123922, 0.177943, 0.1170]

Minimum: 0.065581 Maximum: 0.183949 Average: 0.1055 seconds

Time taken for processing each image 4.3164

Length of task list: 36

Number of processes used: 9

The times for each task are: [0.181079, 0.152581, 0.14943, 0.152928, 0.190825, 0.156884, 0.11627]

Minimum: 0.067415 Maximum: 0.190825 Average: 0.1088 seconds

Time taken for processing each image 4.6392

Length of task list: 36

Number of processes used: 9

The times for each task are: [0.147687, 0.146964, 0.12806, 0.147597, 0.151593, 0.178413, 0.11647]

Minimum: 0.05548 Maximum: 0.178413 Average: 0.1014 seconds

Time taken for processing each image 4.2212

Length of task list: 36

Number of processes used: 9

The times for each task are: [0.145677, 0.178835, 0.162934, 0.17635, 0.152115, 0.114437, 0.17865]

Minimum: 0.056076 Maximum: 0.178835 Average: 0.106 seconds

Time taken for processing each image 4.474

Window sizes used: [1.15, 1.35, 1.55]

9 processes used for testing 3 window sizes

Ti = 1 d = 6 306-1122

```
Time elapsed so far... 300.1182
#####
#####
```

Number of processes used: 10 window size 200

Length of task list: 36

Number of processes used: 10

The times for each task are: [0.185056, 0.155103, 0.158232, 0.16063, 0.19408, 0.206594, 0.130212]

Minimum: 0.056252 Maximum: 0.206594 Average: 0.1098 seconds

Time taken for processing each image 4.4251

Length of task list: 36

Number of processes used: 10

The times for each task are: [0.213733, 0.154538, 0.160454, 0.186713, 0.182122, 0.148775, 0.115911]

Minimum: 0.058339 Maximum: 0.213733 Average: 0.1065 seconds

Time taken for processing each image 4.6339

Length of task list: 36

Number of processes used: 10

The times for each task are: [0.179462, 0.160598, 0.176666, 0.151478, 0.155824, 0.177577, 0.139]

Minimum: 0.058567 Maximum: 0.179462 Average: 0.1101 seconds

Time taken for processing each image 4.5881

Length of task list: 36

Number of processes used: 10

The times for each task are: [0.180025, 0.152542, 0.153629, 0.184968, 0.162199, 0.188476, 0.129411]

Minimum: 0.058098 Maximum: 0.188476 Average: 0.1069 seconds

Time taken for processing each image 4.6363

```
*****
Length of task list: 36
Number of processes used: 10

The times for each task are: [0.179677, 0.174785, 0.155152, 0.120747, 0.145494, 0.179127, 0.1176

Minimum: 0.0577 Maximum: 0.179677 Average: 0.1039 seconds

Time taken for processing each image 4.7563
*****
Length of task list: 36
Number of processes used: 10

The times for each task are: [0.198093, 0.172029, 0.157143, 0.190512, 0.154548, 0.181888, 0.1176

Minimum: 0.059689 Maximum: 0.198093 Average: 0.1101 seconds

Time taken for processing each image 4.8882
*****
Window sizes used: [1.15, 1.35, 1.55]
10 processes used for testing 3 window sizes
Processing times for each image [4.4251, 4.6339, 4.5881, 4.6363, 4.7563, 4.8882] with an average

Time elapsed so far... 334.0461
#####
=====
```

In [122]:

```
use_sub_windows = 1
use_default_number_of_processes = 0
print_find_car_step_times = 0
find_car_hog_parallelism_enabled = 0
step = 170
xwinstarts, xwinstops = calculate_patches(x_start, x_stop, window, scale, step, cells_
ywinstarts, ywinstops = calculate_patches(y_start, y_stop, window, scale, step, cells_
print_task_times = 1

time_elapsed = 0

for i in range(1,2):
    totalTasks = len(scaleslist[i]) * len(xwinstarts) * len(ywinstarts)
    print('Number of Tasks for scale and step', totalTasks)
```

```

time_for_all_images = []
test_images_with_boxes = []
number_of_processes_used = j
print('Number of processes used: ', number_of_processes_used, 'window size', s
print()
for image in test_images:
    img = mpimg.imread(image)
    freeze_support()
    t0 = time.time()
    r = []
    r = parallelize_find_cars(img, number_of_processes_used, i)
    t1 = time.time()
    processing_time = round(t1-t0, 4)
    print('Time taken for processing each image', processing_time)
    time_for_all_images.append(processing_time)
    img_with_boxes = np.copy(img)
    for box in r:
        cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
    #plt.figure(figsize=(16,9))
    #plt.imshow(img_with_boxes)
    test_images_with_boxes.append(img_with_boxes)
    print('*****')
print('Window sizes used:', scaleslist[i])
print(number_of_processes_used, 'processes used for testing', len(scaleslist[i])
print('Processing times for each image', time_for_all_images, \
      'with an average of', round(sum(time_for_all_images)/len(time_for_all_im
print()
time_elapsed += sum(time_for_all_images)
print('Time elapsed so far...', time_elapsed)
print('#####')
print('#####')

print()
print()
print('=====')
print()
print()

Number of Tasks for scale and step 48
Number of processes used: 1 window size 170

Length of task list: 69
Number of processes used: 1

```

The times for each task are: [0.14116, 0.111758, 0.115295, 0.108036, 0.113184, 0.105751, 0.11832

Minimum: 0.044452 Maximum: 0.14116 Average: 0.0853 seconds

Time taken for processing each image 22.9794

Length of task list: 69

Number of processes used: 1

The times for each task are: [0.141114, 0.103831, 0.111555, 0.106505, 0.11507, 0.10751, 0.133163]

Minimum: 0.043704 Maximum: 0.141114 Average: 0.0808 seconds

Time taken for processing each image 24.0537

Length of task list: 69

Number of processes used: 1

The times for each task are: [0.142573, 0.106515, 0.117975, 0.115806, 0.113383, 0.118269, 0.1143]

Minimum: 0.042559 Maximum: 0.147656 Average: 0.0843 seconds

Time taken for processing each image 24.6313

Length of task list: 69

Number of processes used: 1

The times for each task are: [0.138156, 0.102442, 0.11738, 0.107155, 0.11755, 0.113958, 0.116625]

Minimum: 0.040746 Maximum: 0.138156 Average: 0.0805 seconds

Time taken for processing each image 22.7181

Length of task list: 69

Number of processes used: 1

The times for each task are: [0.110067, 0.085828, 0.089675, 0.08309, 0.088944, 0.106395, 0.16216]

Minimum: 0.042085 Maximum: 0.16216 Average: 0.0802 seconds

Time taken for processing each image 23.1506

Length of task list: 69

Number of processes used: 1

The times for each task are: [0.150868, 0.125671, 0.127991, 0.126581, 0.128365, 0.137218, 0.1424]

Minimum: 0.042224 Maximum: 0.150868 Average: 0.0805 seconds

Time taken for processing each image 24.9503

Window sizes used: [1.15, 1.35, 1.55]

1 processes used for testing 3 window sizes

Processing times for each image [22.9794, 24.0537, 24.6313, 22.7181, 23.1506, 24.9503] with an average of 23.8811.

Time elapsed so far... 142.4834

#####
#####

Number of processes used: 2 window size 170

Length of task list: 69

Number of processes used: 2

The times for each task are: [0.144118, 0.129329, 0.122084, 0.108771, 0.110767, 0.109698, 0.134611]

Minimum: 0.041391 Maximum: 0.144118 Average: 0.0812 seconds

Time taken for processing each image 12.7054

Length of task list: 69

Number of processes used: 2

The times for each task are: [0.143162, 0.1288, 0.119029, 0.110165, 0.120807, 0.106707, 0.104905]

Minimum: 0.042247 Maximum: 0.143162 Average: 0.0783 seconds

Time taken for processing each image 13.1943

Length of task list: 69

Number of processes used: 2

The times for each task are: [0.136803, 0.132566, 0.110533, 0.10589, 0.111718, 0.104297, 0.10464]

Minimum: 0.040095 Maximum: 0.136803 Average: 0.0795 seconds

Time taken for processing each image 11.3142

Length of task list: 69

Number of processes used: ?

The times for each task are: [0.126225 0.137037 0.10562 0.114615 0.113214 0.104395 0.11108]

Minimum: 0.043416 Maximum: 0.137037 Average: 0.0806 seconds

Time taken for processing each image 12.7563

TIME TAKEN FOR PROCESSING EACH IMAGE: 12.7000

Length of task list: 69

Length of task list: 65
Number of processes used: 3

The times for each task are: [0.146912, 0.116305, 0.147772, 0.108825, 0.135937, 0.110926, 0.1129

Minimum: 0.044076 Maximum: 0.147772 Average: 0.0776 seconds

Time taken for processing each image 8.0688

Length of task list: 69

Number of processes used: 3

The times for each task are: [0.14149, 0.096609, 0.088804, 0.1223, 0.082002, 0.144283, 0.09048,

Minimum: 0.034683 Maximum: 0.144283 Average: 0.0743 seconds

Time taken for processing each image 9.3631

Length of task list: 69

Number of processes used: 3

The times for each task are: [0.141694, 0.13146, 0.117489, 0.137551, 0.112646, 0.116964, 0.10549

Minimum: 0.041063 Maximum: 0.143607 Average: 0.0818 seconds

Time taken for processing each image 9.2858

Length of task list: 69

Number of processes used: 3

The times for each task are: [0.143457, 0.142226, 0.136807, 0.133991, 0.118964, 0.103165, 0.1256

Minimum: 0.039053 Maximum: 0.155152 Average: 0.0829 seconds

Time taken for processing each image 8.6755

Window sizes used: [1.15, 1.35, 1.55]

3 processes used for testing 3 window sizes

Processing times for each image [9.0366, 8.4082, 8.0688, 9.3631, 9.2858, 8.6755] with an average

Time elapsed so far... 268.4043

#####

Number of processes used: 4 window size 170

Length of task list: 69

Number of processes used: 4

The times for each task are: [0.141758, 0.105083, 0.157372, 0.104057, 0.1062, 0.102374, 0.138545

Minimum: 0.040406 Maximum: 0.157372 Average: 0.0824 seconds

Time taken for processing each image 7.1687

Length of task list: 69
Number of processes used: 4

The times for each task are: [0.145022, 0.112754, 0.143546, 0.105258, 0.155579, 0.172652, 0.1109

Minimum: 0.034897 Maximum: 0.172652 Average: 0.0794 seconds

Time taken for processing each image 7.0507

Length of task list: 69
Number of processes used: 4

The times for each task are: [0.148184, 0.126549, 0.133138, 0.141268, 0.137651, 0.129532, 0.1427

Minimum: 0.038029 Maximum: 0.148184 Average: 0.0789 seconds

Time taken for processing each image 7.1479

Length of task list: 69
Number of processes used: 4

The times for each task are: [0.140247, 0.10868, 0.116493, 0.142654, 0.11286, 0.124761, 0.083852

Minimum: 0.0393 Maximum: 0.142654 Average: 0.0804 seconds

Time taken for processing each image 6.9467

Length of task list: 69
Number of processes used: 4

The times for each task are: [0.139687, 0.106539, 0.157211, 0.130512, 0.139103, 0.105938, 0.1490

Minimum: 0.039267 Maximum: 0.157211 Average: 0.0824 seconds

Time taken for processing each image 7.2641

Length of task list: 69
Number of processes used: 4

The times for each task are: [0.140299, 0.108988, 0.090923, 0.097797, 0.091049, 0.084157, 0.1314

Minimum: 0.037435 Maximum: 0.140299 Average: 0.0755 seconds

Time taken for processing each image 7.0316

Window sizes used: [1.15, 1.35, 1.55]

4 processes used for testing 3 window sizes

Processing times for each image [7.1687, 7.0507, 7.1479, 6.9467, 7.2641, 7.0316] with an average

Time elapsed so far... 311.01399999999995

#####

#####

Number of processes used: 5 window size 170

Length of task list: 69

Number of processes used: 5

The times for each task are: [0.151008, 0.111368, 0.085584, 0.087831, 0.127322, 0.142808, 0.1245

Minimum: 0.042507 Maximum: 0.151008 Average: 0.0825 seconds

Time taken for processing each image 6.2905

Length of task list: 69

Number of processes used: 5

The times for each task are: [0.143628, 0.112898, 0.115025, 0.136238, 0.098012, 0.117207, 0.1157

Minimum: 0.035696 Maximum: 0.154322 Average: 0.0812 seconds

Time taken for processing each image 6.6466

Length of task list: 69

Number of processes used: 5

The times for each task are: [0.149538, 0.104141, 0.096157, 0.126859, 0.112673, 0.096094, 0.1406

Minimum: 0.037674 Maximum: 0.162912 Average: 0.0793 seconds

Time taken for processing each image 6.6543

Length of task list: 69

Number of processes used: 5

The times for each task are: [0.142695, 0.138122, 0.094415, 0.134646, 0.121546, 0.10873, 0.14560

Minimum: 0.043487 Maximum: 0.172041 Average: 0.0823 seconds

Time taken for processing each image 6.0066

Length of task list: 69
Number of processes used: 6

The times for each task are: [0.139515, 0.109049, 0.140861, 0.128432, 0.114916, 0.107806, 0.1503

Minimum: 0.040708 Maximum: 0.153381 Average: 0.083 seconds

Time taken for processing each image 6.2207

Length of task list: 69
Number of processes used: 6

The times for each task are: [0.142483, 0.127641, 0.113682, 0.12916, 0.142624, 0.13228, 0.122884

Minimum: 0.04248 Maximum: 0.143137 Average: 0.0841 seconds

Time taken for processing each image 6.1343

Length of task list: 69
Number of processes used: 6

The times for each task are: [0.147283, 0.128398, 0.104699, 0.133281, 0.12061, 0.130539, 0.14019

Minimum: 0.044073 Maximum: 0.147283 Average: 0.0834 seconds

Time taken for processing each image 6.2274

Length of task list: 69
Number of processes used: 6

The times for each task are: [0.151433, 0.11331, 0.143422, 0.10783, 0.115726, 0.106384, 0.135984

Minimum: 0.040465 Maximum: 0.160387 Average: 0.0821 seconds

Time taken for processing each image 6.0147

Window sizes used: [1.15, 1.35, 1.55]

6 processes used for testing 3 window sizes

Processing times for each image [6.1421, 6.0505, 6.2207, 6.1343, 6.2274, 6.0147] with an average

Time elapsed so far... 386.06639999999993

#####

#####

Number of processes used: 7 window size 170

Length of task list: 69

Number of processes used: 7

The times for each task are: [0.148762, 0.112075, 0.112797, 0.137966, 0.13369, 0.113117, 0.10800

Minimum: 0.040841 Maximum: 0.171063 Average: 0.0833 seconds

Time taken for processing each image 5.8741

Length of task list: 69

Number of processes used: 7

The times for each task are: [0.120643, 0.110555, 0.146553, 0.130502, 0.113708, 0.151549, 0.1296

Minimum: 0.039171 Maximum: 0.176447 Average: 0.0838 seconds

Time taken for processing each image 6.3676

Length of task list: 69

Number of processes used: 7

The times for each task are: [0.151534, 0.110504, 0.114949, 0.092921, 0.13606, 0.132575, 0.10811

Minimum: 0.036775 Maximum: 0.157768 Average: 0.0824 seconds

Time taken for processing each image 5.7222

Length of task list: 69

Number of processes used: 7

The times for each task are: [0.128416, 0.113182, 0.130055, 0.145334, 0.118601, 0.157867, 0.1342

Minimum: 0.040106 Maximum: 0.157867 Average: 0.0835 seconds

Time taken for processing each image 5.7583

Length of task list: 69

Number of processes used: 7

The times for each task are: [0.144744, 0.139097, 0.102096, 0.129989, 0.119701, 0.119204, 0.1056

Minimum: 0.044111 Maximum: 0.147035 Average: 0.0841 seconds

Time taken for processing each image 5.9384

Length of task list: 69

Number of processes used: 7

The times for each task are: [0.135631, 0.102752, 0.114968, 0.106333, 0.145786, 0.145703, 0.1290

Minimum: 0.042642 Maximum: 0.145786 Average: 0.0822 seconds

Time taken for processing each image 5.8743

Window sizes used: [1.15, 1.35, 1.55]

7 processes used for testing 3 window sizes

Ti: 1 1 6 421 601000000000

```
Time elapsed so far... 421.8012999999999  
#####
#####
```

Number of processes used: 8 window size 170

Length of task list: 69

Number of processes used: 8

The times for each task are: [0.147144, 0.139805, 0.119818, 0.155885, 0.139063, 0.137773, 0.121212]

Minimum: 0.036583 Maximum: 0.157752 Average: 0.0839 seconds

Time taken for processing each image 5.799

Length of task list: 69

Number of processes used: 8

The times for each task are: [0.136739, 0.125748, 0.115623, 0.105247, 0.145263, 0.124605, 0.1145]

Minimum: 0.040287 Maximum: 0.145263 Average: 0.0808 seconds

Time taken for processing each image 5.6489

Length of task list: 69

Number of processes used: 8

The times for each task are: [0.137697, 0.135923, 0.110177, 0.120781, 0.131342, 0.139811, 0.1358]

Minimum: 0.042692 Maximum: 0.139811 Average: 0.0844 seconds

Time taken for processing each image 5.6239

Length of task list: 69

Number of processes used: 8

The times for each task are: [0.136113, 0.146232, 0.120913, 0.10515, 0.148831, 0.129752, 0.11758]

Minimum: 0.042659 Maximum: 0.148831 Average: 0.0813 seconds

Time taken for processing each image 5.529

Length of task list: 69

Number of processes used: 8

The times for each task are: [0.141563, 0.109539, 0.114197, 0.133416, 0.114856, 0.149143, 0.1363

Minimum: 0.039421 Maximum: 0.157979 Average: 0.0826 seconds

Time taken for processing each image 5.6683

Length of task list: 69

Number of processes used: 8

The times for each task are: [0.146779, 0.158722, 0.116518, 0.128971, 0.127196, 0.128918, 0.1349

Minimum: 0.043577 Maximum: 0.158722 Average: 0.0835 seconds

Time taken for processing each image 5.3494

Window sizes used: [1.15, 1.35, 1.55]

8 processes used for testing 3 window sizes

Processing times for each image [5.799, 5.6489, 5.6239, 5.529, 5.6683, 5.3494] with an average o

Time elapsed so far... 455.2197999999999

#####

#####

Number of processes used: 9 window size 170

Length of task list: 69

Number of processes used: 9

The times for each task are: [0.147792, 0.109438, 0.124953, 0.141829, 0.122978, 0.132635, 0.1249

Minimum: 0.042079 Maximum: 0.147792 Average: 0.0831 seconds

Time taken for processing each image 5.8836

Length of task list: 69

Number of processes used: 9

The times for each task are: [0.140624, 0.131073, 0.129207, 0.114496, 0.141533, 0.1223, 0.135669

Minimum: 0.038307 Maximum: 0.14953 Average: 0.0828 seconds

Time taken for processing each image 5.6319

Length of task list: 69

Number of processes used: 9

The times for each task are: [0.143782, 0.135441, 0.094372, 0.126002, 0.115757, 0.108845, 0.1491

Minimum: 0.03861 Maximum: 0.149194 Average: 0.0838 seconds

Time taken for processing each image 5.7694

Length of task list: 69

Number of processes used: 9

The times for each task are: [0.146384, 0.138949, 0.096469, 0.108243, 0.122079, 0.110305, 0.1400

Minimum: 0.041541 Maximum: 0.172713 Average: 0.0824 seconds

Time taken for processing each image 5.7016

Length of task list: 69

Number of processes used: 9

The times for each task are: [0.110369, 0.08562, 0.092216, 0.127317, 0.133687, 0.092825, 0.11457

Minimum: 0.040202 Maximum: 0.146737 Average: 0.0833 seconds

Time taken for processing each image 5.214

Length of task list: 69

Number of processes used: 9

The times for each task are: [0.136063, 0.10638, 0.11437, 0.106296, 0.155547, 0.107939, 0.108924

Minimum: 0.038717 Maximum: 0.155547 Average: 0.0833 seconds

Time taken for processing each image 5.7121

Window sizes used: [1.15, 1.35, 1.55]

9 processes used for testing 3 window sizes

Processing times for each image [5.8836, 5.6319, 5.7694, 5.7016, 5.214, 5.7121] with an average

Time elapsed so far... 489.1323999999999

#####

#####

Number of processes used: 10 window size 170

Length of task list: 69

Number of processes used: 10

The times for each task are: [0.117233, 0.101865, 0.097834, 0.089535, 0.141659, 0.114741, 0.1299

Minimum: 0.042014 Maximum: 0.152407 Average: 0.0838 seconds

Time taken for processing each image 5.8618

Length of task list: 69

Number of processes used: 10

The times for each task are: [0.123316, 0.10526, 0.155801, 0.097336, 0.145804, 0.15431, 0.114766

Minimum: 0.042745 Maximum: 0.165793 Average: 0.0838 seconds

Time taken for processing each image 5.7827

Length of task list: 69

Number of processes used: 10

The times for each task are: [0.136067, 0.093316, 0.104228, 0.14307, 0.098455, 0.105342, 0.15675

Minimum: 0.034225 Maximum: 0.156757 Average: 0.0803 seconds

Time taken for processing each image 5.7243

Length of task list: 69

Number of processes used: 10

The times for each task are: [0.141645, 0.132063, 0.146844, 0.138626, 0.133363, 0.109829, 0.1061

Minimum: 0.041552 Maximum: 0.163027 Average: 0.0837 seconds

Time taken for processing each image 5.9713

Length of task list: 69

Number of processes used: 10

The times for each task are: [0.150589, 0.103823, 0.113107, 0.133854, 0.114838, 0.130128, 0.1139

Minimum: 0.040262 Maximum: 0.150589 Average: 0.0832 seconds

Time taken for processing each image 5.8559

Length of task list: 69

Number of processes used: 10

The times for each task are: [0.135626, 0.117528, 0.10482, 0.126773, 0.111295, 0.121578, 0.13755

Minimum: 0.041337 Maximum: 0.137554 Average: 0.0817 seconds

Time taken for processing each image 5.8462

Window sizes used: [1.15, 1.35, 1.55]

10 processes used for testing 3 window sizes

Processing times for each image [5.8618, 5.7827, 5.7243, 5.9713, 5.8559, 5.8462] with an average

Time elapsed so far... 524.1745999999999

#####

#####

Number of processes used: 11 window size 170

Length of task list: 69

Number of processes used: 11

The times for each task are: [0.149557, 0.14148, 0.103358, 0.117658, 0.131298, 0.129504, 0.11987

Minimum: 0.041332 Maximum: 0.18504 Average: 0.0868 seconds

Time taken for processing each image 5.9912

Length of task list: 69

Number of processes used: 11

The times for each task are: [0.138853, 0.107599, 0.11644, 0.119686, 0.124597, 0.130281, 0.11059

Minimum: 0.038942 Maximum: 0.138853 Average: 0.0799 seconds

Time taken for processing each image 5.7273

Length of task list: 69

Number of processes used: 11

The times for each task are: [0.141752, 0.100112, 0.121629, 0.0883, 0.140256, 0.120876, 0.134123

Minimum: 0.041074 Maximum: 0.141752 Average: 0.0811 seconds

Time taken for processing each image 5.9954

Length of task list: 69

Number of processes used: 11

The times for each task are: [0.172841, 0.120209, 0.111557, 0.09171, 0.143728, 0.10669, 0.116899
Minimum: 0.037884 Maximum: 0.203152 Average: 0.085 seconds

Time taken for processing each image 6.2736

Length of task list: 69
Number of processes used: 12

The times for each task are: [0.13759, 0.107632, 0.113343, 0.0882, 0.138821, 0.129519, 0.115113,
Minimum: 0.03945 Maximum: 0.138821 Average: 0.0809 seconds

Time taken for processing each image 6.1361

Length of task list: 69
Number of processes used: 12

The times for each task are: [0.11354, 0.120994, 0.095937, 0.126946, 0.131035, 0.089759, 0.13219
Minimum: 0.038508 Maximum: 0.137111 Average: 0.0852 seconds

Time taken for processing each image 6.071

Length of task list: 69
Number of processes used: 12

The times for each task are: [0.147617, 0.111678, 0.118315, 0.134867, 0.112494, 0.133756, 0.1336
Minimum: 0.039298 Maximum: 0.148801 Average: 0.085 seconds

Time taken for processing each image 6.0916

Length of task list: 69
Number of processes used: 12

The times for each task are: [0.116294, 0.11681, 0.091252, 0.128632, 0.114091, 0.143336, 0.13219
Minimum: 0.041671 Maximum: 0.145578 Average: 0.0831 seconds

Time taken for processing each image 6.0399

Window sizes used: [1.15, 1.35, 1.55]
12 processes used for testing 3 window sizes
Processing times for each image [5.9641, 6.2736, 6.1361, 6.071, 6.0916, 6.0399] with an average

Time elapsed so far... 596.3013999999998

#####

```
#####
Number of processes used: 13 window size 170

Length of task list: 69
Number of processes used: 13

The times for each task are: [0.143576, 0.124364, 0.1159, 0.132066, 0.134139, 0.103426, 0.143584

Minimum: 0.04208 Maximum: 0.146754 Average: 0.0844 seconds

Time taken for processing each image 6.3375
*****
Length of task list: 69
Number of processes used: 13

The times for each task are: [0.148656, 0.14869, 0.112366, 0.112801, 0.119437, 0.10533, 0.140377

Minimum: 0.042948 Maximum: 0.153223 Average: 0.0853 seconds

Time taken for processing each image 6.2991
*****
Length of task list: 69
Number of processes used: 13

The times for each task are: [0.143132, 0.107737, 0.159781, 0.111285, 0.148167, 0.126978, 0.1378

Minimum: 0.040287 Maximum: 0.159781 Average: 0.0847 seconds

Time taken for processing each image 6.4198
*****
Length of task list: 69
Number of processes used: 13

The times for each task are: [0.118313, 0.115033, 0.086443, 0.110956, 0.14399, 0.135051, 0.11326

Minimum: 0.042947 Maximum: 0.146462 Average: 0.0856 seconds

Time taken for processing each image 6.2657
*****
Length of task list: 69
Number of processes used: 13

The times for each task are: [0.139919, 0.088752, 0.1475, 0.090007, 0.123683, 0.108945, 0.135812

Minimum: 0.044501 Maximum: 0.1475 Average: 0.0831 seconds
```

Time taken for processing each image 6.3685

Length of task list: 69

Number of processes used: 13

The times for each task are: [0.164893, 0.101918, 0.161503, 0.112481, 0.14056, 0.14755, 0.123188]

Minimum: 0.04311 Maximum: 0.164893 Average: 0.0861 seconds

Time taken for processing each image 6.2351

Window sizes used: [1.15, 1.35, 1.55]

13 processes used for testing 3 window sizes

Processing times for each image [6.3375, 6.2991, 6.4198, 6.2657, 6.3685, 6.2351] with an average

Time elapsed so far... 634.2270999999998

#####

#####

Number of processes used: 14 window size 170

Length of task list: 69

Number of processes used: 14

The times for each task are: [0.113908, 0.09371, 0.137259, 0.096377, 0.116633, 0.134917, 0.141498]

Minimum: 0.042248 Maximum: 0.153881 Average: 0.0848 seconds

Time taken for processing each image 6.4622

Length of task list: 69

Number of processes used: 14

The times for each task are: [0.146424, 0.112163, 0.127805, 0.171691, 0.12671, 0.127683, 0.127001]

Minimum: 0.038878 Maximum: 0.171691 Average: 0.0866 seconds

Time taken for processing each image 6.4253

Length of task list: 69

Number of processes used: 14

The times for each task are: [0.148711, 0.096749, 0.127049, 0.126907, 0.133389, 0.08901, 0.158094]

Minimum: 0.040118 Maximum: 0.158094 Average: 0.0829 seconds

Time taken for processing each image 6.5861

Length of task list: 69
Number of processes used: 15

The times for each task are: [0.13738, 0.148516, 0.154427, 0.144879, 0.133455, 0.134351, 0.11697

Minimum: 0.03367 Maximum: 0.154427 Average: 0.088 seconds

Time taken for processing each image 6.6332

Length of task list: 69
Number of processes used: 15

The times for each task are: [0.131352, 0.125079, 0.116542, 0.134526, 0.138911, 0.133753, 0.1126

Minimum: 0.038423 Maximum: 0.153345 Average: 0.0855 seconds

Time taken for processing each image 6.5915

Length of task list: 69
Number of processes used: 15

The times for each task are: [0.184418, 0.125967, 0.119796, 0.147316, 0.114923, 0.116452, 0.1398

Minimum: 0.038558 Maximum: 0.184418 Average: 0.0865 seconds

Time taken for processing each image 6.9051

Length of task list: 69
Number of processes used: 15

The times for each task are: [0.110096, 0.084597, 0.113043, 0.087509, 0.13018, 0.158925, 0.13776

Minimum: 0.042899 Maximum: 0.158925 Average: 0.0847 seconds

Time taken for processing each image 6.9999

Length of task list: 69
Number of processes used: 15

The times for each task are: [0.115889, 0.096058, 0.139606, 0.104844, 0.138882, 0.140783, 0.1291

Minimum: 0.039369 Maximum: 0.140783 Average: 0.085 seconds

Time taken for processing each image 6.8249

Window sizes used: [1.15, 1.35, 1.55]
15 processes used for testing 3 window sizes
Processing times for each image [7.0315, 6.6332, 6.5915, 6.9051, 6.9999, 6.8249] with an average

Time elapsed so far... 714.4883999999998

Number of processes used: 16 window size 170

Length of task list: 69

Number of processes used: 16

The times for each task are: [0.137395, 0.128862, 0.116458, 0.105947, 0.097528, 0.108839, 0.109211]

Minimum: 0.036233 Maximum: 0.161866 Average: 0.0863 seconds

Time taken for processing each image 7.0302

Length of task list: 69

Number of processes used: 16

The times for each task are: [0.142292, 0.109734, 0.116473, 0.104733, 0.14689, 0.132664, 0.12161]

Minimum: 0.04173 Maximum: 0.183991 Average: 0.0875 seconds

Time taken for processing each image 6.8907

Length of task list: 69

Number of processes used: 16

The times for each task are: [0.136404, 0.105554, 0.098396, 0.088405, 0.151109, 0.127429, 0.094311]

Minimum: 0.041124 Maximum: 0.167301 Average: 0.0864 seconds

Time taken for processing each image 6.8472

Length of task list: 69

Number of processes used: 16

The times for each task are: [0.141269, 0.10666, 0.105603, 0.119927, 0.142985, 0.126675, 0.11448]

Minimum: 0.041637 Maximum: 0.153561 Average: 0.0853 seconds

Time taken for processing each image 6.8595

***** SEARCH FOR PROCESSING EACH IMAGE *****

Length of task list: 69

Number of processes used: 16

The times for each task are: [0.119197 0.088487 0.091848 0.127429 0.095065 0.103719 0.1436]

Minimum: 0.038647 Maximum: 0.151157 Average: 0.0859 seconds

Time taken for processing each image 7.0282

Length of task list: 69

The entire set of samples obtained from the 11 locations is shown in Table 1.

Minimum: -0.042536 Maximum: 0.143034 Average: 0.0853 second

Time taken for processing each image 7.2059

Time taken for processing each image

Window sizes used: [1.15, 1.35, 1.55]

16 processes used for testing 3 window sizes

Time elapsed so far... 756.3500999999999

#####

Number of processes used: 17 window size 170

Length of task list: 69

Number of processes used: 17

The times for each task are: [0.105112, 0.144047, 0.096709, 0.087178, 0.142676, 0.133365, 0.137711]

Minimum: 0.036852 Maximum: 0.144047 Average: 0.0864 seconds

Time taken for processing each image 7.2869

Length of task list: 69

Number of processes used: 17

The times for each task are: [0.109345, 0.116787, 0.096377, 0.084843, 0.142735, 0.139468, 0.1337]

Minimum: 0.038787 Maximum: 0.142735 Average: 0.0867 seconds

Time taken for processing each image 7.1911

Length of task list: 69

Number of processes used: 17

The times for each task are: [0.1313883, 0.137366, 0.09181, 0.107143, 0.130625, 0.085386, 0.16145]

Minimum: 0.032952 Maximum: 0.358871 Average: 0.1016 seconds

Time taken for processing each image 7.3241

Length of task list: 69

Number of processes used: 17

The times for each task are: [0.139455, 0.105135, 0.109729, 0.133575, 0.112216, 0.109978, 0.1139]

Minimum: 0.041075 Maximum: 0.139455 Average: 0.0865 seconds

Time taken for processing each image 7.3283

Length of task list: 69

Number of processes used: 17

The times for each task are: [0.11138, 0.133397, 0.091979, 0.10214, 0.132337, 0.13465, 0.138703,

Minimum: 0.036131 Maximum: 0.142798 Average: 0.0849 seconds

Time taken for processing each image 7.2

Length of task list: 69

Number of processes used: 17

The times for each task are: [0.137239, 0.105579, 0.133096, 0.108707, 0.136507, 0.12419, 0.10674]

Minimum: 0.036311 Maximum: 0.142945 Average: 0.0841 seconds

Time taken for processing each image 7.0082

Window sizes used: [1.15, 1.35, 1.55]

17 processes used for testing 3 window sizes

Processing times for each image [7.2869, 7.1911, 7.3241, 7.3283, 7.2, 7.0082] with an average of

Time elapsed so far... 799.6886999999999

#####

#####

Number of processes used: 18 window size 170

Length of task list: 69

Number of processes used: 18

The times for each task are: [0.140849, 0.107295, 0.11042, 0.089547, 0.113409, 0.106933, 0.14712]

Minimum: 0.035623 Maximum: 0.174271 Average: 0.0903 seconds

Time taken for processing each image 7.6839

Length of task list: 69

Number of processes used: 18

The times for each task are: [0.145983, 0.101789, 0.118923, 0.135384, 0.116439, 0.133809, 0.1136

Minimum: 0.04255 Maximum: 0.162153 Average: 0.0899 seconds

Time taken for processing each image 7.7419

Length of task list: 69

Number of processes used: 18

The times for each task are: [0.137215, 0.110391, 0.103998, 0.129656, 0.116914, 0.103051, 0.1457

Minimum: 0.037827 Maximum: 0.145715 Average: 0.0891 seconds

Time taken for processing each image 7.29

Length of task list: 69

Number of processes used: 18

The times for each task are: [0.129954, 0.10475, 0.101183, 0.105621, 0.128685, 0.104288, 0.13376

Minimum: 0.040771 Maximum: 0.196934 Average: 0.0865 seconds

Time taken for processing each image 7.4941

Length of task list: 69

Number of processes used: 18

The times for each task are: [0.119313, 0.106921, 0.104883, 0.107807, 0.118192, 0.130283, 0.0917

Minimum: 0.039353 Maximum: 0.145902 Average: 0.0863 seconds

Time taken for processing each image 7.3678

Length of task list: 69

Number of processes used: 18

The times for each task are: [0.144774, 0.094974, 0.124036, 0.102996, 0.11466, 0.131426, 0.10810

Minimum: 0.039221 Maximum: 0.155794 Average: 0.0849 seconds

Time taken for processing each image 7.4766

Window sizes used: [1.15, 1.35, 1.55]

18 processes used for testing 3 window sizes

Processing times for each image [7.6839, 7.7419, 7.29, 7.4941, 7.3678, 7.4766] with an average of 7.5000.

Time elapsed so far... 844.742999999999

#####
#####

Number of processes used: 19 window size 170

Length of task list: 69

Number of processes used: 19

The times for each task are: [0.115412, 0.087838, 0.117772, 0.133398, 0.094721, 0.105294, 0.142711]

Minimum: 0.038641 Maximum: 0.154352 Average: 0.0865 seconds

Time taken for processing each image 7.9637

Length of task list: 69

Number of processes used: 19

The times for each task are: [0.13599, 0.110877, 0.105929, 0.14159, 0.10924, 0.175842, 0.108711,

Minimum: 0.038481 Maximum: 0.175842 Average: 0.0872 seconds

Time taken for processing each image 8.0049

Length of task list: 69

Number of processes used: 19

The times for each task are: [0.110513, 0.110017, 0.091066, 0.104463, 0.11664, 0.105176, 0.135049]

Minimum: 0.038302 Maximum: 0.143411 Average: 0.0837 seconds

Time taken for processing each image 8.0543

Length of task list: 69

Number of processes used: 19

The times for each task are: [0.123034 0.131156 0.099771 0.135404 0.116225 0.131497 0.1228

Minimum: 0.043761 Maximum: 0.169156 Average: 0.0876 seconds

Time taken for processing each image 7.8047

***** time taken for processing each image : 7.501

Length of task list: 69

Number of processes used: 19

The times for each task are: [0.126113, 0.113496, 0.110373, 0.133709, 0.120778, 0.114023, 0.1292

Minimum: 0.040464 Maximum: 0.201957 Average: 0.0897 seconds

Time taken for processing each image 7.7487

Length of task list: 69

Number of processes used: 19

The times for each task are: [0.131976, 0.112252, 0.10145, 0.134675, 0.107519, 0.136213, 0.13545

Minimum: 0.039797 Maximum: 0.140409 Average: 0.0853 seconds

Time taken for processing each image 7.607

Window sizes used: [1.15, 1.35, 1.55]

19 processes used for testing 3 window sizes

Processing times for each image [7.9637, 8.0049, 8.0543, 7.8047, 7.7487, 7.607] with an average

Time elapsed so far... 891.9263

#####

#####

Number of processes used: 20 window size 170

Length of task list: 69

Number of processes used: 20

The times for each task are: [0.125621, 0.091071, 0.118257, 0.115218, 0.114213, 0.130504, 0.1137

Minimum: 0.041685 Maximum: 0.152064 Average: 0.0882 seconds

Time taken for processing each image 8.0524

Length of task list: 69

Number of processes used: 20

The times for each task are: [0.147544, 0.106777, 0.172717, 0.097252, 0.135278, 0.135655, 0.1342

Minimum: 0.042208 Maximum: 0.172717 Average: 0.0918 seconds

Time taken for processing each image 7.9583

Length of task list: 69

Number of processes used: 20

The times for each task are: [0.140549, 0.120588, 0.091315, 0.145845, 0.094469, 0.089633, 0.143711]
Minimum: 0.036777 Maximum: 0.15405 Average: 0.0852 seconds

Time taken for processing each image 8.58

Length of task list: 69
Number of processes used: 21

The times for each task are: [0.114771, 0.126346, 0.136896, 0.127954, 0.118744, 0.107012, 0.144111]
Minimum: 0.036751 Maximum: 0.148356 Average: 0.0901 seconds

Time taken for processing each image 8.5757

Length of task list: 69
Number of processes used: 21

The times for each task are: [0.123097, 0.114752, 0.094422, 0.089643, 0.121171, 0.105889, 0.141311]
Minimum: 0.040073 Maximum: 0.154711 Average: 0.0876 seconds

Time taken for processing each image 8.4887

Length of task list: 69
Number of processes used: 21

The times for each task are: [0.115589, 0.08741, 0.08743, 0.118452, 0.143316, 0.145221, 0.109351]
Minimum: 0.04066 Maximum: 0.148358 Average: 0.0878 seconds

Time taken for processing each image 8.5294

Length of task list: 69
Number of processes used: 21

The times for each task are: [0.130682, 0.144252, 0.136896, 0.120589, 0.092169, 0.130305, 0.114711]
Minimum: 0.03932 Maximum: 0.144252 Average: 0.0867 seconds

Time taken for processing each image 8.2748

Length of task list: 69
Number of processes used: 21

The times for each task are: [0.113906, 0.102626, 0.088546, 0.101886, 0.097752, 0.134826, 0.114511]
Minimum: 0.042702 Maximum: 0.142252 Average: 0.0866 seconds

Time taken for processing each image 8.6088

Window sizes used: [1.15, 1.35, 1.55]

21 processes used for testing 3 window sizes

Processing times for each image [8.58, 8.5757, 8.4887, 8.5294, 8.2748, 8.6088] with an average of 8.5000

Time elapsed so far... 991.2428

#####

Number of processes used: 22 window size 170

Length of task list: 69

Number of processes used: 22

The times for each task are: [0.110208, 0.149308, 0.093504, 0.136899, 0.098281, 0.109788, 0.143881]

Minimum: 0.04101 Maximum: 0.160699 Average: 0.0894 seconds

Time taken for processing each image 8.7082

Length of task list: 69

Number of processes used: 22

The times for each task are: [0.128143, 0.114737, 0.088569, 0.138834, 0.111294, 0.106746, 0.141181]

Minimum: 0.040092 Maximum: 0.145126 Average: 0.0872 seconds

Time taken for processing each image 8.8586

Length of task list: 69

Number of processes used: 22

The times for each task are: [0.133399, 0.17157, 0.149095, 0.137644, 0.131108, 0.135697, 0.115701]

Minimum: 0.042204 Maximum: 0.17157 Average: 0.089 seconds

Time taken for processing each image 8.7785

Length of task list: 69

Number of processes used: 22

The times for each task are: [0.137525, 0.108505, 0.092221, 0.088817, 0.115547, 0.085353, 0.136681]

Minimum: 0.042062 Maximum: 0.15251 Average: 0.09 seconds

Time taken for processing each image 8.7585

* * * * *

Length of task list: 69

Min. : -0.038404 Max. : 0.168889 A : 0.0854

Time taken for processing each image 8.8362

Length of task list: 69

Number of processes used: 22

The times for each task are: [0.122395, 0.095675, 0.138988, 0.138638, 0.133594, 0.136356, 0.099611]

Minimum: 0.043337 Maximum: 0.158221 Average: 0.0888 seconds

Time taken for processing each image 8.697

Window sizes used: [1.15, 1.35, 1.55]

22 processes used for testing 3 window sizes

Processing times for each image [8.7082, 8.8586, 8.7785, 8.7585, 8.8362, 8.697] with an average

Time elapsed so far... 1043.8798

#####
#####

Number of processes used: 23 window size 170

Length of task list: 69

Number of processes used: 23

The times for each task are: [0.141663, 0.159067, 0.153059, 0.132121, 0.115369, 0.1309, 0.127698]

Minimum: 0.043412 Maximum: 0.165423 Average: 0.092 seconds

Time taken for processing each image 9.185

Length of task list: 69

Number of processes used: 23

The times for each task are: [0.139338, 0.134033, 0.124561, 0.118894, 0.102976, 0.126694, 0.114611]

Minimum: 0.042778 Maximum: 0.150568 Average: 0.0919 seconds

Time taken for processing each image 9.0544

```
*****
Length of task list: 69
Number of processes used: 23

The times for each task are: [0.116263, 0.091699, 0.139036, 0.14305, 0.110143, 0.128504, 0.11806

Minimum: 0.03917 Maximum: 0.156674 Average: 0.0867 seconds

Time taken for processing each image 9.1115
*****
Length of task list: 69
Number of processes used: 23

The times for each task are: [0.146803, 0.106705, 0.118118, 0.097782, 0.118727, 0.114228, 0.1331

Minimum: 0.038488 Maximum: 0.156549 Average: 0.0896 seconds

Time taken for processing each image 8.771
*****
Length of task list: 69
Number of processes used: 23

The times for each task are: [0.153138, 0.085385, 0.115988, 0.086628, 0.117222, 0.110324, 0.1315

Minimum: 0.0378 Maximum: 0.221091 Average: 0.0927 seconds

Time taken for processing each image 9.2234
*****
Length of task list: 69
Number of processes used: 23

The times for each task are: [0.137843, 0.087006, 0.126933, 0.152335, 0.124728, 0.108032, 0.1300

Minimum: 0.0409 Maximum: 0.152335 Average: 0.0897 seconds

Time taken for processing each image 8.8745
*****
Window sizes used: [1.15, 1.35, 1.55]
23 processes used for testing 3 window sizes
Processing times for each image [9.185, 9.0544, 9.1115, 8.771, 9.2234, 8.8745] with an average o

Time elapsed so far... 1098.0996
#####
##### Number of processes used: 24 window size 170
```

Length of task list: 69
Number of processes used: 24

The times for each task are: [0.117707, 0.100205, 0.108953, 0.16588, 0.123405, 0.120671, 0.13902

Minimum: 0.039092 Maximum: 0.16588 Average: 0.0903 seconds

Time taken for processing each image 9.4194

Length of task list: 69
Number of processes used: 24

The times for each task are: [0.126309, 0.089535, 0.121548, 0.112595, 0.125658, 0.117542, 0.1168

Minimum: 0.041401 Maximum: 0.141879 Average: 0.0899 seconds

Time taken for processing each image 9.5844

Length of task list: 69
Number of processes used: 24

The times for each task are: [0.128576, 0.114597, 0.109558, 0.137364, 0.100308, 0.138877, 0.1213

Minimum: 0.042293 Maximum: 0.277696 Average: 0.0982 seconds

Time taken for processing each image 9.6506

Length of task list: 69
Number of processes used: 24

The times for each task are: [0.140627, 0.109272, 0.09802, 0.115443, 0.112981, 0.125527, 0.12702

Minimum: 0.039845 Maximum: 0.141558 Average: 0.0898 seconds

Time taken for processing each image 9.3548

Length of task list: 69
Number of processes used: 24

The times for each task are: [0.145089, 0.106357, 0.11209, 0.15721, 0.113836, 0.129616, 0.114101

Minimum: 0.042947 Maximum: 0.15721 Average: 0.0904 seconds

Time taken for processing each image 9.5374

Length of task list: 69
Number of processes used: 24

The times for each task are: [0.124497, 0.111374, 0.086706, 0.111952, 0.116561, 0.12937, 0.13080

Minimum: 0.042855 Maximum: 0.153811 Average: 0.0874 seconds

Time taken for processing each image 9.386

Window sizes used: [1.15, 1.35, 1.55]

24 processes used for testing 3 window sizes

Processing times for each image [9.4194, 9.5844, 9.6506, 9.3548, 9.5374, 9.386] with an average

Time elapsed so far... 1155.0322

#####

#####

Number of processes used: 25 window size 170

Length of task list: 69

Number of processes used: 25

The times for each task are: [0.128037, 0.09003, 0.175706, 0.108796, 0.128337, 0.128943, 0.11641

Minimum: 0.042557 Maximum: 0.175706 Average: 0.091 seconds

Time taken for processing each image 9.5953

Length of task list: 69

Number of processes used: 25

The times for each task are: [0.113918, 0.100472, 0.127618, 0.106237, 0.147151, 0.14614, 0.12025

Minimum: 0.044328 Maximum: 0.183374 Average: 0.0909 seconds

Time taken for processing each image 9.8081

Length of task list: 69

Number of processes used: 25

The times for each task are: [0.121761, 0.130576, 0.107466, 0.132219, 0.096798, 0.133825, 0.1332

Minimum: 0.042912 Maximum: 0.166181 Average: 0.0919 seconds

Time taken for processing each image 9.7653

Length of task list: 69

Number of processes used: 25

The times for each task are: [0.109157, 0.089462, 0.088682, 0.115388, 0.114333, 0.085037, 0.1143

```
Minimum: 0.044162 Maximum: 0.140169 Average: 0.0884 seconds
```

```
Time taken for processing each image 9.7795
```

```
*****
```

```
Length of task list: 69
```

```
Number of processes used: 25
```

```
The times for each task are: [0.165567, 0.108894, 0.092553, 0.113887, 0.092683, 0.097699, 0.1295]
```

```
Minimum: 0.038073 Maximum: 0.165567 Average: 0.0908 seconds
```

```
Time taken for processing each image 9.8383
```

```
*****
```

```
Length of task list: 69
```

```
Number of processes used: 25
```

```
The times for each task are: [0.125659, 0.084269, 0.117396, 0.088296, 0.147664, 0.106838, 0.1449]
```

```
Minimum: 0.041682 Maximum: 0.147664 Average: 0.0888 seconds
```

```
Time taken for processing each image 9.8111
```

```
*****
```

```
Window sizes used: [1.15, 1.35, 1.55]
```

```
25 processes used for testing 3 window sizes
```

```
Processing times for each image [9.5953, 9.8081, 9.7653, 9.7795, 9.8383, 9.8111] with an average
```

```
Time elapsed so far... 1213.6298000000002
```

```
#####
```

```
#####
```

```
=====
```

```
In [123]: use_sub_windows = 0
use_default_number_of_processes = 0
print_find_car_step_times = 1
find_car_hog_parallelism_enabled = 0
step = 170
xwinstarts, xwinstops = calculate_patches(x_start, x_stop, window, scale, step, cells_
ywinstarts, ywinstops = calculate_patches(y_start, y_stop, window, scale, step, cells_
print_task_times = 1

time_elapsed = 0
```

```

for i in range(1,2):
    #totalTasks = len(scaleslist[i]) * len(xwinstarts) * len(ywinstarts)
    #print('Number of Tasks for scale and step', totalTasks)
    for j in range(1, len(scaleslist[i])+1):
        time_for_all_images = []
        test_images_with_boxes = []
        number_of_processes_used = j
        print('Number of processes used: ', number_of_processes_used)
        if use_sub_windows:
            print('Window size', step)
        print()
        for image in test_images:
            img = mpimg.imread(image)
            freeze_support()
            t0 = time.time()
            r = []
            r = parallelize_find_cars(img, number_of_processes_used, i)
            t1 = time.time()
            processing_time = round(t1-t0, 4)
            print('Time taken for processing each image', processing_time)
            time_for_all_images.append(processing_time)
            img_with_boxes = np.copy(img)
            for box in r:
                cv2.rectangle(img_with_boxes, box[0], box[1], (0,0,255), 2)
            #plt.figure(figsize=(16,9))
            #plt.imshow(img_with_boxes)
            test_images_with_boxes.append(img_with_boxes)
            print('*****')
        print('Window sizes used:', scaleslist[i])
        print(number_of_processes_used, 'processes used for testing', len(scaleslist[i]))
        print('Processing times for each image', time_for_all_images, \
              'with an average of', round(sum(time_for_all_images)/len(time_for_all_im
        print()
        time_elapsed += sum(time_for_all_images)
        print('Time elapsed so far...', time_elapsed)
        print('#####')
        print('#####')

        print()
        print()
        print('=====')
        print()
        print()

```

Number of processes used: 1

Length of task list: 3

Number of processes used: 1

```
Process-4615, Step 1: Divide with 255, processing time: 0.034127 seconds
Process-4615, Step 2: Resize if scale is not 1: 0.020404 seconds
Process-4615, Step 3: Get HOG channels: 0.000185 seconds
Process-4615, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4615, Step 5: Compute individual channel HOG features for the entire image: 0.474862 seconds
Process-4615, Step 6: Misc initializations: 0.0 seconds
Process-4615, Step 7: for loop: 0.032837 seconds
```

Process-4615All steps time: 0.562425 seconds

Process-4615, Find cars processing time: 0.562599 seconds

```
Process-4615, Step 1: Divide with 255, processing time: 0.011353 seconds
Process-4615, Step 2: Resize if scale is not 1: 0.002038 seconds
Process-4615, Step 3: Get HOG channels: 9e-06 seconds
Process-4615, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4615, Step 5: Compute individual channel HOG features for the entire image: 0.358485 seconds
Process-4615, Step 6: Misc initializations: 1e-06 seconds
Process-4615, Step 7: for loop: 0.022656 seconds
```

Process-4615All steps time: 0.39455199999999996 seconds

Process-4615, Find cars processing time: 0.394637 seconds

```
Process-4615, Step 1: Divide with 255, processing time: 0.008386 seconds
Process-4615, Step 2: Resize if scale is not 1: 0.006209 seconds
Process-4615, Step 3: Get HOG channels: 9e-06 seconds
Process-4615, Step 4: Define blocks and steps as above: 8e-06 seconds
Process-4615, Step 5: Compute individual channel HOG features for the entire image: 0.324449 seconds
Process-4615, Step 6: Misc initializations: 1e-06 seconds
Process-4615, Step 7: for loop: 0.015739 seconds
```

Process-4615All steps time: 0.354801 seconds

Process-4615, Find cars processing time: 0.354885 seconds

The times for each task are: [0.562599, 0.394637, 0.354885] with:

Minimum: 0.354885 Maximum: 0.562599 Average: 0.4374 seconds

Time taken for processing each image 3.0707

Length of task list: 3

Number of processes used: 1

```
Process-4616, Step 1: Divide with 255, processing time: 0.035485 seconds
Process-4616, Step 2: Resize if scale is not 1: 0.017155 seconds
```

Process-4616, Step 3: Get HOG channels: 9.7e-05 seconds
Process-4616, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-4616, Step 5: Compute individual channel HOG features for the entire image: 0.49181 seconds
Process-4616, Step 6: Misc initializations: 1e-06 seconds
Process-4616, Step 7: for loop: 0.033621 seconds

Process-4616All steps time: 0.5781780000000001 seconds

Process-4616, Find cars processing time: 0.578342 seconds

Process-4616, Step 1: Divide with 255, processing time: 0.011069 seconds
Process-4616, Step 2: Resize if scale is not 1: 0.00692 seconds
Process-4616, Step 3: Get HOG channels: 1e-05 seconds
Process-4616, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4616, Step 5: Compute individual channel HOG features for the entire image: 0.382992 seconds
Process-4616, Step 6: Misc initializations: 0.0 seconds
Process-4616, Step 7: for loop: 0.021735 seconds

Process-4616All steps time: 0.422736 seconds

Process-4616, Find cars processing time: 0.422822 seconds

Process-4616, Step 1: Divide with 255, processing time: 0.007986 seconds
Process-4616, Step 2: Resize if scale is not 1: 0.005766 seconds
Process-4616, Step 3: Get HOG channels: 7e-06 seconds
Process-4616, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-4616, Step 5: Compute individual channel HOG features for the entire image: 0.257078 seconds
Process-4616, Step 6: Misc initializations: 1e-06 seconds
Process-4616, Step 7: for loop: 0.017521 seconds

Process-4616All steps time: 0.2883679999999996 seconds

Process-4616, Find cars processing time: 0.288445 seconds

The times for each task are: [0.578342, 0.422822, 0.288445] with:

Minimum: 0.288445 Maximum: 0.578342 Average: 0.4299 seconds

Time taken for processing each image 2.2799

Length of task list: 3

Number of processes used: 1

Process-4617, Step 1: Divide with 255, processing time: 0.029422 seconds
Process-4617, Step 2: Resize if scale is not 1: 0.020034 seconds
Process-4617, Step 3: Get HOG channels: 0.000164 seconds
Process-4617, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-4617, Step 5: Compute individual channel HOG features for the entire image: 0.615164 seconds

Process-4617, Step 6: Misc initializations: 1e-06 seconds
Process-4617, Step 7: for loop: 0.041757 seconds

Process-4617All steps time: 0.7065540000000001 seconds

Process-4617, Find cars processing time: 0.706741 seconds

Process-4617, Step 1: Divide with 255, processing time: 0.008807 seconds
Process-4617, Step 2: Resize if scale is not 1: 0.004692 seconds
Process-4617, Step 3: Get HOG channels: 1.2e-05 seconds
Process-4617, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-4617, Step 5: Compute individual channel HOG features for the entire image: 0.410804 seconds
Process-4617, Step 6: Misc initializations: 1e-06 seconds
Process-4617, Step 7: for loop: 0.025336 seconds

Process-4617All steps time: 0.449663 seconds

Process-4617, Find cars processing time: 0.449767 seconds

Process-4617, Step 1: Divide with 255, processing time: 0.016529 seconds
Process-4617, Step 2: Resize if scale is not 1: 0.005176 seconds
Process-4617, Step 3: Get HOG channels: 9e-06 seconds
Process-4617, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4617, Step 5: Compute individual channel HOG features for the entire image: 0.303041 seconds
Process-4617, Step 6: Misc initializations: 1e-06 seconds
Process-4617, Step 7: for loop: 0.016314 seconds

Process-4617All steps time: 0.34108 seconds

Process-4617, Find cars processing time: 0.341165 seconds

The times for each task are: [0.706741, 0.449767, 0.341165] with:

Minimum: 0.341165 Maximum: 0.706741 Average: 0.4992 seconds

Time taken for processing each image 2.9635

Length of task list: 3

Number of processes used: 1

Process-4618, Step 1: Divide with 255, processing time: 0.033648 seconds
Process-4618, Step 2: Resize if scale is not 1: 0.018745 seconds
Process-4618, Step 3: Get HOG channels: 0.000217 seconds
Process-4618, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4618, Step 5: Compute individual channel HOG features for the entire image: 0.470148 seconds
Process-4618, Step 6: Misc initializations: 1e-06 seconds
Process-4618, Step 7: for loop: 0.033699 seconds

Process-4618All steps time: 0.5564680000000001 seconds

Process-4618, Find cars processing time: 0.556628 seconds

Process-4618, Step 1: Divide with 255, processing time: 0.007459 seconds

Process-4618, Step 2: Resize if scale is not 1: 0.001771 seconds

Process-4618, Step 3: Get HOG channels: 9e-06 seconds

Process-4618, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-4618, Step 5: Compute individual channel HOG features for the entire image: 0.396045 seconds

Process-4618, Step 6: Misc initializations: 1e-06 seconds

Process-4618, Step 7: for loop: 0.024014 seconds

Process-4618All steps time: 0.4293079999999999 seconds

Process-4618, Find cars processing time: 0.429397 seconds

Process-4618, Step 1: Divide with 255, processing time: 0.015482 seconds

Process-4618, Step 2: Resize if scale is not 1: 0.006115 seconds

Process-4618, Step 3: Get HOG channels: 7e-06 seconds

Process-4618, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-4618, Step 5: Compute individual channel HOG features for the entire image: 0.259049 seconds

Process-4618, Step 6: Misc initializations: 1e-06 seconds

Process-4618, Step 7: for loop: 0.015862 seconds

Process-4618All steps time: 0.2965239999999995 seconds

Process-4618, Find cars processing time: 0.296597 seconds

The times for each task are: [0.556628, 0.429397, 0.296597] with:

Minimum: 0.296597 Maximum: 0.556628 Average: 0.4275 seconds

Time taken for processing each image 2.3071

Length of task list: 3

Number of processes used: 1

Process-4619, Step 1: Divide with 255, processing time: 0.03109 seconds

Process-4619, Step 2: Resize if scale is not 1: 0.019038 seconds

Process-4619, Step 3: Get HOG channels: 0.000113 seconds

Process-4619, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-4619, Step 5: Compute individual channel HOG features for the entire image: 0.537352 seconds

Process-4619, Step 6: Misc initializations: 1e-06 seconds

Process-4619, Step 7: for loop: 0.035332 seconds

Process-4619All steps time: 0.6229360000000002 seconds

Process-4619, Find cars processing time: 0.623096 seconds

```
Process-4619, Step 1: Divide with 255, processing time: 0.009559 seconds
Process-4619, Step 2: Resize if scale is not 1: 0.007803 seconds
Process-4619, Step 3: Get HOG channels: 9e-06 seconds
Process-4619, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4619, Step 5: Compute individual channel HOG features for the entire image: 0.350704 seconds
Process-4619, Step 6: Misc initializations: 1e-06 seconds
Process-4619, Step 7: for loop: 0.024225 seconds
```

Process-4619All steps time: 0.39231099999999997 seconds

Process-4619, Find cars processing time: 0.392404 seconds

```
Process-4619, Step 1: Divide with 255, processing time: 0.015297 seconds
Process-4619, Step 2: Resize if scale is not 1: 0.005523 seconds
Process-4619, Step 3: Get HOG channels: 1.5e-05 seconds
Process-4619, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4619, Step 5: Compute individual channel HOG features for the entire image: 0.276328 seconds
Process-4619, Step 6: Misc initializations: 0.0 seconds
Process-4619, Step 7: for loop: 0.015254 seconds
```

Process-4619All steps time: 0.312427 seconds

Process-4619, Find cars processing time: 0.312523 seconds

The times for each task are: [0.623096, 0.392404, 0.312523] with:

Minimum: 0.312523 Maximum: 0.623096 Average: 0.4427 seconds

Time taken for processing each image 2.1605

Length of task list: 3

Number of processes used: 1

```
Process-4620, Step 1: Divide with 255, processing time: 0.034218 seconds
Process-4620, Step 2: Resize if scale is not 1: 0.018974 seconds
Process-4620, Step 3: Get HOG channels: 0.000143 seconds
Process-4620, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4620, Step 5: Compute individual channel HOG features for the entire image: 0.66504 seconds
Process-4620, Step 6: Misc initializations: 0.0 seconds
Process-4620, Step 7: for loop: 0.035891 seconds
```

Process-4620All steps time: 0.754276 seconds

Process-4620, Find cars processing time: 0.754447 seconds

```
Process-4620, Step 1: Divide with 255, processing time: 0.01489 seconds
Process-4620, Step 2: Resize if scale is not 1: 0.009209 seconds
```

Process-4620, Step 3: Get HOG channels: 1.3e-05 seconds
Process-4620, Step 4: Define blocks and steps as above: 1.1e-05 seconds
Process-4620, Step 5: Compute individual channel HOG features for the entire image: 0.402687 seconds
Process-4620, Step 6: Misc initializations: 0.0 seconds
Process-4620, Step 7: for loop: 0.028709 seconds

Process-4620All steps time: 0.455519 seconds

Process-4620, Find cars processing time: 0.455628 seconds

Process-4620, Step 1: Divide with 255, processing time: 0.014768 seconds
Process-4620, Step 2: Resize if scale is not 1: 0.005193 seconds
Process-4620, Step 3: Get HOG channels: 7e-06 seconds
Process-4620, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-4620, Step 5: Compute individual channel HOG features for the entire image: 0.25601 seconds
Process-4620, Step 6: Misc initializations: 1e-06 seconds
Process-4620, Step 7: for loop: 0.015307 seconds

Process-4620All steps time: 0.291295 seconds

Process-4620, Find cars processing time: 0.291365 seconds

The times for each task are: [0.754447, 0.455628, 0.291365] with:

Minimum: 0.291365 Maximum: 0.754447 Average: 0.5005 seconds

Time taken for processing each image 2.5828

Window sizes used: [1.15, 1.35, 1.55]

1 processes used for testing 3 window sizes

Processing times for each image [3.0707, 2.2799, 2.9635, 2.3071, 2.1605, 2.5828] with an average

Time elapsed so far... 15.364500000000001

#####

#####

Number of processes used: 2

Length of task list: 3

Number of processes used: 2

Process-4621, Step 1: Divide with 255, processing time: 0.032706 seconds
Process-4621, Step 2: Resize if scale is not 1: 0.018737 seconds
Process-4621, Step 3: Get HOG channels: 0.000115 seconds
Process-4621, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4621, Step 5: Compute individual channel HOG features for the entire image: 0.491591 seconds
Process-4621, Step 6: Misc initializations: 1e-06 seconds

Process-4621, Step 7: for loop: 0.035055 seconds

Process-4621All steps time: 0.5782149999999999 seconds

Process-4621, Find cars processing time: 0.578414 seconds

Process-4622, Step 1: Divide with 255, processing time: 0.035873 seconds

Process-4622, Step 2: Resize if scale is not 1: 0.015874 seconds

Process-4622, Step 3: Get HOG channels: 3e-05 seconds

Process-4622, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-4622, Step 5: Compute individual channel HOG features for the entire image: 0.369552 seconds

Process-4622, Step 6: Misc initializations: 1e-06 seconds

Process-4622, Step 7: for loop: 0.024723 seconds

Process-4622All steps time: 0.4460599999999996 seconds

Process-4622, Find cars processing time: 0.446227 seconds

Process-4621, Step 1: Divide with 255, processing time: 0.010311 seconds

Process-4621, Step 2: Resize if scale is not 1: 0.006177 seconds

Process-4621, Step 3: Get HOG channels: 1e-05 seconds

Process-4621, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-4621, Step 5: Compute individual channel HOG features for the entire image: 0.366089 seconds

Process-4621, Step 6: Misc initializations: 1e-06 seconds

Process-4621, Step 7: for loop: 0.017519 seconds

Process-4621All steps time: 0.400117 seconds

Process-4621, Find cars processing time: 0.400202 seconds

The times for each task are: [0.578414, 0.446227, 0.400202] with:

Minimum: 0.400202 Maximum: 0.578414 Average: 0.4749 seconds

Time taken for processing each image 2.0941

Length of task list: 3

Number of processes used: 2

Process-4623, Step 1: Divide with 255, processing time: 0.026529 seconds

Process-4623, Step 2: Resize if scale is not 1: 0.021606 seconds

Process-4623, Step 3: Get HOG channels: 7.4e-05 seconds

Process-4623, Step 4: Define blocks and steps as above: 1.1e-05 seconds

Process-4623, Step 5: Compute individual channel HOG features for the entire image: 0.486347 seconds

Process-4623, Step 6: Misc initializations: 1e-06 seconds

Process-4623, Step 7: for loop: 0.040638 seconds

Process-4623All steps time: 0.575206 seconds

Process-4623, Find cars processing time: 0.575458 seconds

Process-4624, Step 1: Divide with 255, processing time: 0.028034 seconds

Process-4624, Step 2: Resize if scale is not 1: 0.01492 seconds

Process-4624, Step 3: Get HOG channels: 4.8e-05 seconds

Process-4624, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-4624, Step 5: Compute individual channel HOG features for the entire image: 0.365314 seconds

Process-4624, Step 6: Misc initializations: 1e-06 seconds

Process-4624, Step 7: for loop: 0.026861 seconds

Process-4624All steps time: 0.43518300000000004 seconds

Process-4624, Find cars processing time: 0.435293 seconds

Process-4623, Step 1: Divide with 255, processing time: 0.010497 seconds

Process-4623, Step 2: Resize if scale is not 1: 0.005754 seconds

Process-4623, Step 3: Get HOG channels: 7e-06 seconds

Process-4623, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-4623, Step 5: Compute individual channel HOG features for the entire image: 0.271066 seconds

Process-4623, Step 6: Misc initializations: 1e-06 seconds

Process-4623, Step 7: for loop: 0.016221 seconds

Process-4623All steps time: 0.3035529999999999 seconds

Process-4623, Find cars processing time: 0.303629 seconds

The times for each task are: [0.575458, 0.435293, 0.303629] with:

Minimum: 0.303629 Maximum: 0.575458 Average: 0.4381 seconds

Time taken for processing each image 1.6376

Length of task list: 3

Number of processes used: 2

Process-4625, Step 1: Divide with 255, processing time: 0.028717 seconds

Process-4625, Step 2: Resize if scale is not 1: 0.018657 seconds

Process-4625, Step 3: Get HOG channels: 9e-05 seconds

Process-4625, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-4625, Step 5: Compute individual channel HOG features for the entire image: 0.494723 seconds

Process-4625, Step 6: Misc initializations: 1e-06 seconds

Process-4625, Step 7: for loop: 0.036093 seconds

Process-4625All steps time: 0.5782900000000001 seconds

Process-4625, Find cars processing time: 0.578475 seconds

Process-4626, Step 1: Divide with 255, processing time: 0.028902 seconds
Process-4626, Step 2: Resize if scale is not 1: 0.01572 seconds
Process-4626, Step 3: Get HOG channels: 2.8e-05 seconds
Process-4626, Step 4: Define blocks and steps as above: 5e-06 seconds
Process-4626, Step 5: Compute individual channel HOG features for the entire image: 0.356578 seconds
Process-4626, Step 6: Misc initializations: 1e-06 seconds
Process-4626, Step 7: for loop: 0.023996 seconds

Process-4626All steps time: 0.42523 seconds

Process-4626, Find cars processing time: 0.425381 seconds

Process-4625, Step 1: Divide with 255, processing time: 0.00952 seconds
Process-4625, Step 2: Resize if scale is not 1: 0.007891 seconds
Process-4625, Step 3: Get HOG channels: 8e-06 seconds
Process-4625, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4625, Step 5: Compute individual channel HOG features for the entire image: 0.306687 seconds
Process-4625, Step 6: Misc initializations: 1e-06 seconds
Process-4625, Step 7: for loop: 0.022972 seconds

Process-4625All steps time: 0.347089 seconds

Process-4625, Find cars processing time: 0.347171 seconds

The times for each task are: [0.578475, 0.425381, 0.347171] with:

Minimum: 0.347171 Maximum: 0.578475 Average: 0.4503 seconds

Time taken for processing each image 2.1381

Length of task list: 3

Number of processes used: 2

Process-4627, Step 1: Divide with 255, processing time: 0.032429 seconds
Process-4627, Step 2: Resize if scale is not 1: 0.020264 seconds
Process-4627, Step 3: Get HOG channels: 7.8e-05 seconds
Process-4627, Step 4: Define blocks and steps as above: 1e-05 seconds
Process-4627, Step 5: Compute individual channel HOG features for the entire image: 0.502278 seconds
Process-4627, Step 6: Misc initializations: 1e-06 seconds
Process-4627, Step 7: for loop: 0.034251 seconds

Process-4627All steps time: 0.589311 seconds

Process-4627, Find cars processing time: 0.589488 seconds

Process-4628, Step 1: Divide with 255, processing time: 0.029968 seconds
Process-4628, Step 2: Resize if scale is not 1: 0.01795 seconds
Process-4628, Step 3: Get HOG channels: 4e-05 seconds

Process-4628, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-4628, Step 5: Compute individual channel HOG features for the entire image: 0.413078 seconds

Process-4628, Step 6: Misc initializations: 1e-06 seconds

Process-4628, Step 7: for loop: 0.025242 seconds

Process-4628All steps time: 0.4862839999999994 seconds

Process-4628, Find cars processing time: 0.486403 seconds

Process-4627, Step 1: Divide with 255, processing time: 0.008814 seconds

Process-4627, Step 2: Resize if scale is not 1: 0.004602 seconds

Process-4627, Step 3: Get HOG channels: 7e-06 seconds

Process-4627, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-4627, Step 5: Compute individual channel HOG features for the entire image: 0.271182 seconds

Process-4627, Step 6: Misc initializations: 1e-06 seconds

Process-4627, Step 7: for loop: 0.016913 seconds

Process-4627All steps time: 0.3015259999999996 seconds

Process-4627, Find cars processing time: 0.301599 seconds

The times for each task are: [0.589488, 0.486403, 0.301599] with:

Minimum: 0.301599 Maximum: 0.589488 Average: 0.4592 seconds

Time taken for processing each image 1.5144

Length of task list: 3

Number of processes used: 2

Process-4630, Step 1: Divide with 255, processing time: 0.031539 seconds

Process-4630, Step 2: Resize if scale is not 1: 0.016227 seconds

Process-4630, Step 3: Get HOG channels: 4.7e-05 seconds

Process-4630, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-4630, Step 5: Compute individual channel HOG features for the entire image: 0.354034 seconds

Process-4630, Step 6: Misc initializations: 1e-06 seconds

Process-4630, Step 7: for loop: 0.024257 seconds

Process-4630All steps time: 0.4261119999999994 seconds

Process-4630, Find cars processing time: 0.426275 seconds

Process-4629, Step 1: Divide with 255, processing time: 0.033925 seconds

Process-4629, Step 2: Resize if scale is not 1: 0.019572 seconds

Process-4629, Step 3: Get HOG channels: 6.3e-05 seconds

Process-4629, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-4629, Step 5: Compute individual channel HOG features for the entire image: 0.588662 seconds

Process-4629, Step 6: Misc initializations: 1e-06 seconds

Process-4629, Step 7: for loop: 0.036398 seconds

Process-4629All steps time: 0.6786310000000001 seconds

Process-4629, Find cars processing time: 0.678791 seconds

Process-4630, Step 1: Divide with 255, processing time: 0.009631 seconds

Process-4630, Step 2: Resize if scale is not 1: 0.005613 seconds

Process-4630, Step 3: Get HOG channels: 9e-06 seconds

Process-4630, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-4630, Step 5: Compute individual channel HOG features for the entire image: 0.246387 seconds

Process-4630, Step 6: Misc initializations: 0.0 seconds

Process-4630, Step 7: for loop: 0.015243 seconds

Process-4630All steps time: 0.276893 seconds

Process-4630, Find cars processing time: 0.276973 seconds

The times for each task are: [0.426275, 0.678791, 0.276973] with:

Minimum: 0.276973 Maximum: 0.678791 Average: 0.4607 seconds

Time taken for processing each image 1.9399

Length of task list: 3

Number of processes used: 2

Process-4632, Step 1: Divide with 255, processing time: 0.028831 seconds

Process-4632, Step 2: Resize if scale is not 1: 0.01648 seconds

Process-4632, Step 3: Get HOG channels: 3.1e-05 seconds

Process-4632, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-4632, Step 5: Compute individual channel HOG features for the entire image: 0.471941 seconds

Process-4632, Step 6: Misc initializations: 1e-06 seconds

Process-4632, Step 7: for loop: 0.030593 seconds

Process-4632All steps time: 0.547884 seconds

Process-4632, Find cars processing time: 0.548039 seconds

Process-4631, Step 1: Divide with 255, processing time: 0.053271 seconds

Process-4631, Step 2: Resize if scale is not 1: 0.02216 seconds

Process-4631, Step 3: Get HOG channels: 0.000143 seconds

Process-4631, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-4631, Step 5: Compute individual channel HOG features for the entire image: 0.567848 seconds

Process-4631, Step 6: Misc initializations: 1e-06 seconds

Process-4631, Step 7: for loop: 0.03778 seconds

Process-4631All steps time: 0.6812130000000001 seconds

Process-4631, Find cars processing time: 0.681413 seconds

Process-4632, Step 1: Divide with 255, processing time: 0.010924 seconds

Process-4632, Step 2: Resize if scale is not 1: 0.005719 seconds

Process-4632, Step 3: Get HOG channels: 9e-06 seconds

Process-4632, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-4632, Step 5: Compute individual channel HOG features for the entire image: 0.325895 seconds

Process-4632, Step 6: Misc initializations: 1e-06 seconds

Process-4632, Step 7: for loop: 0.028307 seconds

Process-4632All steps time: 0.37086399999999997 seconds

Process-4632, Find cars processing time: 0.370959 seconds

The times for each task are: [0.548039, 0.681413, 0.370959] with:

Minimum: 0.370959 Maximum: 0.681413 Average: 0.5335 seconds

Time taken for processing each image 2.0619

Window sizes used: [1.15, 1.35, 1.55]

2 processes used for testing 3 window sizes

Processing times for each image [2.0941, 1.6376, 2.1381, 1.5144, 1.9399, 2.0619] with an average

Time elapsed so far... 26.750500000000002

#####

#####

Number of processes used: 3

Length of task list: 3

Number of processes used: 3

Process-4634, Step 1: Divide with 255, processing time: 0.02824 seconds

Process-4634, Step 2: Resize if scale is not 1: 0.016095 seconds

Process-4634, Step 3: Get HOG channels: 2.5e-05 seconds

Process-4634, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-4634, Step 5: Compute individual channel HOG features for the entire image: 0.411193 seconds

Process-4634, Step 6: Misc initializations: 1e-06 seconds

Process-4634, Step 7: for loop: 0.025537 seconds

Process-4634All steps time: 0.4810979999999999 seconds

Process-4634, Find cars processing time: 0.481233 seconds

Process-4633, Step 1: Divide with 255, processing time: 0.028754 seconds

```
Process-4633, Step 2: Resize if scale is not 1: 0.019131 seconds
Process-4633, Step 3: Get HOG channels: 0.000116 seconds
Process-4633, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-4633, Step 5: Compute individual channel HOG features for the entire image: 0.636523 seconds
Process-4633, Step 6: Misc initializations: 1e-06 seconds
Process-4633, Step 7: for loop: 0.036681 seconds
```

Process-4633All steps time: 0.7212189999999999 seconds

Process-4633, Find cars processing time: 0.721414 seconds

```
Process-4635, Step 1: Divide with 255, processing time: 0.03048 seconds
Process-4635, Step 2: Resize if scale is not 1: 0.016702 seconds
Process-4635, Step 3: Get HOG channels: 3.3e-05 seconds
Process-4635, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-4635, Step 5: Compute individual channel HOG features for the entire image: 0.364702 seconds
Process-4635, Step 6: Misc initializations: 2e-06 seconds
Process-4635, Step 7: for loop: 0.022616 seconds
```

Process-4635All steps time: 0.43454400000000004 seconds

Process-4635, Find cars processing time: 0.43469 seconds

The times for each task are: [0.481233, 0.721414, 0.43469] with:

Minimum: 0.43469 Maximum: 0.721414 Average: 0.5458 seconds

Time taken for processing each image 1.7266

```
*****
Length of task list: 3
Number of processes used: 3
```

```
Process-4636, Step 1: Divide with 255, processing time: 0.027853 seconds
Process-4636, Step 2: Resize if scale is not 1: 0.020198 seconds
Process-4636, Step 3: Get HOG channels: 7.6e-05 seconds
Process-4636, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-4636, Step 5: Compute individual channel HOG features for the entire image: 0.594063 seconds
Process-4636, Step 6: Misc initializations: 0.0 seconds
Process-4636, Step 7: for loop: 0.036572 seconds
```

Process-4636All steps time: 0.678771 seconds

Process-4636, Find cars processing time: 0.678994 seconds

```
Process-4637, Step 1: Divide with 255, processing time: 0.030588 seconds
Process-4637, Step 2: Resize if scale is not 1: 0.016468 seconds
Process-4637, Step 3: Get HOG channels: 2.2e-05 seconds
Process-4637, Step 4: Define blocks and steps as above: 6e-06 seconds
```

Process-4637, Step 5: Compute individual channel HOG features for the entire image: 0.416347 seconds
Process-4637, Step 6: Misc initializations: 1e-06 seconds
Process-4637, Step 7: for loop: 0.035326 seconds

Process-4637All steps time: 0.49875800000000003 seconds

Process-4637, Find cars processing time: 0.498903 seconds

Process-4638, Step 1: Divide with 255, processing time: 0.028667 seconds
Process-4638, Step 2: Resize if scale is not 1: 0.015297 seconds
Process-4638, Step 3: Get HOG channels: 4.5e-05 seconds
Process-4638, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-4638, Step 5: Compute individual channel HOG features for the entire image: 0.344201 seconds
Process-4638, Step 6: Misc initializations: 0.0 seconds
Process-4638, Step 7: for loop: 0.020766 seconds

Process-4638All steps time: 0.408983 seconds

Process-4638, Find cars processing time: 0.409132 seconds

The times for each task are: [0.678994, 0.498903, 0.409132] with:

Minimum: 0.409132 Maximum: 0.678994 Average: 0.529 seconds

Time taken for processing each image 2.0087

Length of task list: 3

Number of processes used: 3

Process-4640, Step 1: Divide with 255, processing time: 0.026939 seconds
Process-4640, Step 2: Resize if scale is not 1: 0.016303 seconds
Process-4640, Step 3: Get HOG channels: 3.3e-05 seconds
Process-4640, Step 4: Define blocks and steps as above: 7e-06 seconds
Process-4640, Step 5: Compute individual channel HOG features for the entire image: 0.447286 seconds
Process-4640, Step 6: Misc initializations: 1e-06 seconds
Process-4640, Step 7: for loop: 0.025188 seconds

Process-4640All steps time: 0.515757 seconds

Process-4640, Find cars processing time: 0.515885 seconds

Process-4639, Step 1: Divide with 255, processing time: 0.030753 seconds
Process-4639, Step 2: Resize if scale is not 1: 0.018808 seconds
Process-4639, Step 3: Get HOG channels: 8.3e-05 seconds
Process-4639, Step 4: Define blocks and steps as above: 1.2e-05 seconds
Process-4639, Step 5: Compute individual channel HOG features for the entire image: 0.62811 seconds
Process-4639, Step 6: Misc initializations: 1e-06 seconds
Process-4639, Step 7: for loop: 0.042922 seconds

Process-4639All steps time: 0.720689 seconds

Process-4639, Find cars processing time: 0.720913 seconds

Process-4641, Step 1: Divide with 255, processing time: 0.026497 seconds

Process-4641, Step 2: Resize if scale is not 1: 0.013593 seconds

Process-4641, Step 3: Get HOG channels: 4.6e-05 seconds

Process-4641, Step 4: Define blocks and steps as above: 5e-06 seconds

Process-4641, Step 5: Compute individual channel HOG features for the entire image: 0.323092 seconds

Process-4641, Step 6: Misc initializations: 1e-06 seconds

Process-4641, Step 7: for loop: 0.018098 seconds

Process-4641All steps time: 0.3813319999999995 seconds

Process-4641, Find cars processing time: 0.381453 seconds

The times for each task are: [0.515885, 0.720913, 0.381453] with:

Minimum: 0.381453 Maximum: 0.720913 Average: 0.5394 seconds

Time taken for processing each image 1.5659

Length of task list: 3

Number of processes used: 3

Process-4642, Step 1: Divide with 255, processing time: 0.026791 seconds

Process-4642, Step 2: Resize if scale is not 1: 0.018426 seconds

Process-4642, Step 3: Get HOG channels: 9.7e-05 seconds

Process-4642, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-4642, Step 5: Compute individual channel HOG features for the entire image: 0.558354 seconds

Process-4642, Step 6: Misc initializations: 1e-06 seconds

Process-4642, Step 7: for loop: 0.037089 seconds

Process-4642All steps time: 0.6407670000000001 seconds

Process-4642, Find cars processing time: 0.640953 seconds

Process-4644, Step 1: Divide with 255, processing time: 0.027981 seconds

Process-4644, Step 2: Resize if scale is not 1: 0.015347 seconds

Process-4644, Step 3: Get HOG channels: 7.8e-05 seconds

Process-4644, Step 4: Define blocks and steps as above: 8e-06 seconds

Process-4644, Step 5: Compute individual channel HOG features for the entire image: 0.355119 seconds

Process-4644, Step 6: Misc initializations: 1e-06 seconds

Process-4644, Step 7: for loop: 0.019284 seconds

Process-4644All steps time: 0.417818 seconds

Process-4644, Find cars processing time: 0.41796 seconds

Process-4643, Step 1: Divide with 255, processing time: 0.028371 seconds

Process-4643, Step 2: Resize if scale is not 1: 0.013805 seconds

Process-4643, Step 3: Get HOG channels: 5.2e-05 seconds

Process-4643, Step 4: Define blocks and steps as above: 9e-06 seconds

Process-4643, Step 5: Compute individual channel HOG features for the entire image: 0.394786 seconds

Process-4643, Step 6: Misc initializations: 0.0 seconds

Process-4643, Step 7: for loop: 0.023863 seconds

Process-4643All steps time: 0.4608860000000001 seconds

Process-4643, Find cars processing time: 0.461023 seconds

The times for each task are: [0.640953, 0.41796, 0.461023] with:

Minimum: 0.41796 Maximum: 0.640953 Average: 0.5066 seconds

Time taken for processing each image 1.5836

Length of task list: 3

Number of processes used: 3

Process-4646, Step 1: Divide with 255, processing time: 0.039928 seconds

Process-4646, Step 2: Resize if scale is not 1: 0.015033 seconds

Process-4646, Step 3: Get HOG channels: 3.2e-05 seconds

Process-4646, Step 4: Define blocks and steps as above: 2.2e-05 seconds

Process-4646, Step 5: Compute individual channel HOG features for the entire image: 0.416888 seconds

Process-4646, Step 6: Misc initializations: 1e-06 seconds

Process-4646, Step 7: for loop: 0.025552 seconds

Process-4646All steps time: 0.4974559999999995 seconds

Process-4646, Find cars processing time: 0.497592 seconds

Process-4645, Step 1: Divide with 255, processing time: 0.047373 seconds

Process-4647, Step 1: Divide with 255, processing time: 0.024941 seconds

Process-4645, Step 2: Resize if scale is not 1: 0.020079 seconds

Process-4647, Step 2: Resize if scale is not 1: 0.015635 seconds

Process-4645, Step 3: Get HOG channels: 0.000182 seconds

Process-4647, Step 3: Get HOG channels: 3.4e-05 seconds

Process-4645, Step 4: Define blocks and steps as above: 1e-05 seconds

Process-4647, Step 4: Define blocks and steps as above: 7e-06 seconds

Process-4645, Step 5: Compute individual channel HOG features for the entire image: 0.629699 seconds

Process-4647, Step 5: Compute individual channel HOG features for the entire image: 0.319822 seconds

Process-4645, Step 6: Misc initializations: 1e-06 seconds

Process-4647, Step 6: Misc initializations: 0.0 seconds

Process-4645, Step 7: for loop: 0.036507 seconds

Process-4647, Step 7: for loop: 0.017661 seconds
Process-4645All steps time: 0.733851 seconds

Process-4647All steps time: 0.3781 seconds
Process-4645, Find cars processing time: 0.734007 seconds

Process-4647, Find cars processing time: 0.378217 seconds

The times for each task are: [0.497592, 0.734007, 0.378217] with:

Minimum: 0.378217 Maximum: 0.734007 Average: 0.5366 seconds

Time taken for processing each image 1.5429

Length of task list: 3
Number of processes used: 3

Process-4648, Step 1: Divide with 255, processing time: 0.024625 seconds
Process-4648, Step 2: Resize if scale is not 1: 0.019143 seconds
Process-4648, Step 3: Get HOG channels: 8.5e-05 seconds
Process-4648, Step 4: Define blocks and steps as above: 1.3e-05 seconds
Process-4648, Step 5: Compute individual channel HOG features for the entire image: 0.623784 seconds
Process-4648, Step 6: Misc initializations: 0.0 seconds
Process-4648, Step 7: for loop: 0.034551 seconds

Process-4648All steps time: 0.702201 seconds

Process-4648, Find cars processing time: 0.702389 seconds

Process-4649, Step 1: Divide with 255, processing time: 0.024373 seconds
Process-4649, Step 2: Resize if scale is not 1: 0.016268 seconds
Process-4649, Step 3: Get HOG channels: 3.6e-05 seconds
Process-4649, Step 4: Define blocks and steps as above: 9e-06 seconds
Process-4649, Step 5: Compute individual channel HOG features for the entire image: 0.388427 seconds
Process-4649, Step 6: Misc initializations: 1e-06 seconds
Process-4649, Step 7: for loop: 0.024816 seconds

Process-4649All steps time: 0.45393 seconds

Process-4649, Find cars processing time: 0.45408 seconds

Process-4650, Step 1: Divide with 255, processing time: 0.025779 seconds
Process-4650, Step 2: Resize if scale is not 1: 0.015137 seconds
Process-4650, Step 3: Get HOG channels: 5.1e-05 seconds
Process-4650, Step 4: Define blocks and steps as above: 1.7e-05 seconds

```
Process-4650, Step 5: Compute individual channel HOG features for the entire image: 0.35707 seconds
Process-4650, Step 6: Misc initializations: 0.0 seconds
Process-4650, Step 7: for loop: 0.021793 seconds

Process-4650All steps time: 0.419847 seconds

Process-4650, Find cars processing time: 0.420003 seconds

The times for each task are: [0.702389, 0.45408, 0.420003] with:

Minimum: 0.420003 Maximum: 0.702389 Average: 0.5255 seconds

Time taken for processing each image 1.9048
*****
Window sizes used: [1.15, 1.35, 1.55]
3 processes used for testing 3 window sizes
Processing times for each image [1.7266, 2.0087, 1.5659, 1.5836, 1.5429, 1.9048] with an average

Time elapsed so far... 37.083
#####
#####
```

```
In [124]: def centroid(p1, p2):
    return (p1[0] + int(abs(p1[0] - p2[0])/2), p1[1] + int(abs(p1[1] - p2[1])/2))

def distance(p1, p2):
    #print('P1', p1)
    #print('P2', p2)
    return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)

class Car:
    totalCars = 0

    def __init__(self, carid, carname):
        self.detected = False
        self.id = carid
        self.name = carname
        self.positions = []
    Car.totalCars += 1

    def printCarInfo(self):
        print('Car ID: ', self.id)
```

```

        print('Car Name: ', self.name)
        print('Car Positions', self.positions)

    def getCarPosition(self):
        return self.positions[-1]

    def getCarId(self):
        return self.id

    def addCarPosition(self, position):
        self.positions.append(position)

    def getAllPositions(self):
        return self.positions

```

```

In [146]: # Read in a pickle file with bboxes saved
# Each item in the "all_bboxes" list will contain a
# list of boxes for one of the images shown above
#box_list = pickle.load( open( "/home/guenkat/Downloads/bbox_pickle.p", "rb" ) )

# Read in image similar to one shown above

Car.totalCars = 0
allCars = []
def add_heat(heatmap, bbox_list):
    # Iterate through list of bboxes
    for box in bbox_list:
        # Add += 1 for all pixels inside each bbox
        # Assuming each "box" takes the form ((x1, y1), (x2, y2))
        '''print('-----')
        print(box)
        print('-----')'''
        heatmap[box[0][1]:box[1][1], box[0][0]:box[1][0]] += 1

    # Return updated heatmap
    return heatmap# Iterate through list of bboxes

def apply_threshold(heatmap, threshold):
    # Zero out pixels below the threshold

    heatmap[heatmap <= threshold] = 0
    # Return thresholded map
    return heatmap

vicinity_check = 1
vicinity_print = 1

```

```

def draw_labeled_bboxes(img, labels, title = ''):
    global vicinity_check
    global vicinity_print
    # Iterate through all detected cars
    print('Number of cars', labels[1])
    activeCars = []
    for car_number in range(1, labels[1]+1):
        # Find pixels with each car_number label value
        nonzero = (labels[0] == car_number).nonzero()
        #print(car_number)
        #print(nonzero)
        #print()
        # Identify x and y values of those pixels
        nonzeroy = np.array(nonzero[0])
        nonzerox = np.array(nonzero[1])
        # Define a bounding box based on min/max x and y
        bbox = ((np.min(nonzerox), np.min(nonzeroy)), (np.max(nonzerox), np.max(nonzeroy)))
        print('bbox:car_number', bbox, ':', car_number)

        #checking if the box size is above the minimim size

        if (abs(bbox[0][0] - bbox[1][0]) > 48) and (abs(bbox[0][1] - bbox[1][1]) > 36):
            if not allCars:
                newCar = Car(car_number, 'car'+str(car_number))
                newCar.addCarPosition(bbox)
                allCars.append(newCar)
                activeCars.append(newCar)
            else:
                #check for vicinity and find the lease distance among already found cars
                minDistance = math.inf
                closestCarFound = False
                carToAddPosition = None

                for car in allCars:
                    #get min distance between current car centroid and the previous car
                    dist = distance(centroid(bbox[0], bbox[1]), centroid(car.getCarPosition()))
                    if dist < minDistance:
                        minDistance = dist
                        carToAddPosition = car

                #add car
                if minDistance < 48:
                    print('The minimum distance from car:', carToAddPosition.id, 'is', minDistance)
                    carToAddPosition.addCarPosition(bbox)
                    carToAddPosition.detected = True
                    activeCars.append(carToAddPosition)
                else:
                    newCar = Car(car_number, 'car'+str(car_number))

```

```

        newCar.addCarPosition(bbox)
        allCars.append(newCar)
        activeCars.append(newCar)

    else:
        pass

    # Draw the box on the image
    #for car in allCars[-labels[1]:]:
    for car in activeCars:
        allRects = car.getAllPositions()
        #print(allRects)

        if vicinity_check:
            if car.detected:
                if vicinity_print:
                    for rect in allRects[:-1]:
                        cv2.rectangle(img, rect[0], rect[1], (255,255,0), 4)
                    cv2.rectangle(img, allRects[-1][0], allRects[-1][1], (0,0,255), 4)
            else:
                pass
        else:
            cv2.rectangle(img, allRects[-1][0], allRects[-1][1], (0,0,255), 4)

        print('totalCars:', (len(allCars)))
        print('Car Number:', car.id, 'Car Positions:', allRects)
        #cv2.rectangle(img, (30,30), (150,150), (0,255,0), 3)

        #cv2.rectangle(img, bbox[0], bbox[1], (0,0,255), 4)
        #cv2.circle(img, centroid(allRects[-1][0], allRects[-1][1]), 30, (160,10,60),
        for car_number in range(1, labels[1]+1):

            nonzero = (labels[0] == car_number).nonzero()

            # Identify x and y values of those pixels
            nonzeroy = np.array(nonzero[0])
            nonzerox = np.array(nonzero[1])
            # Define a bounding box based on min/max x and y
            bbox = ((np.min(nonzerox), np.min(nonzeroy)), (np.max(nonzerox), np.max(nonzeroy)))
            #cv2.line(img, centroid(allRects[-1][0], allRects[-1][1]), centroid(bbox[0], bbox[1]), 2)
            #print('Line points are: ', centroid(allRects[-1][0], allRects[-1][1]), centroid(bbox[0], bbox[1]), 2)

            cv2.putText(img, title, (10,100), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255), 2, cv2.LINE_AA)
            # Return the image
            return img

```

In [147]: Car.totalCars
allCars

```
Out[147]: []
```

```
In [148]: # Adding heatmaps and boxes for all test images
use_default_number_of_processes = 1
use_sub_windows = 0
print_find_car_step_times = 0
find_car_hog_parallelism_enabled = 0

test_images_with_boxes = []
vicinity_print = 0
vicinity_check = 0
for image in test_images:
    allCars = []
    img = mpimg.imread(image)
    freeze_support()
    t0 = time.time()
    r = []
    r = parallelize_find_cars(img, 2)
    t1 = time.time()
    #print('Time taken for processing each image', round(t1-t0, 6))

    heat = np.zeros_like(img[:, :, 0]).astype(np.float)
    # Add heat to each box in box list
    heat = add_heat(heat,r)

    # Apply threshold to help remove false positives
    heat = apply_threshold(heat,3)
    #print(heat)

    # Visualize the heatmap when displaying
    heatmap = np.clip(heat, 0, 255)

    # Find final boxes from heatmap using label function
    labels = label(heatmap)

    draw_img = draw_labeled_bboxes(np.copy(img), labels)

    test_images_with_boxes.append(draw_img)

fig = plt.figure(figsize=(16,9))
plt.subplot(131)
plt.imshow(img)
plt.title('Car Positions')
plt.subplot(132)
plt.imshow(heatmap, cmap='hot')
plt.title('Heat Map')
plt.subplot(133)
plt.imshow(draw_img)
```

```
plt.title('final image')
fig.tight_layout()
```

Length of task list: 5

Number of processes used: 3

The times for each task are: [0.957202, 0.680369, 1.564253, 0.387931, 0.493829] with:

Minimum: 0.387931 Maximum: 1.564253 Average: 0.8167 seconds

Number of cars 2

bbox:car_number ((835, 400), (932, 497)) : 1

bbox:car_number ((1051, 400), (1273, 518)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((835, 400), (932, 497))]

totalCars: 2

Car Number: 2 Car Positions: [((1051, 400), (1273, 518))]

Length of task list: 5

Number of processes used: 3

The times for each task are: [0.905773, 1.416617, 0.631248, 0.435278, 0.387059] with:

Minimum: 0.387059 Maximum: 1.416617 Average: 0.7552 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

The times for each task are: [0.63203, 1.006826, 1.48108, 0.389066, 0.317116] with:

Minimum: 0.317116 Maximum: 1.48108 Average: 0.7652 seconds

Number of cars 1

bbox:car_number ((886, 400), (963, 478)) : 1

totalCars: 1

Car Number: 1 Car Positions: [((886, 400), (963, 478))]

Length of task list: 5

Number of processes used: 3

The times for each task are: [0.6387, 1.446704, 0.919077, 0.401181, 0.302374] with:

Minimum: 0.302374 Maximum: 1.446704 Average: 0.7416 seconds

Number of cars 2

bbox:car_number ((1048, 400), (1257, 498)) : 1

bbox:car_number ((837, 419), (932, 498)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1048, 400), (1257, 498))]

```
totalCars: 2
Car Number: 2 Car Positions: [((837, 419), (932, 498))]
Length of task list: 5
Number of processes used: 3
```

The times for each task are: [0.93964, 0.671368, 1.491534, 0.300531, 0.520643] with:

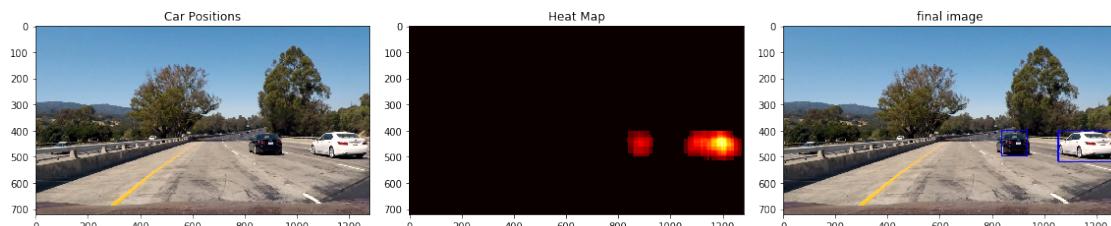
Minimum: 0.300531 Maximum: 1.491534 Average: 0.7847 seconds

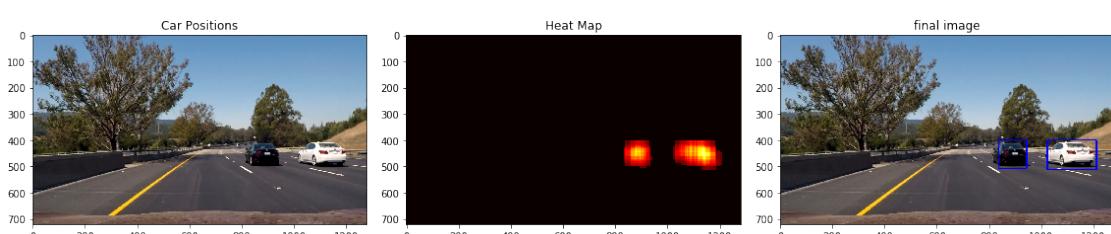
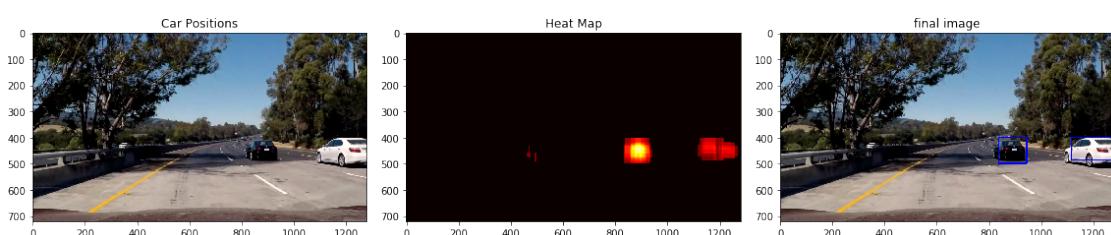
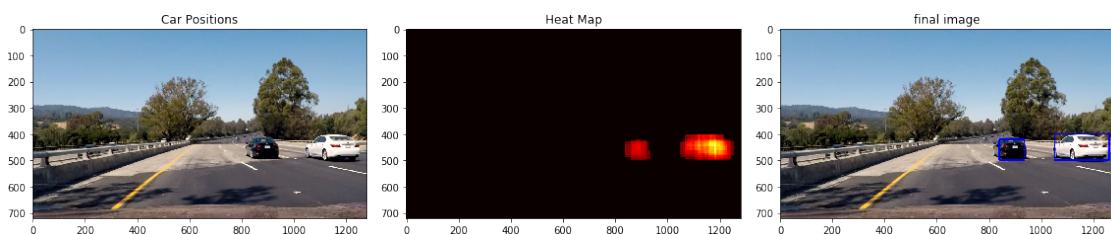
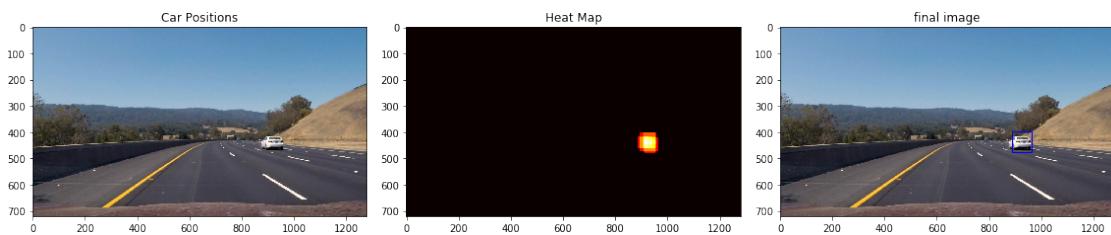
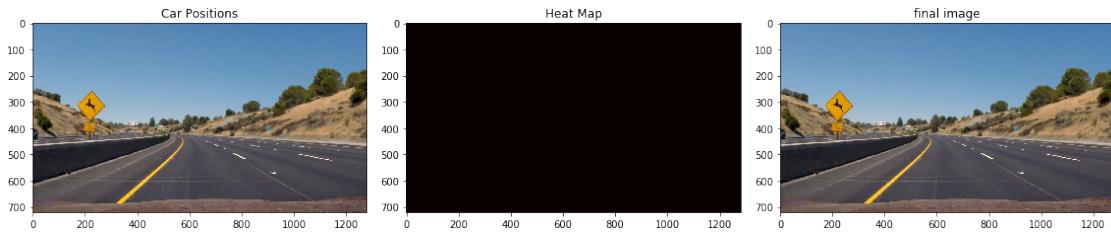
```
Number of cars 5
bbox:car_number ((834, 400), (942, 498)) : 1
bbox:car_number ((1113, 400), (1268, 485)) : 2
bbox:car_number ((462, 430), (477, 477)) : 3
bbox:car_number ((490, 457), (497, 492)) : 4
bbox:car_number ((535, 475), (537, 492)) : 5
totalCars: 2
Car Number: 1 Car Positions: [((834, 400), (942, 498))]
totalCars: 2
Car Number: 2 Car Positions: [((1113, 400), (1268, 485))]
Length of task list: 5
Number of processes used: 3
```

The times for each task are: [0.584478, 0.940101, 0.400793, 0.393006, 1.650379] with:

Minimum: 0.393006 Maximum: 1.650379 Average: 0.7938 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
bbox:car_number ((1021, 400), (1208, 512)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((834, 400), (942, 507))]
totalCars: 2
Car Number: 2 Car Positions: [((1021, 400), (1208, 512))]
```



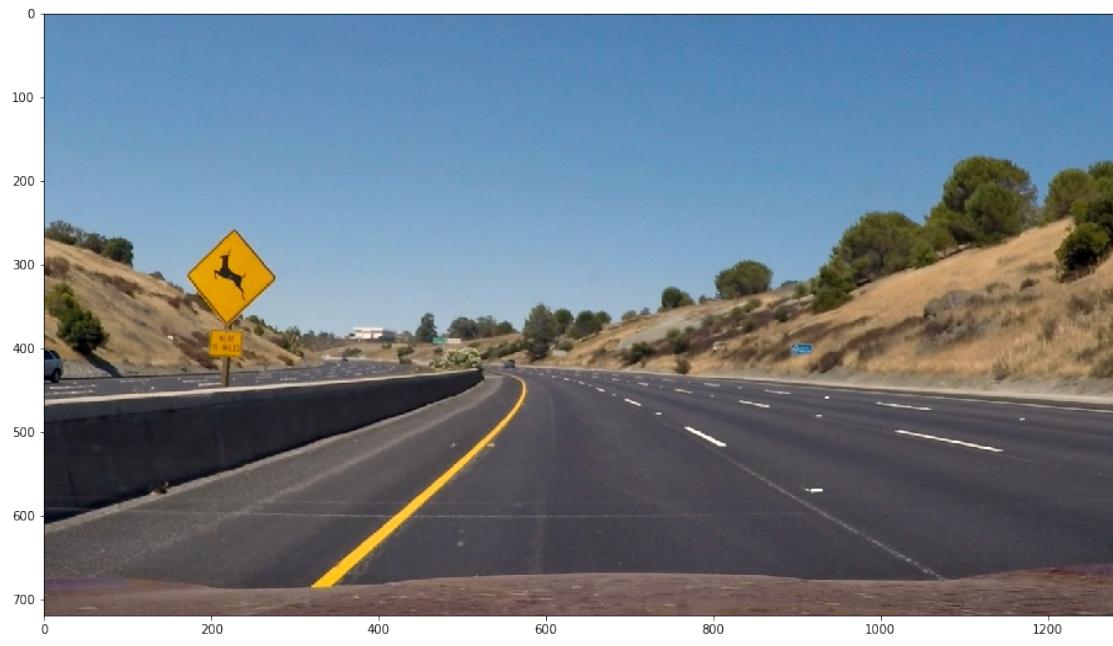


```
In [149]: Car.totalCars  
allCars
```

```
Out[149]: [<__main__.Car at 0x7f91490791d0>, <__main__.Car at 0x7f9126e73a58>]
```

```
In [150]: for image in test_images_with_boxes:  
    plt.figure(figsize=(16,9))  
    plt.imshow(image)
```









```
In [151]: # Adding heatmaps and boxes for all test images
allCars = []

test_images_with_boxes_and_vicinity = []

vicinity_check = 1
vicinity_print = 1
use_default_number_of_processes = 1

for image in test_images:
    img = mpimg.imread(image)
    freeze_support()
    t0 = time.time()
    r = []
    r = parallelize_find_cars(img, 2)
    t1 = time.time()
    #print('Time taken for processing each image', round(t1-t0, 6))

    heat = np.zeros_like(img[:, :, 0]).astype(np.float)
    # Add heat to each box in box list
    heat = add_heat(heat,r)

    # Apply threshold to help remove false positives
    heat = apply_threshold(heat,3)
    #print(heat)
```

```

# Visualize the heatmap when displaying
heatmap = np.clip(heat, 0, 255)

# Find final boxes from heatmap using label function
labels = label(heatmap)

draw_img = draw_labeled_bboxes(np.copy(img), labels)

test_images_with_boxes_and_vicinity.append(draw_img)

fig = plt.figure(figsize=(16,9))
plt.subplot(131)
plt.imshow(img)
plt.title('Car Positions')
plt.subplot(132)
plt.imshow(heatmap, cmap='hot')
plt.title('Heat Map')
plt.subplot(133)
plt.imshow(draw_img)
plt.title('final image')
fig.tight_layout()

```

Length of task list: 5

Number of processes used: 3

The times for each task are: [1.470642, 0.766231, 0.970883, 0.347614, 0.447416] with:

Minimum: 0.347614 Maximum: 1.470642 Average: 0.8006 seconds

Number of cars 2

bbox:car_number ((835, 400), (932, 497)) : 1

bbox:car_number ((1051, 400), (1273, 518)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((835, 400), (932, 497))]

totalCars: 2

Car Number: 2 Car Positions: [((1051, 400), (1273, 518))]

Length of task list: 5

Number of processes used: 3

The times for each task are: [0.990905, 0.803613, 0.527464, 1.642979, 0.343746] with:

Minimum: 0.343746 Maximum: 1.642979 Average: 0.8617 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

The times for each task are: [1.010341, 1.407531, 0.661536, 0.452674, 0.384485] with:

Minimum: 0.384485 Maximum: 1.407531 Average: 0.7833 seconds

Number of cars 1
bbox:car_number ((886, 400), (963, 478)) : 1
The minimum distance from car: 1 is 41.97618372363071
totalCars: 2
Car Number: 1 Car Positions: [((835, 400), (932, 497)), ((886, 400), (963, 478))]
Length of task list: 5
Number of processes used: 3

The times for each task are: [0.964408, 1.521052, 0.69557, 0.4762, 0.287346] with:

Minimum: 0.287346 Maximum: 1.521052 Average: 0.7889 seconds

Number of cars 2
bbox:car_number ((1048, 400), (1257, 498)) : 1
The minimum distance from car: 2 is 14.142135623730951
bbox:car_number ((837, 419), (932, 498)) : 2
The minimum distance from car: 1 is 44.28317965096906
totalCars: 2
Car Number: 2 Car Positions: [((1051, 400), (1273, 518)), ((1048, 400), (1257, 498))]
totalCars: 2
Car Number: 1 Car Positions: [((835, 400), (932, 497)), ((886, 400), (963, 478)), ((837, 419), (932, 498))]
Length of task list: 5
Number of processes used: 3

The times for each task are: [0.96782, 1.383271, 0.79454, 0.38984, 0.403341] with:

Minimum: 0.38984 Maximum: 1.383271 Average: 0.7878 seconds

Number of cars 5
bbox:car_number ((834, 400), (942, 498)) : 1
The minimum distance from car: 1 is 9.848857801796104
bbox:car_number ((1113, 400), (1268, 485)) : 2
The minimum distance from car: 2 is 38.63935817272331
bbox:car_number ((462, 430), (477, 477)) : 3
bbox:car_number ((490, 457), (497, 492)) : 4
bbox:car_number ((535, 475), (537, 492)) : 5
totalCars: 2
Car Number: 1 Car Positions: [((835, 400), (932, 497)), ((886, 400), (963, 478)), ((837, 419), (932, 498))]
totalCars: 2
Car Number: 2 Car Positions: [((1051, 400), (1273, 518)), ((1048, 400), (1257, 498)), ((1113, 400), (1268, 485))]
Length of task list: 5
Number of processes used: 3

The times for each task are: [0.625919, 1.005268, 1.411205, 0.455469, 0.340848] with:

Minimum: 0.340848 Maximum: 1.411205 Average: 0.76777 seconds

Number of cars 2

bbox:car_number ((834, 400), (942, 507)) : 1

The minimum distance from car: 1 is 4.0

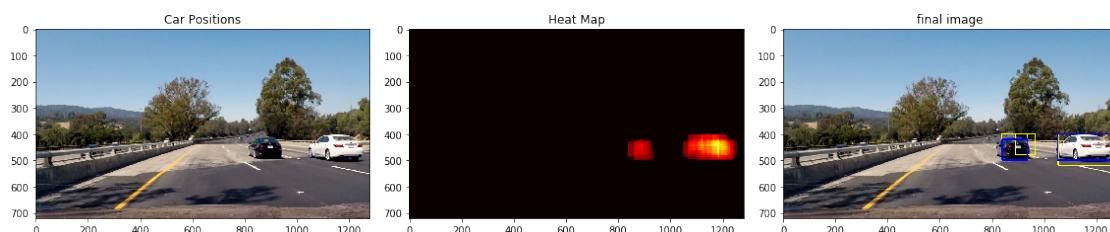
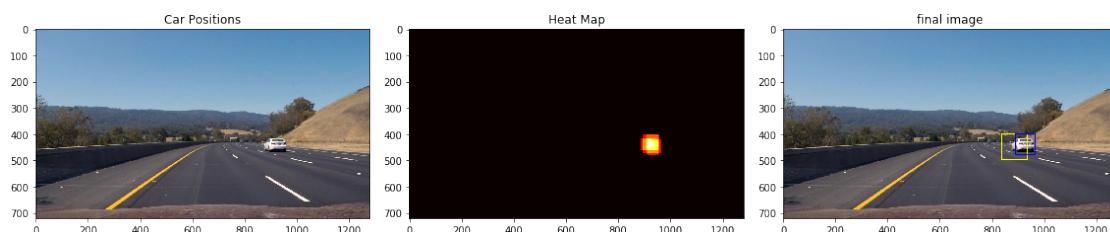
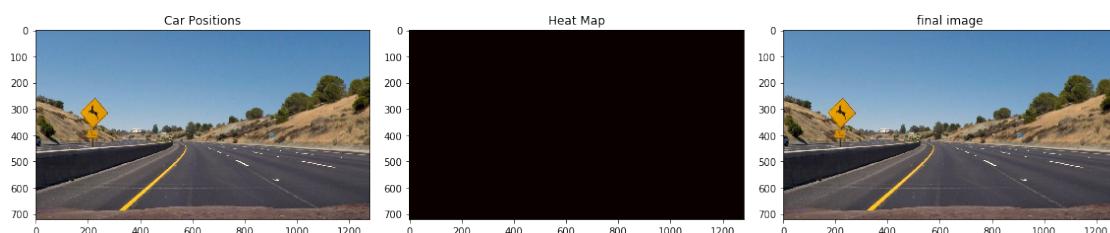
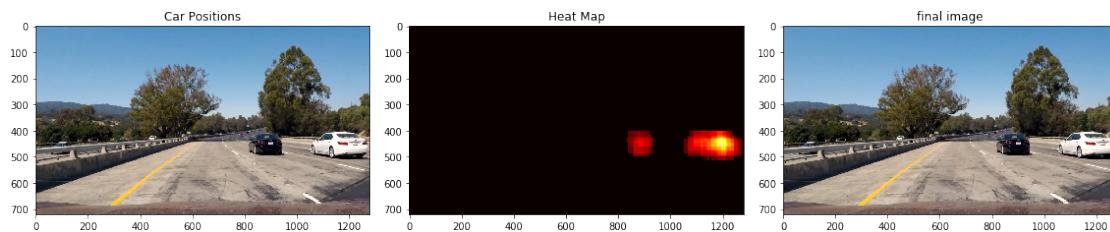
bbox:car_number ((1021, 400), (1208, 512)) : 2

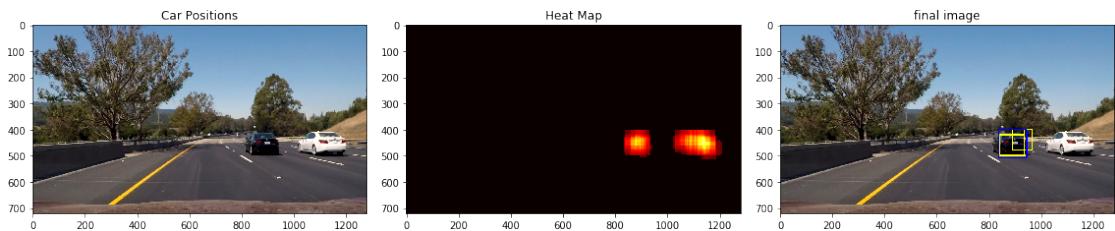
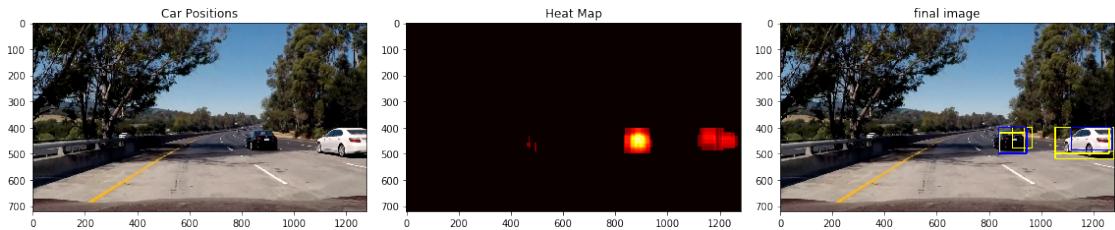
totalCars: 3

Car Number: 1 Car Positions: [((835, 400), (932, 497)), ((886, 400), (963, 478)), ((837, 419), (934, 497))]

totalCars: 3

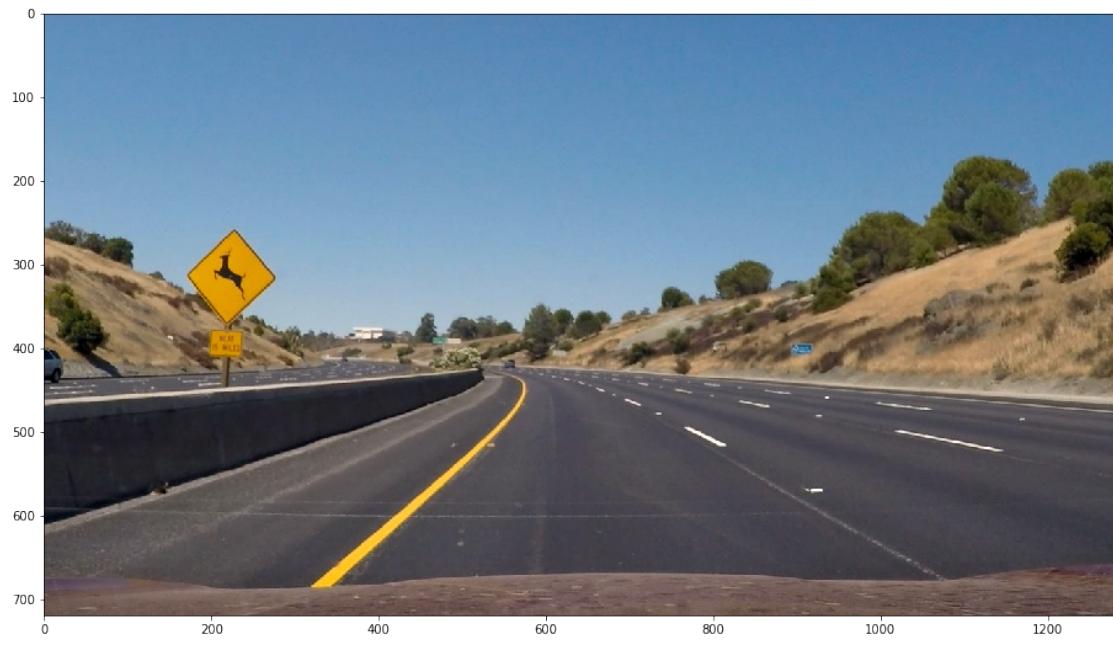
Car Number: 2 Car Positions: [((1021, 400), (1208, 512))]



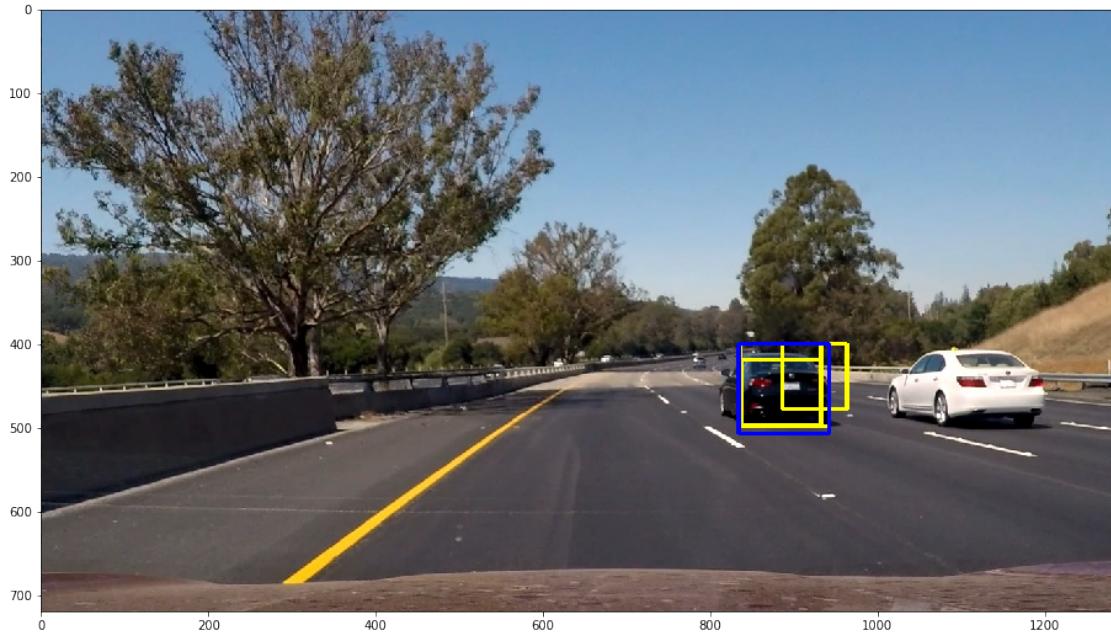


```
In [152]: for image in test_images_with_boxes_and_vicinity:
    plt.figure(figsize=(16,9))
    plt.imshow(image)
```









```
In [153]: heatmap_template = np.copy(img)
threshold = 3
def process_image(img, title=''):
    use_default_number_of_processes = 1

    boxes_list = []
    #image_with_all_boxes, boxes_list = find_cars(img, ystart, ystop, scale, svc, X_s
    #
    #freeze_support()
    parallelize_start_time = time.time()
    boxes_list = parallelize_find_cars(img, 3)
    parallelize_end_time = time.time()
    #print('Length of rectangles: ', len(boxes_list))
    #print('after parallelizing, and time taken is, ', round(parallelize_end_time - pa
    heat_template = np.zeros_like(img[:, :, 0]).astype(np.float)
    heat_template = add_heat(heat_template, boxes_list)
    heat_template = apply_threshold(heat_template, threshold)
    heatmap_template = np.clip(heat_template, 0, 255)
    labels = label(heatmap_template)
    #print(labels)
    draw_img = draw_labeled_bboxes(np.copy(img), labels, title)

    return draw_img
```

```
In [154]: vicinity_check = 1
vicinity_print = 1
```

```

images = []
heatmaps = []
processed_images = []

Car.totalCars = 0
allCars = []
for image in test_images:
    #print(image)
    img = mpimg.imread(image)
    images.append(img)
    t1 = time.time()
    freeze_support()
    pi = process_image(img, title=image)
    t2 = time.time()
    print("Time for processing an image", round(t2-t1, 4), "seconds")
    heatmaps.append(heatmap_template)
    processed_images.append(pi)

```

Length of task list: 5

Number of processes used: 3

The times for each task are: [1.371695, 1.112267, 0.719313, 0.39644, 0.362925] with:

Minimum: 0.362925 Maximum: 1.371695 Average: 0.7925 seconds

Number of cars 2

bbox:car_number ((835, 400), (932, 497)) : 1

bbox:car_number ((1051, 400), (1273, 518)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((835, 400), (932, 497))]

totalCars: 2

Car Number: 2 Car Positions: [((1051, 400), (1273, 518))]

Time for processing an image 3.1089 seconds

Length of task list: 5

Number of processes used: 3

The times for each task are: [0.650437, 0.903778, 1.615771, 0.298262, 0.400819] with:

Minimum: 0.298262 Maximum: 1.615771 Average: 0.7738 seconds

Number of cars 0

Time for processing an image 2.7163 seconds

Length of task list: 5

Number of processes used: 3

The times for each task are: [0.636857, 0.898795, 1.400289, 0.313535, 0.373459] with:

Minimum: 0.313535 Maximum: 1.400289 Average: 0.7246 seconds

Number of cars 1
bbox:car_number ((886, 400), (963, 478)) : 1
The minimum distance from car: 1 is 41.97618372363071
totalCars: 2
Car Number: 1 Car Positions: [((835, 400), (932, 497)), ((886, 400), (963, 478))]
Time for processing an image 2.533 seconds
Length of task list: 5
Number of processes used: 3

The times for each task are: [0.657221, 0.933552, 1.48646, 0.436013, 0.401572] with:

Minimum: 0.401572 Maximum: 1.48646 Average: 0.783 seconds

Number of cars 2
bbox:car_number ((1048, 400), (1257, 498)) : 1
The minimum distance from car: 2 is 14.142135623730951
bbox:car_number ((837, 419), (932, 498)) : 2
The minimum distance from car: 1 is 44.28317965096906
totalCars: 2
Car Number: 2 Car Positions: [((1051, 400), (1273, 518)), ((1048, 400), (1257, 498))]
totalCars: 2
Car Number: 1 Car Positions: [((835, 400), (932, 497)), ((886, 400), (963, 478)), ((837, 419), (1113, 400))]
Time for processing an image 2.7327 seconds
Length of task list: 5
Number of processes used: 3

The times for each task are: [0.586171, 0.906098, 1.460266, 0.399216, 0.361756] with:

Minimum: 0.361756 Maximum: 1.460266 Average: 0.7427 seconds

Number of cars 5
bbox:car_number ((834, 400), (942, 498)) : 1
The minimum distance from car: 1 is 9.848857801796104
bbox:car_number ((1113, 400), (1268, 485)) : 2
The minimum distance from car: 2 is 38.63935817272331
bbox:car_number ((462, 430), (477, 477)) : 3
bbox:car_number ((490, 457), (497, 492)) : 4
bbox:car_number ((535, 475), (537, 492)) : 5
totalCars: 2
Car Number: 1 Car Positions: [((835, 400), (932, 497)), ((886, 400), (963, 478)), ((837, 419), (1113, 400))]
totalCars: 2
Car Number: 2 Car Positions: [((1051, 400), (1273, 518)), ((1048, 400), (1257, 498)), ((1113, 400), (1268, 485))]
Time for processing an image 2.9357 seconds
Length of task list: 5
Number of processes used: 3

```
The times for each task are: [0.872787, 1.511294, 0.724144, 0.408122, 0.351061] with:
```

```
Minimum: 0.351061 Maximum: 1.511294 Average: 0.7735 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1021, 400), (1208, 512)) : 2
totalCars: 3
Car Number: 1 Car Positions: [((835, 400), (932, 497)), ((886, 400), (963, 478)), ((837, 419),
totalCars: 3
Car Number: 2 Car Positions: [((1021, 400), (1208, 512))]
Time for processing an image 2.5512 seconds
```

```
In [155]: allCars
```

```
Out[155]: [<__main__.Car at 0x7f9142b53b00>,
<__main__.Car at 0x7f914289a2b0>,
<__main__.Car at 0x7f9126e73518>]
```

```
In [156]: for car in allCars:
    p = car.getCarPosition()
    print('Car Position:', p[0], p[1])
    print('Centroid: ', centroid(p[0], p[1]))
```

```
Car Position: (834, 400) (942, 507)
Centroid: (888, 453)
Car Position: (1113, 400) (1268, 485)
Centroid: (1190, 442)
Car Position: (1021, 400) (1208, 512)
Centroid: (1114, 456)
```

```
In [157]: for car in allCars:
    print(car.getAllPositions())
    print(car.detected)
```

```
[((835, 400), (932, 497)), ((886, 400), (963, 478)), ((837, 419), (932, 498)), ((834, 400), (942,
True
[((1051, 400), (1273, 518)), ((1048, 400), (1257, 498)), ((1113, 400), (1268, 485))]
True
[((1021, 400), (1208, 512))]
False
```

```
In [158]: vicinity_check = 1
vicinity_print = 1
```

```

Car.totalCars = 0
allCars= []

project_video_output = 'project_videos_output/test_video_after_process_with_vicinity9.mp4'
## To speed up the testing process you may want to try your pipeline on a shorter subclip
## To do so add .subclip(start_second,end_second) to the end of the line below
## Where start_second and end_second are integer values representing the start and end
## You may also uncomment the following line for a subclip of the first 5 seconds
##clip1 = VideoFileClip("test_videos/solidWhiteRight.mp4").subclip(0,5)
clip1 = VideoFileClip("test_video.mp4")
white_clip = clip1.fl_image(process_image) #NOTE: this function expects color images!
%time white_clip.write_videofile(project_video_output, audio=False)

```

Length of task list: 5

Number of processes used: 3

The times for each task are: [0.937701, 1.467114, 0.743602, 0.41732, 0.325925] with:

Minimum: 0.325925 Maximum: 1.467114 Average: 0.7783 seconds

Number of cars 2

bbox:car_number ((1030, 415), (1162, 495)) : 1

bbox:car_number ((834, 419), (942, 507)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495))]

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507))]

[MoviePy] >>> Building video project_videos_output/test_video_after_process_with_vicinity9.mp4

[MoviePy] Writing video project_videos_output/test_video_after_process_with_vicinity9.mp4

0% | 0/39 [00:00<?, ?it/s]

Length of task list: 5

Number of processes used: 3

3% | 1/39 [00:02<01:38, 2.60s/it]

The times for each task are: [0.901135, 1.392057, 0.693628, 0.29529, 0.406323] with:

Minimum: 0.29529 Maximum: 1.392057 Average: 0.7377 seconds

Number of cars 2

bbox:car_number ((1030, 415), (1162, 495)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((834, 419), (942, 507)) : 2

The minimum distance from car: 2 is 0.0

```
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

5% | 2/39 [00:05<01:34, 2.56s/it]

The times for each task are: [0.983949, 0.610144, 1.749087, 0.382272, 0.387763] with:
Minimum: 0.382272 Maximum: 1.749087 Average: 0.8226 seconds

```
Number of cars 2
bbox:car_number ((1030, 400), (1181, 495)) : 1
The minimum distance from car: 1 is 12.041594578792296
bbox:car_number ((835, 419), (942, 498)) : 2
The minimum distance from car: 2 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (
Length of task list: 5
Number of processes used: 3
```

8% | 3/39 [00:07<01:31, 2.55s/it]

The times for each task are: [0.950008, 0.634422, 1.49882, 0.405188, 0.386687] with:
Minimum: 0.386687 Maximum: 1.49882 Average: 0.775 seconds

```
Number of cars 2
bbox:car_number ((1045, 400), (1180, 485)) : 1
The minimum distance from car: 1 is 8.602325267042627
bbox:car_number ((834, 423), (942, 507)) : 2
The minimum distance from car: 2 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (
Length of task list: 5
Number of processes used: 3
```

10% | 4/39 [00:10<01:30, 2.60s/it]

The times for each task are: [0.918309, 0.602956, 1.441177, 0.300542, 0.508627] with:

Minimum: 0.300542 Maximum: 1.441177 Average: 0.7543 seconds

Number of cars 2

bbox:car_number ((1030, 400), (1181, 495)) : 1

The minimum distance from car: 1 is 8.602325267042627

bbox:car_number ((835, 419), (942, 507)) : 2

The minimum distance from car: 2 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 400), (1181, 495))]

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

Length of task list: 5

Number of processes used: 3

13% | 5/39 [00:13<01:29, 2.64s/it]

The times for each task are: [0.638113, 0.876783, 1.366435, 0.398081, 0.305453] with:

Minimum: 0.305453 Maximum: 1.366435 Average: 0.717 seconds

Number of cars 2

bbox:car_number ((1030, 400), (1182, 497)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((835, 415), (942, 507)) : 2

The minimum distance from car: 2 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 400), (1182, 495))]

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

Length of task list: 5

Number of processes used: 3

15% | 6/39 [00:15<01:27, 2.66s/it]

The times for each task are: [0.893013, 0.706265, 1.525954, 0.42592, 0.325345] with:

Minimum: 0.325345 Maximum: 1.525954 Average: 0.7753 seconds

Number of cars 2

bbox:car_number ((1027, 400), (1185, 497)) : 1

The minimum distance from car: 1 is 0.0

```
bbox:car_number ((835, 415), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

18% | 7/39 [00:18<01:25, 2.68s/it]

The times for each task are: [0.861189, 1.358724, 0.709578, 0.407013, 0.498375] with:

Minimum: 0.407013 Maximum: 1.358724 Average: 0.767 seconds

```
Number of cars 2
bbox:car_number ((1027, 400), (1185, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((834, 415), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

21% | 8/39 [00:21<01:22, 2.67s/it]

The times for each task are: [0.62858, 1.605622, 0.92216, 0.393852, 0.29396] with:

Minimum: 0.29396 Maximum: 1.605622 Average: 0.7688 seconds

```
Number of cars 2
bbox:car_number ((1021, 400), (1185, 497)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((834, 415), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

23% | 9/39 [00:24<01:20, 2.68s/it]

The times for each task are: [1.490749, 0.648965, 0.986229, 0.413019, 0.29217] with:

Minimum: 0.29217 Maximum: 1.490749 Average: 0.7662 seconds

Number of cars 2

bbox:car_number ((1027, 400), (1185, 498)) : 1

The minimum distance from car: 1 is 3.1622776601683795

bbox:car_number ((835, 415), (942, 507)) : 2

The minimum distance from car: 2 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (

Length of task list: 5

Number of processes used: 3

26% | 10/39 [00:26<01:17, 2.66s/it]

The times for each task are: [1.021623, 1.395977, 0.638543, 0.402343, 0.348846] with:

Minimum: 0.348846 Maximum: 1.395977 Average: 0.7615 seconds

Number of cars 2

bbox:car_number ((1027, 400), (1185, 498)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((834, 415), (942, 507)) : 2

The minimum distance from car: 2 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (

Length of task list: 5

Number of processes used: 3

28% | 11/39 [00:29<01:14, 2.65s/it]

The times for each task are: [0.980627, 0.606827, 1.367542, 0.287558, 0.422659] with:

Minimum: 0.287558 Maximum: 1.367542 Average: 0.733 seconds

```
Number of cars 2
bbox:car_number ((1021, 400), (1185, 498)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((834, 415), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (
Length of task list: 5
Number of processes used: 3
```

31% | 12/39 [00:31<01:11, 2.64s/it]

The times for each task are: [0.837865, 0.717055, 1.639264, 0.450155, 0.349452] with:

Minimum: 0.349452 Maximum: 1.639264 Average: 0.7988 seconds

```
Number of cars 2
bbox:car_number ((1027, 400), (1197, 498)) : 1
The minimum distance from car: 1 is 9.0
bbox:car_number ((834, 415), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (
Length of task list: 5
Number of processes used: 3
```

33% | 13/39 [00:34<01:09, 2.65s/it]

The times for each task are: [1.330133, 1.002794, 0.713219, 0.325143, 0.442342] with:

Minimum: 0.325143 Maximum: 1.330133 Average: 0.7627 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 2 is 8.0
bbox:car_number ((1027, 400), (1208, 498)) : 2
The minimum distance from car: 1 is 5.0
bbox:car_number ((1210, 430), (1211, 477)) : 3
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]  
Length of task list: 5  
Number of processes used: 3
```

36% | 14/39 [00:37<01:06, 2.67s/it]

The times for each task are: [0.98086, 0.697109, 1.415092, 0.382551, 0.320913] with:

Minimum: 0.320913 Maximum: 1.415092 Average: 0.7593 seconds

```
Number of cars 3  
bbox:car_number ((834, 400), (934, 498)) : 1  
The minimum distance from car: 2 is 5.656854249492381  
bbox:car_number ((1021, 400), (1208, 498)) : 2  
The minimum distance from car: 1 is 3.0  
bbox:car_number ((1210, 430), (1211, 477)) : 3  
totalCars: 2  
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]  
totalCars: 2  
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]  
Length of task list: 5  
Number of processes used: 3
```

38% | 15/39 [00:40<01:04, 2.68s/it]

The times for each task are: [0.827016, 1.39836, 0.66891, 0.364791, 0.375047] with:

Minimum: 0.364791 Maximum: 1.39836 Average: 0.7268 seconds

```
Number of cars 2  
bbox:car_number ((834, 400), (942, 507)) : 1  
The minimum distance from car: 2 is 5.656854249492381  
bbox:car_number ((1030, 400), (1211, 512)) : 2  
The minimum distance from car: 1 is 9.219544457292887  
totalCars: 2  
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]  
totalCars: 2  
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]  
Length of task list: 5  
Number of processes used: 3
```

41% | 16/39 [00:42<01:01, 2.67s/it]

The times for each task are: [0.865011, 0.658433, 1.611353, 0.367154, 0.482796] with:

Minimum: 0.367154 Maximum: 1.611353 Average: 0.7969 seconds

Number of cars 3

bbox:car_number ((835, 400), (942, 507)) : 1

The minimum distance from car: 2 is 0.0

bbox:car_number ((1021, 400), (1208, 498)) : 2

The minimum distance from car: 1 is 9.219544457292887

bbox:car_number ((1210, 427), (1211, 462)) : 3

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

Length of task list: 5

Number of processes used: 3

44% | 17/39 [00:45<00:58, 2.66s/it]

The times for each task are: [0.693687, 0.879708, 1.549325, 0.439976, 0.289778] with:

Minimum: 0.289778 Maximum: 1.549325 Average: 0.7705 seconds

Number of cars 2

bbox:car_number ((834, 400), (942, 507)) : 1

The minimum distance from car: 2 is 0.0

bbox:car_number ((1044, 400), (1208, 498)) : 2

The minimum distance from car: 1 is 12.0

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

Length of task list: 5

Number of processes used: 3

46% | 18/39 [00:47<00:55, 2.67s/it]

The times for each task are: [0.867817, 0.68611, 1.341555, 0.413603, 0.367185] with:

Minimum: 0.367185 Maximum: 1.341555 Average: 0.7353 seconds

Number of cars 2

bbox:car_number ((834, 400), (942, 512)) : 1

The minimum distance from car: 2 is 3.0

bbox:car_number ((1060, 400), (1212, 498)) : 2

```
The minimum distance from car: 1 is 10.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

49% | 19/39 [00:50<00:53, 2.67s/it]

The times for each task are: [0.86365, 0.625943, 1.405761, 0.397559, 0.372901] with:

Minimum: 0.372901 Maximum: 1.405761 Average: 0.7332 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (934, 512)) : 1
The minimum distance from car: 2 is 3.0
bbox:car_number ((1046, 400), (1214, 498)) : 2
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

51% | 20/39 [00:53<00:50, 2.67s/it]

The times for each task are: [0.867489, 0.684841, 1.414738, 0.397411, 0.388897] with:

Minimum: 0.388897 Maximum: 1.414738 Average: 0.7507 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (942, 512)) : 1
The minimum distance from car: 2 is 4.0
bbox:car_number ((1046, 400), (1214, 498)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

54%| 21/39 [00:56<00:48, 2.69s/it]

The times for each task are: [1.035725, 0.673961, 1.522444, 0.430243, 0.295208] with:

Minimum: 0.295208 Maximum: 1.522444 Average: 0.7915 seconds

Number of cars 2

bbox:car_number ((1060, 400), (1214, 498)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((837, 415), (932, 507)) : 2

The minimum distance from car: 2 is 7.0710678118654755

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

Length of task list: 5

Number of processes used: 3

56%| 22/39 [00:59<00:45, 2.69s/it]

The times for each task are: [0.615135, 0.910694, 1.398014, 0.281487, 0.36981] with:

Minimum: 0.281487 Maximum: 1.398014 Average: 0.715 seconds

Number of cars 3

bbox:car_number ((1048, 400), (1219, 498)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((835, 419), (934, 507)) : 2

The minimum distance from car: 2 is 2.0

bbox:car_number ((552, 476), (586, 522)) : 3

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

Length of task list: 5

Number of processes used: 3

59%| 23/39 [01:02<00:43, 2.70s/it]

The times for each task are: [0.850592, 1.41095, 0.762677, 0.398161, 0.285522] with:

Minimum: 0.285522 Maximum: 1.41095 Average: 0.7416 seconds

Number of cars 3

bbox:car_number ((1060, 400), (1219, 498)) : 1

```
The minimum distance from car: 1 is 6.0
bbox:car_number ((835, 419), (934, 507)) : 2
The minimum distance from car: 2 is 0.0
bbox:car_number ((552, 476), (597, 522)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

62% | 24/39 [01:04<00:40, 2.69s/it]

```
The times for each task are: [0.605662, 1.056707, 1.402424, 0.362595, 0.394141] with:
Minimum: 0.362595 Maximum: 1.402424 Average: 0.7643 seconds
```

```
Number of cars 2
bbox:car_number ((836, 400), (934, 498)) : 1
The minimum distance from car: 2 is 14.035668847618199
bbox:car_number ((1046, 400), (1230, 498)) : 2
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

64% | 25/39 [01:07<00:37, 2.70s/it]

```
The times for each task are: [0.996791, 1.421027, 0.637841, 0.477514, 0.374735] with:
Minimum: 0.374735 Maximum: 1.421027 Average: 0.7816 seconds
```

```
Number of cars 2
bbox:car_number ((836, 400), (932, 497)) : 1
The minimum distance from car: 2 is 1.4142135623730951
bbox:car_number ((1051, 400), (1230, 498)) : 2
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
```

Number of processes used: 3

67%| 26/39 [01:10<00:35, 2.70s/it]

The times for each task are: [0.886839, 0.668372, 1.421979, 0.39007, 0.343273] with:

Minimum: 0.343273 Maximum: 1.421979 Average: 0.7421 seconds

Number of cars 2

bbox:car_number ((836, 400), (934, 497)) : 1

The minimum distance from car: 2 is 1.0

bbox:car_number ((1051, 400), (1219, 498)) : 2

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

Length of task list: 5

Number of processes used: 3

69%| 27/39 [01:12<00:32, 2.70s/it]

The times for each task are: [0.893027, 0.701565, 1.380678, 0.364415, 0.381369] with:

Minimum: 0.364415 Maximum: 1.380678 Average: 0.7442 seconds

Number of cars 2

bbox:car_number ((836, 400), (934, 497)) : 1

The minimum distance from car: 2 is 0.0

bbox:car_number ((1048, 400), (1219, 498)) : 2

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

Length of task list: 5

Number of processes used: 3

72%| 28/39 [01:15<00:29, 2.71s/it]

The times for each task are: [0.585969, 0.923946, 0.411761, 0.305943, 1.613784] with:

Minimum: 0.305943 Maximum: 1.613784 Average: 0.7683 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 2 is 5.830951894845301
bbox:car_number ((1048, 400), (1230, 498)) : 2
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

74%| | 29/39 [01:18<00:27, 2.70s/it]

The times for each task are: [0.92557, 0.61783, 1.426764, 0.383614, 0.294276] with:

Minimum: 0.294276 Maximum: 1.426764 Average: 0.7296 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (934, 498)) : 1
The minimum distance from car: 2 is 5.656854249492381
bbox:car_number ((1051, 400), (1242, 498)) : 2
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

77%| | 30/39 [01:20<00:24, 2.69s/it]

The times for each task are: [1.355113, 0.869788, 0.414028, 0.709016, 0.297485] with:

Minimum: 0.297485 Maximum: 1.355113 Average: 0.7291 seconds

```
Number of cars 2
bbox:car_number ((856, 400), (934, 498)) : 1
The minimum distance from car: 2 is 11.0
bbox:car_number ((1051, 400), (1241, 498)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
```

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3

79% | 31/39 [01:23<00:21, 2.69s/it]

The times for each task are: [1.100012, 1.516692, 0.763974, 0.381355, 0.494044] with:

Minimum: 0.381355 Maximum: 1.516692 Average: 0.8512 seconds

Number of cars 2
bbox:car_number ((1065, 400), (1182, 495)) : 1
The minimum distance from car: 1 is 23.08679276123039
bbox:car_number ((859, 427), (917, 485)) : 2
The minimum distance from car: 2 is 9.899494936611665
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (943, 507))]
Length of task list: 5
Number of processes used: 3

82% | 32/39 [01:25<00:18, 2.69s/it]

The times for each task are: [0.888215, 1.561025, 0.626446, 0.325041, 0.377802] with:

Minimum: 0.325041 Maximum: 1.561025 Average: 0.7557 seconds

Number of cars 1
bbox:car_number ((1075, 400), (1230, 497)) : 1
The minimum distance from car: 1 is 29.017236257093817
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3

85% | 33/39 [01:28<00:16, 2.68s/it]

The times for each task are: [1.365438, 0.631601, 0.919913, 0.442244, 0.286235] with:

Minimum: 0.286235 Maximum: 1.365438 Average: 0.7291 seconds

Number of cars 2

```
bbox:car_number ((1102, 400), (1211, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((859, 427), (931, 498)) : 2
The minimum distance from car: 2 is 9.219544457292887
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (943, 507))]
Length of task list: 5
Number of processes used: 3
```

87%| | 34/39 [01:30<00:13, 2.67s/it]

The times for each task are: [0.884907, 0.63189, 1.496588, 0.425736, 0.338088] with:
Minimum: 0.338088 Maximum: 1.496588 Average: 0.7554 seconds

```
Number of cars 1
bbox:car_number ((1113, 400), (1214, 498)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

90%| | 35/39 [01:33<00:10, 2.67s/it]

The times for each task are: [1.443932, 0.577981, 1.033907, 0.393313, 0.442276] with:
Minimum: 0.393313 Maximum: 1.443932 Average: 0.7783 seconds

```
Number of cars 1
bbox:car_number ((1067, 400), (1227, 498)) : 1
The minimum distance from car: 1 is 16.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

92%|| 36/39 [01:36<00:08, 2.67s/it]

The times for each task are: [0.906781, 1.402326, 0.712218, 0.412001, 0.380705] with:

```
Minimum: 0.380705 Maximum: 1.402326 Average: 0.7628 seconds
```

```
Number of cars 1
bbox:car_number ((1067, 400), (1230, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
```

```
Length of task list: 5
Number of processes used: 3
```

```
95%|| 37/39 [01:38<00:05, 2.66s/it]
```

```
The times for each task are: [0.620217, 0.915671, 0.401611, 0.293286, 1.621856] with:
```

```
Minimum: 0.293286 Maximum: 1.621856 Average: 0.7705 seconds
```

```
Number of cars 1
bbox:car_number ((1067, 400), (1238, 498)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
```

```
Length of task list: 5
Number of processes used: 3
```

```
97%|| 38/39 [01:41<00:02, 2.66s/it]
```

```
The times for each task are: [0.982317, 1.383523, 0.627324, 0.465644, 0.331979] with:
```

```
Minimum: 0.331979 Maximum: 1.383523 Average: 0.7582 seconds
```

```
Number of cars 1
bbox:car_number ((1067, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
```

```
[MoviePy] Done.
```

```
[MoviePy] >>> Video ready: project_videos_output/test_video_after_process_with_vicinity9.mp4
```

```
CPU times: user 3.44 s, sys: 39.1 s, total: 42.6 s
```

```
Wall time: 1min 42s
```

```
In [159]: Car.totalCars
```

```
Out[159]: 2
```

```
In [160]: for car in allCars:
```

```
    print(car.getAllPositions())
    print(car.detected)
```

```
[((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 400), (1181, 495)), ((1045, 400), (1181, 495))]
```

```
True
```

```
[((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 498)), ((834, 423), (942, 498))]
```

```
True
```

```
In [161]: vicinity_check = 1
```

```
vicinity_print = 0
```

```
Car.totalCars = 0
```

```
allCars= []
```

```
project_video_output = 'project_videos_output/test_video_after_process_without_vicinity9.mp4'
## To speed up the testing process you may want to try your pipeline on a shorter subclip
## To do so add .subclip(start_second,end_second) to the end of the line below
## Where start_second and end_second are integer values representing the start and end
## You may also uncomment the following line for a subclip of the first 5 seconds
##clip1 = VideoFileClip( "test_videos/solidWhiteRight.mp4").subclip(0,5)
clip1 = VideoFileClip("test_video.mp4")
white_clip = clip1.fl_image(process_image) #NOTE: this function expects color images!
%time white_clip.write_videofile(project_video_output, audio=False)
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
The times for each task are: [0.889306, 1.472563, 0.726748, 0.344131, 0.435059] with:
```

```
Minimum: 0.344131 Maximum: 1.472563 Average: 0.7736 seconds
```

```
Number of cars 2
```

```
bbox:car_number ((1030, 415), (1162, 495)) : 1
```

```
bbox:car_number ((834, 419), (942, 507)) : 2
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1030, 415), (1162, 495))]
```

```
totalCars: 2
```

```
Car Number: 2 Car Positions: [((834, 419), (942, 507))]
```

```
[MoviePy] >>> Building video project_videos_output/test_video_after_process_without_vicinity9.mp4
```

```
[MoviePy] Writing video project_videos_output/test_video_after_process_without_vicinity9.mp4
```

```
0% | 0/39 [00:00<?, ?it/s]
```

```
Length of task list: 5
Number of processes used: 3
```

```
3%|           | 1/39 [00:02<01:42,  2.69s/it]
```

```
The times for each task are: [0.878447, 0.670677, 1.569061, 0.349916, 0.395695] with:
```

```
Minimum: 0.349916 Maximum: 1.569061 Average: 0.7728 seconds
```

```
Number of cars 2
bbox:car_number ((1030, 415), (1162, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((834, 419), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

```
5%|           | 2/39 [00:05<01:35,  2.58s/it]
```

```
The times for each task are: [0.644531, 0.864161, 1.451973, 0.48983, 0.290949] with:
```

```
Minimum: 0.290949 Maximum: 1.451973 Average: 0.7483 seconds
```

```
Number of cars 2
bbox:car_number ((1030, 400), (1181, 495)) : 1
The minimum distance from car: 1 is 12.041594578792296
bbox:car_number ((835, 419), (942, 498)) : 2
The minimum distance from car: 2 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 400),
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

```
8%|           | 3/39 [00:07<01:35,  2.65s/it]
```

```
The times for each task are: [0.929864, 1.44083, 0.652086, 0.487456, 0.401209] with:
```

Minimum: 0.401209 Maximum: 1.44083 Average: 0.7823 seconds

```
Number of cars 2
bbox:car_number ((1045, 400), (1180, 485)) : 1
The minimum distance from car: 1 is 8.602325267042627
bbox:car_number ((834, 423), (942, 507)) : 2
The minimum distance from car: 2 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

10% | 4/39 [00:10<01:32, 2.63s/it]

The times for each task are: [0.90741, 1.322238, 0.66811, 0.327983, 0.472628] with:

Minimum: 0.327983 Maximum: 1.322238 Average: 0.7397 seconds

```
Number of cars 2
bbox:car_number ((1030, 400), (1181, 495)) : 1
The minimum distance from car: 1 is 8.602325267042627
bbox:car_number ((835, 419), (942, 507)) : 2
The minimum distance from car: 2 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

13% | 5/39 [00:13<01:31, 2.70s/it]

The times for each task are: [0.942607, 0.630681, 1.52879, 0.417608, 0.370394] with:

Minimum: 0.370394 Maximum: 1.52879 Average: 0.778 seconds

```
Number of cars 2
bbox:car_number ((1030, 400), (1182, 497)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((835, 415), (942, 507)) : 2
The minimum distance from car: 2 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
```

```
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419),
Length of task list: 5
Number of processes used: 3
```

15% | 6/39 [00:15<01:27, 2.66s/it]

The times for each task are: [0.847568, 0.642921, 1.557829, 0.429258, 0.282547] with:

Minimum: 0.282547 Maximum: 1.557829 Average: 0.752 seconds

```
Number of cars 2
bbox:car_number ((1027, 400), (1185, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((835, 415), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419),
Length of task list: 5
Number of processes used: 3
```

18% | 7/39 [00:18<01:24, 2.64s/it]

The times for each task are: [0.62285, 0.99789, 1.405536, 0.380485, 0.298787] with:

Minimum: 0.298787 Maximum: 1.405536 Average: 0.7411 seconds

```
Number of cars 2
bbox:car_number ((1027, 400), (1185, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((834, 415), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419),
Length of task list: 5
Number of processes used: 3
```

21% | 8/39 [00:21<01:22, 2.67s/it]

The times for each task are: [0.606929, 1.411273, 0.953428, 0.435549, 0.342002] with:

Minimum: 0.342002 Maximum: 1.411273 Average: 0.7498 seconds

Number of cars 2

bbox:car_number ((1021, 400), (1185, 497)) : 1

The minimum distance from car: 1 is 3.1622776601683795

bbox:car_number ((834, 415), (942, 507)) : 2

The minimum distance from car: 2 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (

Length of task list: 5

Number of processes used: 3

23% | 9/39 [00:24<01:20, 2.67s/it]

The times for each task are: [0.831028, 1.376868, 0.627066, 0.396096, 0.43723] with:

Minimum: 0.396096 Maximum: 1.376868 Average: 0.7337 seconds

Number of cars 2

bbox:car_number ((1027, 400), (1185, 498)) : 1

The minimum distance from car: 1 is 3.1622776601683795

bbox:car_number ((835, 415), (942, 507)) : 2

The minimum distance from car: 2 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (

Length of task list: 5

Number of processes used: 3

26% | 10/39 [00:26<01:17, 2.69s/it]

The times for each task are: [0.625631, 1.337557, 0.888388, 0.384874, 0.277285] with:

Minimum: 0.277285 Maximum: 1.337557 Average: 0.7027 seconds

Number of cars 2

bbox:car_number ((1027, 400), (1185, 498)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((834, 415), (942, 507)) : 2

The minimum distance from car: 2 is 0.0

```
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((834, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

28%| 11/39 [00:29<01:14, 2.67s/it]

The times for each task are: [0.870646, 0.614765, 1.432761, 0.291674, 0.369841] with:

Minimum: 0.291674 Maximum: 1.432761 Average: 0.7159 seconds

```
Number of cars 2
bbox:car_number ((1021, 400), (1185, 498)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((834, 415), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((834, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

31%| 12/39 [00:31<01:11, 2.65s/it]

The times for each task are: [1.388066, 0.83954, 0.655634, 0.429662, 0.387393] with:

Minimum: 0.387393 Maximum: 1.388066 Average: 0.7401 seconds

```
Number of cars 2
bbox:car_number ((1027, 400), (1197, 498)) : 1
The minimum distance from car: 1 is 9.0
bbox:car_number ((834, 415), (942, 507)) : 2
The minimum distance from car: 2 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((834, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

33% | 13/39 [00:34<01:09, 2.66s/it]

The times for each task are: [1.33485, 0.995031, 0.694491, 0.42721, 0.373733] with:

Minimum: 0.373733 Maximum: 1.33485 Average: 0.7651 seconds

Number of cars 3

bbox:car_number ((834, 400), (942, 507)) : 1

The minimum distance from car: 2 is 8.0

bbox:car_number ((1027, 400), (1208, 498)) : 2

The minimum distance from car: 1 is 5.0

bbox:car_number ((1210, 430), (1211, 477)) : 3

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

Length of task list: 5

Number of processes used: 3

36% | 14/39 [00:37<01:06, 2.67s/it]

The times for each task are: [0.946302, 0.658824, 1.446223, 0.294479, 0.512523] with:

Minimum: 0.294479 Maximum: 1.446223 Average: 0.7717 seconds

Number of cars 3

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 2 is 5.656854249492381

bbox:car_number ((1021, 400), (1208, 498)) : 2

The minimum distance from car: 1 is 3.0

bbox:car_number ((1210, 430), (1211, 477)) : 3

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

Length of task list: 5

Number of processes used: 3

38% | 15/39 [00:40<01:04, 2.67s/it]

The times for each task are: [0.879337, 0.676856, 1.483309, 0.285474, 0.467138] with:

Minimum: 0.285474 Maximum: 1.483309 Average: 0.7584 seconds

Number of cars 2

```
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 2 is 5.656854249492381
bbox:car_number ((1030, 400), (1211, 512)) : 2
The minimum distance from car: 1 is 9.219544457292887
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

41% | 16/39 [00:42<01:01, 2.66s/it]

The times for each task are: [0.914406, 0.583191, 0.398868, 1.670206, 0.306998] with:
Minimum: 0.306998 Maximum: 1.670206 Average: 0.7747 seconds

```
Number of cars 3
bbox:car_number ((835, 400), (942, 507)) : 1
The minimum distance from car: 2 is 0.0
bbox:car_number ((1021, 400), (1208, 498)) : 2
The minimum distance from car: 1 is 9.219544457292887
bbox:car_number ((1210, 427), (1211, 462)) : 3
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

44% | 17/39 [00:45<00:58, 2.67s/it]

The times for each task are: [0.968753, 0.624939, 1.402876, 0.376544, 0.286899] with:
Minimum: 0.286899 Maximum: 1.402876 Average: 0.732 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 2 is 0.0
bbox:car_number ((1044, 400), (1208, 498)) : 2
The minimum distance from car: 1 is 12.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
46%|     | 18/39 [00:48<00:56,  2.68s/it]
```

```
The times for each task are: [0.848846, 0.612733, 1.384577, 0.295021, 0.377502] with:
```

```
Minimum: 0.295021 Maximum: 1.384577 Average: 0.7037 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (942, 512)) : 1
The minimum distance from car: 2 is 3.0
bbox:car_number ((1060, 400), (1212, 498)) : 2
The minimum distance from car: 1 is 10.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419),
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
Length of task list: 5
Number of processes used: 3
```

```
49%|     | 19/39 [00:50<00:53,  2.68s/it]
```

```
The times for each task are: [0.866372, 0.668447, 1.346618, 0.442582, 0.335892] with:
```

```
Minimum: 0.335892 Maximum: 1.346618 Average: 0.732 seconds
```

```
Number of cars 2
bbox:car_number ((836, 400), (934, 512)) : 1
The minimum distance from car: 2 is 3.0
bbox:car_number ((1046, 400), (1214, 498)) : 2
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419),
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
Length of task list: 5
Number of processes used: 3
```

```
51%|     | 20/39 [00:53<00:50,  2.68s/it]
```

```
The times for each task are: [0.574875, 1.348516, 0.856636, 0.395979, 0.286415] with:
```

Minimum: 0.286415 Maximum: 1.348516 Average: 0.6925 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (942, 512)) : 1
The minimum distance from car: 2 is 4.0
bbox:car_number ((1046, 400), (1214, 498)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

54% | 21/39 [00:56<00:48, 2.67s/it]

The times for each task are: [0.598115, 1.407949, 0.976728, 0.450001, 0.292972] with:

Minimum: 0.292972 Maximum: 1.407949 Average: 0.7452 seconds

```
Number of cars 2
bbox:car_number ((1060, 400), (1214, 498)) : 1
The minimum distance from car: 1 is 7.0
bbox:car_number ((837, 415), (932, 507)) : 2
The minimum distance from car: 2 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
Length of task list: 5
Number of processes used: 3
```

56% | 22/39 [00:58<00:45, 2.68s/it]

The times for each task are: [0.644679, 0.910338, 1.569772, 0.296293, 0.485798] with:

Minimum: 0.296293 Maximum: 1.569772 Average: 0.7814 seconds

```
Number of cars 3
bbox:car_number ((1048, 400), (1219, 498)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((835, 419), (934, 507)) : 2
The minimum distance from car: 2 is 2.0
bbox:car_number ((552, 476), (586, 522)) : 3
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419),
Length of task list: 5
Number of processes used: 3
```

59% | 23/39 [01:01<00:42, 2.69s/it]

The times for each task are: [1.363287, 1.051237, 0.670914, 0.392734, 0.331129] with:

Minimum: 0.331129 Maximum: 1.363287 Average: 0.7619 seconds

```
Number of cars 3
bbox:car_number ((1060, 400), (1219, 498)) : 1
The minimum distance from car: 1 is 6.0
bbox:car_number ((835, 419), (934, 507)) : 2
The minimum distance from car: 2 is 0.0
bbox:car_number ((552, 476), (597, 522)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419),
Length of task list: 5
Number of processes used: 3
```

62% | 24/39 [01:04<00:40, 2.67s/it]

The times for each task are: [1.33644, 0.589694, 0.852494, 0.402114, 0.295208] with:

Minimum: 0.295208 Maximum: 1.33644 Average: 0.6952 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (934, 498)) : 1
The minimum distance from car: 2 is 14.035668847618199
bbox:car_number ((1046, 400), (1230, 498)) : 2
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419),
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40
Length of task list: 5
Number of processes used: 3
```

64%| 25/39 [01:06<00:37, 2.67s/it]

The times for each task are: [0.939783, 1.340263, 0.694845, 0.39046, 0.340342] with:

Minimum: 0.340342 Maximum: 1.340263 Average: 0.7411 seconds

Number of cars 2

bbox:car_number ((836, 400), (932, 497)) : 1

The minimum distance from car: 2 is 1.4142135623730951

bbox:car_number ((1051, 400), (1230, 498)) : 2

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

Length of task list: 5

Number of processes used: 3

67%| 26/39 [01:09<00:34, 2.68s/it]

The times for each task are: [1.015949, 1.372161, 0.6466, 0.285762, 0.38278] with:

Minimum: 0.285762 Maximum: 1.372161 Average: 0.7407 seconds

Number of cars 2

bbox:car_number ((836, 400), (934, 497)) : 1

The minimum distance from car: 2 is 1.0

bbox:car_number ((1051, 400), (1219, 498)) : 2

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

Length of task list: 5

Number of processes used: 3

69%| 27/39 [01:12<00:32, 2.67s/it]

The times for each task are: [1.382256, 0.598733, 1.020129, 0.295576, 0.378721] with:

Minimum: 0.295576 Maximum: 1.382256 Average: 0.7351 seconds

Number of cars 2

bbox:car_number ((836, 400), (934, 497)) : 1

The minimum distance from car: 2 is 0.0

```
bbox:car_number ((1048, 400), (1219, 498)) : 2
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

72%| 28/39 [01:14<00:29, 2.67s/it]

The times for each task are: [0.849248, 0.745511, 0.440508, 1.62095, 0.293356] with:

Minimum: 0.293356 Maximum: 1.62095 Average: 0.7899 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 2 is 5.830951894845301
bbox:car_number ((1048, 400), (1230, 498)) : 2
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

74%| 29/39 [01:17<00:26, 2.67s/it]

The times for each task are: [0.998672, 1.414241, 0.663824, 0.296369, 0.437476] with:

Minimum: 0.296369 Maximum: 1.414241 Average: 0.7621 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (934, 498)) : 1
The minimum distance from car: 2 is 5.656854249492381
bbox:car_number ((1051, 400), (1242, 498)) : 2
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

77%| | 30/39 [01:20<00:24, 2.67s/it]

The times for each task are: [0.863702, 0.737958, 0.386517, 1.540543, 0.328627] with:

Minimum: 0.328627 Maximum: 1.540543 Average: 0.7715 seconds

Number of cars 2

bbox:car_number ((856, 400), (934, 498)) : 1

The minimum distance from car: 2 is 11.0

bbox:car_number ((1051, 400), (1241, 498)) : 2

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

Length of task list: 5

Number of processes used: 3

79%| | 31/39 [01:22<00:21, 2.67s/it]

The times for each task are: [0.617986, 0.856382, 1.372022, 0.318547, 0.49019] with:

Minimum: 0.318547 Maximum: 1.372022 Average: 0.731 seconds

Number of cars 2

bbox:car_number ((1065, 400), (1182, 495)) : 1

The minimum distance from car: 1 is 23.08679276123039

bbox:car_number ((859, 427), (917, 485)) : 2

The minimum distance from car: 2 is 9.899494936611665

totalCars: 2

Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]

totalCars: 2

Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 507))]

Length of task list: 5

Number of processes used: 3

82%| | 32/39 [01:25<00:18, 2.67s/it]

The times for each task are: [0.611672, 1.490271, 0.864471, 0.394404, 0.298742] with:

Minimum: 0.298742 Maximum: 1.490271 Average: 0.7319 seconds

```
Number of cars 1
bbox:car_number ((1075, 400), (1230, 497)) : 1
The minimum distance from car: 1 is 29.017236257093817
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

85%| 33/39 [01:28<00:16, 2.68s/it]

The times for each task are: [0.96347, 0.605246, 1.437853, 0.473364, 0.299852] with:

Minimum: 0.299852 Maximum: 1.437853 Average: 0.756 seconds

```
Number of cars 2
bbox:car_number ((1102, 400), (1211, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((859, 427), (931, 498)) : 2
The minimum distance from car: 2 is 9.219544457292887
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
totalCars: 2
Car Number: 2 Car Positions: [((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (943, 507))]
Length of task list: 5
Number of processes used: 3
```

87%| 34/39 [01:31<00:13, 2.68s/it]

The times for each task are: [0.869441, 0.69097, 1.343109, 0.360879, 0.483122] with:

Minimum: 0.360879 Maximum: 1.343109 Average: 0.7495 seconds

```
Number of cars 1
bbox:car_number ((1113, 400), (1214, 498)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 415), (1162, 495))]
Length of task list: 5
Number of processes used: 3
```

90%| 35/39 [01:33<00:10, 2.68s/it]

The times for each task are: [1.347884, 0.616456, 0.874866, 0.296287, 0.485907] with:

Minimum: 0.296287 Maximum: 1.347884 Average: 0.7243 seconds

Number of cars 1
bbox:car_number ((1067, 400), (1227, 498)) : 1
The minimum distance from car: 1 is 16.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40

Length of task list: 5
Number of processes used: 3

92%|| 36/39 [01:36<00:08, 2.68s/it]

The times for each task are: [0.644468, 1.498314, 1.010173, 0.388489, 0.301176] with:

Minimum: 0.301176 Maximum: 1.498314 Average: 0.7685 seconds

Number of cars 1
bbox:car_number ((1067, 400), (1230, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40

Length of task list: 5
Number of processes used: 3

95%|| 37/39 [01:39<00:05, 2.68s/it]

The times for each task are: [0.840043, 0.615289, 0.284363, 1.366385, 0.391259] with:

Minimum: 0.284363 Maximum: 1.366385 Average: 0.6995 seconds

Number of cars 1
bbox:car_number ((1067, 400), (1238, 498)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 40

Length of task list: 5
Number of processes used: 3

97%|| 38/39 [01:41<00:02, 2.68s/it]

The times for each task are: [0.895706, 1.656878, 0.627072, 0.465827, 0.355755] with:

Minimum: 0.355755 Maximum: 1.656878 Average: 0.8002 seconds

```

Number of cars 1
bbox:car_number ((1067, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 400), (1181, 495)), ((1045, 400), (942, 507)), ((834, 419), (942, 507)), ((834, 419), (942, 498)), ((834, 423), (942, 498))]

[MoviePy] Done.
[MoviePy] >>> Video ready: project_videos_output/test_video_after_process_without_vicinity9.mp4

CPU times: user 3.52 s, sys: 39.8 s, total: 43.3 s
Wall time: 1min 43s

In [162]: Car.totalCars

Out[162]: 2

In [163]: for car in allCars:
    print(car.getAllPositions())
    print(car.detected)

[((1030, 415), (1162, 495)), ((1030, 415), (1162, 495)), ((1030, 400), (1181, 495)), ((1045, 400), (942, 507)))
True
[((834, 419), (942, 507)), ((834, 419), (942, 507)), ((835, 419), (942, 498)), ((834, 423), (942, 498))]
True

In [164]: vicinity_check = 1
vicinity_print = 0

Car.totalCars = 0
allCars= []

project_video_output_no_vicinity = 'project_videos_output/project_video_after_process_no_vicinity'
## To speed up the testing process you may want to try your pipeline on a shorter subclip
## To do so add .subclip(start_second,end_second) to the end of the line below
## Where start_second and end_second are integer values representing the start and end of the clip
## You may also uncomment the following line for a subclip of the first 5 seconds
##clip1 = VideoFileClip("test_videos/solidWhiteRight.mp4").subclip(0,5)
clip1 = VideoFileClip("project_video/project_video_with_lane.mp4")
white_clip = clip1.fl_image(process_image) #NOTE: this function expects color images!!
%time white_clip.write_videofile(project_video_output, audio=False)

Length of task list: 5
Number of processes used: 3

```

The times for each task are: [0.615372, 1.441399, 0.962669, 0.401613, 0.294302] with:

Minimum: 0.294302 Maximum: 1.441399 Average: 0.7431 seconds

Number of cars 0

[MoviePy] >>> Building video project_videos_output/project_video_after_process_without_vicitniy

[MoviePy] Writing video project_videos_output/project_video_after_process_without_vicitniy9.mp4

0% | 0/1261 [00:00<?, ?it/s]

Length of task list: 5

Number of processes used: 3

0% | 1/1261 [00:02<54:09, 2.58s/it]

The times for each task are: [0.633195, 0.917185, 1.696608, 0.311119, 0.407725] with:

Minimum: 0.311119 Maximum: 1.696608 Average: 0.7932 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

0% | 2/1261 [00:05<53:18, 2.54s/it]

The times for each task are: [0.997882, 1.318427, 0.690666, 0.446959, 0.279855] with:

Minimum: 0.279855 Maximum: 1.318427 Average: 0.7468 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

0% | 3/1261 [00:07<52:13, 2.49s/it]

The times for each task are: [0.646933, 1.587731, 0.995111, 0.427699, 0.38265] with:

Minimum: 0.38265 Maximum: 1.587731 Average: 0.808 seconds

Number of cars 1

bbox:car_number ((565, 460), (611, 507)) : 1

Length of task list: 5
Number of processes used: 3

0% | 4/1261 [00:09<51:33, 2.46s/it]

The times for each task are: [0.712066, 1.437404, 0.924055, 0.298631, 0.455808] with:

Minimum: 0.298631 Maximum: 1.437404 Average: 0.7656 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

0% | 5/1261 [00:12<52:34, 2.51s/it]

The times for each task are: [0.849007, 0.615717, 1.612803, 0.403678, 0.290112] with:

Minimum: 0.290112 Maximum: 1.612803 Average: 0.7543 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

0% | 6/1261 [00:15<52:35, 2.51s/it]

The times for each task are: [0.606593, 0.973369, 1.397649, 0.482131, 0.391023] with:

Minimum: 0.391023 Maximum: 1.397649 Average: 0.7702 seconds

Number of cars 1
bbox:car_number ((565, 475), (597, 492)) : 1
Length of task list: 5
Number of processes used: 3

1% | 7/1261 [00:17<52:27, 2.51s/it]

The times for each task are: [1.448725, 0.639691, 0.919535, 0.292305, 0.395739] with:

Minimum: 0.292305 Maximum: 1.448725 Average: 0.7392 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

1% | 8/1261 [00:19<51:56, 2.49s/it]

The times for each task are: [0.907432, 1.413309, 0.69963, 0.474927, 0.294899] with:

Minimum: 0.294899 Maximum: 1.413309 Average: 0.758 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

1% | 9/1261 [00:22<52:15, 2.50s/it]

The times for each task are: [1.39542, 0.712569, 1.109435, 0.478065, 0.343338] with:

Minimum: 0.343338 Maximum: 1.39542 Average: 0.8078 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

1% | 10/1261 [00:25<52:47, 2.53s/it]

The times for each task are: [0.723871, 1.059752, 1.400559, 0.430893, 0.300152] with:

Minimum: 0.300152 Maximum: 1.400559 Average: 0.783 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

1% | 11/1261 [00:28<53:03, 2.55s/it]

The times for each task are: [0.912725, 0.628578, 1.418689, 0.297705, 0.518871] with:

Minimum: 0.297705 Maximum: 1.418689 Average: 0.7553 seconds

Number of cars 1
bbox:car_number ((565, 460), (597, 492)) : 1

```
Length of task list: 5
Number of processes used: 3
```

```
1%|           | 12/1261 [00:30<52:56,  2.54s/it]
```

The times for each task are: [0.578911, 1.383745, 1.083576, 0.301736, 0.399595] with:

Minimum: 0.301736 Maximum: 1.383745 Average: 0.7495 seconds

```
Number of cars 1
bbox:car_number ((552, 475), (582, 507)) : 1
Length of task list: 5
Number of processes used: 3
```

```
1%|           | 13/1261 [00:33<52:54,  2.54s/it]
```

The times for each task are: [0.855474, 1.392562, 0.637372, 0.31193, 0.385601] with:

Minimum: 0.31193 Maximum: 1.392562 Average: 0.7166 seconds

```
Number of cars 1
bbox:car_number ((552, 460), (597, 507)) : 1
Length of task list: 5
Number of processes used: 3
```

```
1%|           | 14/1261 [00:35<52:37,  2.53s/it]
```

The times for each task are: [0.635759, 1.367444, 0.930568, 0.397625, 0.29318] with:

Minimum: 0.29318 Maximum: 1.367444 Average: 0.7249 seconds

```
Number of cars 1
bbox:car_number ((550, 475), (582, 516)) : 1
Length of task list: 5
Number of processes used: 3
```

```
1%|           | 15/1261 [00:38<52:49,  2.54s/it]
```

The times for each task are: [1.404598, 0.733139, 0.909073, 0.390448, 0.338202] with:

Minimum: 0.338202 Maximum: 1.404598 Average: 0.7551 seconds

```
Number of cars 0
Length of task list: 5
Number of processes used: 3
```

```
1%|           | 16/1261 [00:40<52:46, 2.54s/it]
```

The times for each task are: [0.866612, 0.692635, 1.383268, 0.410884, 0.379776] with:
Minimum: 0.379776 Maximum: 1.383268 Average: 0.7466 seconds

```
Number of cars 0
Length of task list: 5
Number of processes used: 3
```

```
1%|           | 17/1261 [00:43<52:34, 2.54s/it]
```

The times for each task are: [0.898129, 1.38268, 0.64231, 0.284295, 0.379173] with:
Minimum: 0.284295 Maximum: 1.38268 Average: 0.7173 seconds

```
Number of cars 1
bbox:car_number ((552, 505), (552, 516)) : 1
Length of task list: 5
Number of processes used: 3
```

```
1%|           | 18/1261 [00:45<52:43, 2.55s/it]
```

The times for each task are: [0.889849, 1.400855, 0.753811, 0.390791, 0.285714] with:
Minimum: 0.285714 Maximum: 1.400855 Average: 0.7442 seconds

```
Number of cars 0
Length of task list: 5
Number of processes used: 3
```

```
2%|           | 19/1261 [00:48<52:46, 2.55s/it]
```

The times for each task are: [1.400715, 0.937959, 0.745294, 0.349844, 0.402637] with:
Minimum: 0.349844 Maximum: 1.400715 Average: 0.7673 seconds

```
Number of cars 0
Length of task list: 5
Number of processes used: 3
```

2%| | 20/1261 [00:51<53:07, 2.57s/it]

The times for each task are: [0.635659, 1.429567, 1.109597, 0.425581, 0.308479] with:
Minimum: 0.308479 Maximum: 1.429567 Average: 0.7818 seconds

```
Number of cars 0
Length of task list: 5
Number of processes used: 3
```

2%| | 21/1261 [00:54<53:17, 2.58s/it]

The times for each task are: [1.02749, 1.505571, 0.618007, 0.536108, 0.306909] with:
Minimum: 0.306909 Maximum: 1.505571 Average: 0.7988 seconds

```
Number of cars 1
bbox:car_number ((550, 460), (597, 518)) : 1
Length of task list: 5
Number of processes used: 3
```

2%| | 22/1261 [00:56<53:21, 2.58s/it]

The times for each task are: [0.963202, 0.584294, 1.579307, 0.299481, 0.392767] with:
Minimum: 0.299481 Maximum: 1.579307 Average: 0.7638 seconds

```
Number of cars 0
Length of task list: 5
Number of processes used: 3
```

2%| | 23/1261 [00:59<53:35, 2.60s/it]

The times for each task are: [0.846563, 0.627691, 1.381575, 0.403593, 0.381164] with:
Minimum: 0.381164 Maximum: 1.381575 Average: 0.7281 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 24/1261 [01:02<53:42, 2.61s/it]

The times for each task are: [0.872861, 0.68553, 1.553073, 0.360095, 0.47958] with:
Minimum: 0.360095 Maximum: 1.553073 Average: 0.7902 seconds

Number of cars 1
bbox:car_number ((1075, 415), (1086, 447)) : 1
Length of task list: 5
Number of processes used: 3

2%| 25/1261 [01:04<53:31, 2.60s/it]

The times for each task are: [0.86818, 1.579075, 0.622449, 0.400513, 0.334628] with:
Minimum: 0.334628 Maximum: 1.579075 Average: 0.761 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 26/1261 [01:07<53:19, 2.59s/it]

The times for each task are: [0.611367, 0.989012, 1.640974, 0.401544, 0.354] with:
Minimum: 0.354 Maximum: 1.640974 Average: 0.7994 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 27/1261 [01:09<53:11, 2.59s/it]

The times for each task are: [1.352331, 0.657088, 1.06335, 0.326266, 0.385006] with:
Minimum: 0.326266 Maximum: 1.352331 Average: 0.7568 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 28/1261 [01:12<53:16, 2.59s/it]

The times for each task are: [0.573452, 0.899851, 1.631966, 0.392511, 0.288333] with:
Minimum: 0.288333 Maximum: 1.631966 Average: 0.7572 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 29/1261 [01:15<53:28, 2.60s/it]

The times for each task are: [0.674893, 1.526731, 1.109351, 0.428418, 0.276886] with:
Minimum: 0.276886 Maximum: 1.526731 Average: 0.8033 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 30/1261 [01:18<53:25, 2.60s/it]

The times for each task are: [1.322257, 0.964134, 0.718663, 0.332414, 0.44559] with:
Minimum: 0.332414 Maximum: 1.322257 Average: 0.7566 seconds

Number of cars 1
bbox:car_number ((1045, 415), (1077, 447)) : 1
Length of task list: 5
Number of processes used: 3

2%| 31/1261 [01:20<53:18, 2.60s/it]

The times for each task are: [0.896689, 1.440462, 0.704753, 0.309961, 0.38403] with:
Minimum: 0.309961 Maximum: 1.440462 Average: 0.7472 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

3%| 32/1261 [01:23<53:25, 2.61s/it]

The times for each task are: [1.377784, 0.899644, 0.681443, 0.362952, 0.537241] with:
Minimum: 0.362952 Maximum: 1.377784 Average: 0.7718 seconds

Number of cars 2
bbox:car_number ((1120, 400), (1152, 447)) : 1
bbox:car_number ((1105, 415), (1105, 447)) : 2
Length of task list: 5
Number of processes used: 3

3%| 33/1261 [01:26<53:23, 2.61s/it]

The times for each task are: [0.87185, 0.62521, 1.612719, 0.300565, 0.468218] with:
Minimum: 0.300565 Maximum: 1.612719 Average: 0.7757 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

3%| 34/1261 [01:28<53:27, 2.61s/it]

The times for each task are: [0.907494, 0.648723, 1.440721, 0.448758, 0.303227] with:
Minimum: 0.303227 Maximum: 1.440721 Average: 0.7498 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

3%| 35/1261 [01:31<53:18, 2.61s/it]

The times for each task are: [0.927747, 0.62748, 1.412022, 0.296201, 0.390035] with:

Minimum: 0.296201 Maximum: 1.412022 Average: 0.7307 seconds

Number of cars 1
bbox:car_number ((552, 475), (597, 492)) : 1
Length of task list: 5
Number of processes used: 3

3% | 36/1261 [01:33<53:16, 2.61s/it]

The times for each task are: [1.018665, 0.623208, 1.381666, 0.446466, 0.291181] with:

Minimum: 0.291181 Maximum: 1.381666 Average: 0.7522 seconds

Number of cars 1
bbox:car_number ((552, 490), (567, 492)) : 1
Length of task list: 5
Number of processes used: 3

3% | 37/1261 [01:36<53:22, 2.62s/it]

The times for each task are: [0.610098, 0.887659, 0.499985, 1.445766, 0.326938] with:

Minimum: 0.326938 Maximum: 1.445766 Average: 0.7541 seconds

Number of cars 1
bbox:car_number ((1135, 415), (1152, 447)) : 1
Length of task list: 5
Number of processes used: 3

3% | 38/1261 [01:39<53:11, 2.61s/it]

The times for each task are: [0.917598, 0.702686, 1.41149, 0.432238, 0.368157] with:

Minimum: 0.368157 Maximum: 1.41149 Average: 0.7664 seconds

Number of cars 3
bbox:car_number ((1150, 400), (1167, 447)) : 1
bbox:car_number ((1179, 400), (1182, 447)) : 2
bbox:car_number ((1120, 415), (1122, 447)) : 3
Length of task list: 5
Number of processes used: 3

3%| | 39/1261 [01:41<53:15, 2.61s/it]

The times for each task are: [0.755909, 1.045618, 1.384896, 0.415978, 0.478754] with:

Minimum: 0.415978 Maximum: 1.384896 Average: 0.8162 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

3%| | 40/1261 [01:44<53:18, 2.62s/it]

The times for each task are: [0.592748, 0.901652, 1.667585, 0.388914, 0.337408] with:

Minimum: 0.337408 Maximum: 1.667585 Average: 0.7777 seconds

Number of cars 1

bbox:car_number ((552, 505), (567, 507)) : 1

Length of task list: 5

Number of processes used: 3

3%| | 41/1261 [01:47<53:14, 2.62s/it]

The times for each task are: [0.874626, 0.576329, 1.423758, 0.290597, 0.482318] with:

Minimum: 0.290597 Maximum: 1.423758 Average: 0.7295 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

3%| | 42/1261 [01:50<53:12, 2.62s/it]

The times for each task are: [0.625262, 1.062087, 1.419647, 0.442163, 0.307167] with:

Minimum: 0.307167 Maximum: 1.419647 Average: 0.7713 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

3%| | 43/1261 [01:52<53:13, 2.62s/it]

The times for each task are: [0.593718, 1.525247, 1.006066, 0.386368, 0.282729] with:

Minimum: 0.282729 Maximum: 1.525247 Average: 0.7588 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

3% | 44/1261 [01:55<53:08, 2.62s/it]

The times for each task are: [0.86445, 0.618528, 1.400045, 0.438744, 0.301985] with:

Minimum: 0.301985 Maximum: 1.400045 Average: 0.7248 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 45/1261 [01:58<53:09, 2.62s/it]

The times for each task are: [0.643084, 0.956916, 1.489383, 0.414781, 0.398851] with:

Minimum: 0.398851 Maximum: 1.489383 Average: 0.7806 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 46/1261 [02:00<53:04, 2.62s/it]

The times for each task are: [1.389471, 0.639981, 1.081079, 0.400833, 0.384112] with:

Minimum: 0.384112 Maximum: 1.389471 Average: 0.7791 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 47/1261 [02:02<52:56, 2.62s/it]

The times for each task are: [1.000709, 1.398106, 0.62912, 0.296893, 0.460913] with:

Minimum: 0.296893 Maximum: 1.398106 Average: 0.7571 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 48/1261 [02:05<52:56, 2.62s/it]

The times for each task are: [0.865063, 0.702087, 1.447662, 0.516918, 0.29169] with:

Minimum: 0.29169 Maximum: 1.447662 Average: 0.7647 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 49/1261 [02:08<52:56, 2.62s/it]

The times for each task are: [0.885191, 0.663723, 1.362142, 0.386315, 0.29707] with:

Minimum: 0.29707 Maximum: 1.362142 Average: 0.7189 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 50/1261 [02:11<53:05, 2.63s/it]

The times for each task are: [0.615627, 1.040638, 1.498418, 0.369465, 0.330506] with:

Minimum: 0.330506 Maximum: 1.498418 Average: 0.7709 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 51/1261 [02:14<53:06, 2.63s/it]

The times for each task are: [0.628754, 1.009763, 1.422504, 0.398659, 0.421884] with:

Minimum: 0.398659 Maximum: 1.422504 Average: 0.7763 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4%| 52/1261 [02:17<53:10, 2.64s/it]

The times for each task are: [1.417047, 0.66128, 1.023276, 0.366821, 0.409203] with:
Minimum: 0.366821 Maximum: 1.417047 Average: 0.7755 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4%| 53/1261 [02:19<53:06, 2.64s/it]

The times for each task are: [0.628409, 0.870836, 1.406941, 0.385688, 0.296277] with:
Minimum: 0.296277 Maximum: 1.406941 Average: 0.7176 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4%| 54/1261 [02:22<53:03, 2.64s/it]

The times for each task are: [0.720784, 1.441868, 0.925579, 0.290935, 0.448324] with:
Minimum: 0.290935 Maximum: 1.441868 Average: 0.7655 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4%| 55/1261 [02:25<53:01, 2.64s/it]

The times for each task are: [0.645717, 0.961602, 1.627529, 0.411187, 0.297532] with:
Minimum: 0.297532 Maximum: 1.627529 Average: 0.7887 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4% | 56/1261 [02:27<52:55, 2.63s/it]

The times for each task are: [0.856039, 0.666871, 1.63666, 0.413073, 0.298656] with:
Minimum: 0.298656 Maximum: 1.63666 Average: 0.7743 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 57/1261 [02:29<52:48, 2.63s/it]

The times for each task are: [0.982382, 0.665993, 1.461519, 0.409488, 0.280461] with:
Minimum: 0.280461 Maximum: 1.461519 Average: 0.76 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 58/1261 [02:32<52:49, 2.63s/it]

The times for each task are: [0.819221, 0.691463, 1.559463, 0.439186, 0.346433] with:
Minimum: 0.346433 Maximum: 1.559463 Average: 0.7712 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 59/1261 [02:35<52:52, 2.64s/it]

The times for each task are: [0.693081, 0.882843, 0.440908, 1.327734, 0.402145] with:
Minimum: 0.402145 Maximum: 1.327734 Average: 0.7493 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

5% | 60/1261 [02:38<52:52, 2.64s/it]

The times for each task are: [0.70217, 0.995437, 1.506911, 0.452196, 0.29516] with:

Minimum: 0.29516 Maximum: 1.506911 Average: 0.7904 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 61/1261 [02:41<52:50, 2.64s/it]

The times for each task are: [0.653619, 0.901934, 1.526962, 0.415632, 0.473191] with:

Minimum: 0.415632 Maximum: 1.526962 Average: 0.7943 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 62/1261 [02:43<52:48, 2.64s/it]

The times for each task are: [0.905388, 0.629211, 1.602955, 0.286942, 0.377671] with:

Minimum: 0.286942 Maximum: 1.602955 Average: 0.7604 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 63/1261 [02:46<52:51, 2.65s/it]

The times for each task are: [0.696786, 1.621266, 0.905022, 0.393313, 0.329729] with:

Minimum: 0.329729 Maximum: 1.621266 Average: 0.7892 seconds

Number of cars 0
Length of task list: 5

Number of processes used: 3

5%| | 64/1261 [02:49<52:46, 2.65s/it]

The times for each task are: [1.342777, 0.865882, 0.694939, 0.416639, 0.313055] with:

Minimum: 0.313055 Maximum: 1.342777 Average: 0.7267 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 65/1261 [02:52<52:51, 2.65s/it]

The times for each task are: [1.015766, 0.601047, 1.447265, 0.430662, 0.303565] with:

Minimum: 0.303565 Maximum: 1.447265 Average: 0.7597 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 66/1261 [02:54<52:46, 2.65s/it]

The times for each task are: [1.321771, 1.100205, 0.638054, 0.402745, 0.33421] with:

Minimum: 0.33421 Maximum: 1.321771 Average: 0.7594 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 67/1261 [02:57<52:44, 2.65s/it]

The times for each task are: [0.901644, 0.680956, 0.41289, 1.610443, 0.298216] with:

Minimum: 0.298216 Maximum: 1.610443 Average: 0.7808 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 68/1261 [03:00<52:39, 2.65s/it]

The times for each task are: [0.623608, 0.841924, 0.392665, 1.414158, 0.311905] with:

Minimum: 0.311905 Maximum: 1.414158 Average: 0.7169 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 69/1261 [03:02<52:38, 2.65s/it]

The times for each task are: [0.880114, 0.605938, 1.610703, 0.38997, 0.286078] with:

Minimum: 0.286078 Maximum: 1.610703 Average: 0.7546 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6%| | 70/1261 [03:05<52:33, 2.65s/it]

The times for each task are: [1.053956, 0.697346, 1.552076, 0.460346, 0.384282] with:

Minimum: 0.384282 Maximum: 1.552076 Average: 0.8296 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6%| | 71/1261 [03:08<52:32, 2.65s/it]

The times for each task are: [0.915187, 0.647101, 1.347581, 0.325853, 0.418586] with:

Minimum: 0.325853 Maximum: 1.347581 Average: 0.7309 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6%| | 72/1261 [03:10<52:28, 2.65s/it]

The times for each task are: [0.903149, 0.630141, 1.700647, 0.399017, 0.45831] with:

Minimum: 0.399017 Maximum: 1.700647 Average: 0.8183 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 73/1261 [03:13<52:25, 2.65s/it]

The times for each task are: [0.622446, 0.88489, 1.396533, 0.293566, 0.384616] with:

Minimum: 0.293566 Maximum: 1.396533 Average: 0.7164 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 74/1261 [03:15<52:22, 2.65s/it]

The times for each task are: [0.893236, 0.72817, 1.358902, 0.444124, 0.351424] with:

Minimum: 0.351424 Maximum: 1.358902 Average: 0.7552 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 75/1261 [03:18<52:21, 2.65s/it]

The times for each task are: [0.883998, 0.624164, 1.590799, 0.405477, 0.29532] with:

Minimum: 0.29532 Maximum: 1.590799 Average: 0.76 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 76/1261 [03:21<52:17, 2.65s/it]

The times for each task are: [0.872927, 1.408188, 0.647546, 0.299132, 0.441374] with:

Minimum: 0.299132 Maximum: 1.408188 Average: 0.7338 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 77/1261 [03:23<52:15, 2.65s/it]

The times for each task are: [0.927655, 0.627, 1.484378, 0.416996, 0.325061] with:

Minimum: 0.325061 Maximum: 1.484378 Average: 0.7562 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 78/1261 [03:26<52:13, 2.65s/it]

The times for each task are: [1.384144, 0.893785, 0.784041, 0.342235, 0.38347] with:

Minimum: 0.342235 Maximum: 1.384144 Average: 0.7575 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 79/1261 [03:29<52:14, 2.65s/it]

The times for each task are: [1.362298, 0.708056, 1.010692, 0.433744, 0.338561] with:

Minimum: 0.338561 Maximum: 1.362298 Average: 0.7707 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 80/1261 [03:32<52:13, 2.65s/it]

The times for each task are: [1.36623, 0.747949, 0.893335, 0.338229, 0.427487] with:

Minimum: 0.338229 Maximum: 1.36623 Average: 0.7546 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

6% | 81/1261 [03:34<52:10, 2.65s/it]

The times for each task are: [0.569358, 1.34209, 0.969812, 0.398042, 0.327736] with:
Minimum: 0.327736 Maximum: 1.34209 Average: 0.7214 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7% | 82/1261 [03:37<52:11, 2.66s/it]

The times for each task are: [0.61476, 1.344888, 0.944035, 0.306814, 0.434122] with:
Minimum: 0.306814 Maximum: 1.344888 Average: 0.7289 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7% | 83/1261 [03:40<52:06, 2.65s/it]

The times for each task are: [0.881166, 0.703807, 1.385751, 0.405599, 0.340707] with:
Minimum: 0.340707 Maximum: 1.385751 Average: 0.7434 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7% | 84/1261 [03:42<52:02, 2.65s/it]

The times for each task are: [0.883116, 0.676548, 1.343718, 0.292577, 0.440978] with:
Minimum: 0.292577 Maximum: 1.343718 Average: 0.7274 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 85/1261 [03:45<52:01, 2.65s/it]

The times for each task are: [0.654031, 0.990299, 1.495037, 0.398064, 0.324178] with:
Minimum: 0.324178 Maximum: 1.495037 Average: 0.7723 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 86/1261 [03:48<51:58, 2.65s/it]

The times for each task are: [0.908458, 0.576827, 1.449868, 0.284037, 0.389172] with:
Minimum: 0.284037 Maximum: 1.449868 Average: 0.7217 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 87/1261 [03:50<51:56, 2.65s/it]

The times for each task are: [0.722278, 0.885501, 1.445927, 0.386261, 0.286978] with:
Minimum: 0.286978 Maximum: 1.445927 Average: 0.7454 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 88/1261 [03:53<51:52, 2.65s/it]

The times for each task are: [0.889656, 0.659264, 1.686853, 0.452954, 0.392393] with:
Minimum: 0.392393 Maximum: 1.686853 Average: 0.8162 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

7% | 89/1261 [03:56<51:49, 2.65s/it]

The times for each task are: [0.717276, 0.961695, 1.400623, 0.296988, 0.451356] with:

Minimum: 0.296988 Maximum: 1.400623 Average: 0.7656 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7% | 90/1261 [03:58<51:47, 2.65s/it]

The times for each task are: [0.942289, 0.719476, 1.459991, 0.364874, 0.497162] with:

Minimum: 0.364874 Maximum: 1.459991 Average: 0.7968 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7% | 91/1261 [04:01<51:46, 2.65s/it]

The times for each task are: [1.621327, 0.699801, 1.022797, 0.54631, 0.384819] with:

Minimum: 0.384819 Maximum: 1.621327 Average: 0.855 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7% | 92/1261 [04:04<51:45, 2.66s/it]

The times for each task are: [0.910182, 1.639447, 0.735508, 0.363031, 0.460537] with:

Minimum: 0.363031 Maximum: 1.639447 Average: 0.8217 seconds

Number of cars 0
Length of task list: 5

Number of processes used: 3

7%| | 93/1261 [04:07<51:42, 2.66s/it]

The times for each task are: [0.876462, 0.622666, 1.446446, 0.393337, 0.284268] with:

Minimum: 0.284268 Maximum: 1.446446 Average: 0.7246 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

7%| | 94/1261 [04:09<51:39, 2.66s/it]

The times for each task are: [0.681433, 0.896451, 1.634126, 0.50702, 0.334743] with:

Minimum: 0.334743 Maximum: 1.634126 Average: 0.8108 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 95/1261 [04:12<51:36, 2.66s/it]

The times for each task are: [0.616857, 0.938787, 1.669262, 0.390009, 0.308919] with:

Minimum: 0.308919 Maximum: 1.669262 Average: 0.7848 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 96/1261 [04:14<51:29, 2.65s/it]

The times for each task are: [0.981354, 0.663503, 1.434105, 0.40489, 0.336877] with:

Minimum: 0.336877 Maximum: 1.434105 Average: 0.7641 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 97/1261 [04:17<51:26, 2.65s/it]

The times for each task are: [0.969262, 1.393646, 0.687132, 0.380304, 0.292151] with:

Minimum: 0.292151 Maximum: 1.393646 Average: 0.7445 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 98/1261 [04:19<51:22, 2.65s/it]

The times for each task are: [0.882255, 0.612735, 1.585409, 0.403131, 0.284327] with:

Minimum: 0.284327 Maximum: 1.585409 Average: 0.7536 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 99/1261 [04:22<51:20, 2.65s/it]

The times for each task are: [0.623953, 0.909633, 1.379364, 0.328426, 0.530448] with:

Minimum: 0.328426 Maximum: 1.379364 Average: 0.7544 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 100/1261 [04:25<51:18, 2.65s/it]

The times for each task are: [0.916545, 0.742201, 1.436056, 0.57037, 0.389083] with:

Minimum: 0.389083 Maximum: 1.436056 Average: 0.8109 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 101/1261 [04:27<51:13, 2.65s/it]

The times for each task are: [0.664401, 0.917918, 1.572807, 0.351122, 0.481981] with:

Minimum: 0.351122 Maximum: 1.572807 Average: 0.7976 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 102/1261 [04:30<51:10, 2.65s/it]

The times for each task are: [0.625757, 1.003359, 1.339833, 0.294713, 0.379687] with:

Minimum: 0.294713 Maximum: 1.339833 Average: 0.7287 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 103/1261 [04:32<51:06, 2.65s/it]

The times for each task are: [0.629511, 1.413975, 1.053296, 0.291111, 0.391197] with:

Minimum: 0.291111 Maximum: 1.413975 Average: 0.7558 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 104/1261 [04:35<51:04, 2.65s/it]

The times for each task are: [0.667229, 0.873186, 1.393799, 0.29896, 0.400588] with:

Minimum: 0.29896 Maximum: 1.393799 Average: 0.7268 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 105/1261 [04:37<51:00, 2.65s/it]

The times for each task are: [0.67113, 1.360582, 1.070153, 0.286329, 0.375099] with:

Minimum: 0.286329 Maximum: 1.360582 Average: 0.7527 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 106/1261 [04:40<50:57, 2.65s/it]

The times for each task are: [0.871856, 1.37059, 0.742527, 0.394355, 0.289817] with:

Minimum: 0.289817 Maximum: 1.37059 Average: 0.7338 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 107/1261 [04:43<50:54, 2.65s/it]

The times for each task are: [0.588109, 1.042806, 1.431882, 0.450154, 0.295321] with:

Minimum: 0.295321 Maximum: 1.431882 Average: 0.7617 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

9% | 108/1261 [04:45<50:53, 2.65s/it]

The times for each task are: [0.888034, 1.455337, 0.697162, 0.443975, 0.344924] with:

Minimum: 0.344924 Maximum: 1.455337 Average: 0.7659 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

9% | 109/1261 [04:48<50:50, 2.65s/it]

The times for each task are: [0.569986, 0.879417, 1.396087, 0.299196, 0.386333] with:

Minimum: 0.299196 Maximum: 1.396087 Average: 0.7062 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9%| 110/1261 [04:51<50:46, 2.65s/it]

The times for each task are: [0.837878, 1.354028, 0.673834, 0.290294, 0.394368] with:
Minimum: 0.290294 Maximum: 1.354028 Average: 0.7101 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9%| 111/1261 [04:54<50:46, 2.65s/it]

The times for each task are: [0.726668, 0.999907, 1.386805, 0.465275, 0.324] with:
Minimum: 0.324 Maximum: 1.386805 Average: 0.7805 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9%| 112/1261 [04:56<50:42, 2.65s/it]

The times for each task are: [0.886681, 0.653152, 0.509793, 1.67746, 0.314351] with:
Minimum: 0.314351 Maximum: 1.67746 Average: 0.8083 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9%| 113/1261 [04:59<50:41, 2.65s/it]

The times for each task are: [1.369803, 0.918976, 0.617119, 0.388774, 0.299544] with:
Minimum: 0.299544 Maximum: 1.369803 Average: 0.7188 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 114/1261 [05:02<50:38, 2.65s/it]

The times for each task are: [0.885515, 0.641572, 1.415905, 0.445828, 0.278484] with:
Minimum: 0.278484 Maximum: 1.415905 Average: 0.7335 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 115/1261 [05:04<50:35, 2.65s/it]

The times for each task are: [0.628453, 0.902419, 1.680826, 0.40215, 0.353045] with:
Minimum: 0.353045 Maximum: 1.680826 Average: 0.7934 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 116/1261 [05:07<50:30, 2.65s/it]

The times for each task are: [0.903432, 0.703788, 0.408454, 1.570053, 0.304174] with:
Minimum: 0.304174 Maximum: 1.570053 Average: 0.778 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 117/1261 [05:09<50:26, 2.65s/it]

The times for each task are: [0.586036, 0.902706, 1.625823, 0.392672, 0.447285] with:
Minimum: 0.392672 Maximum: 1.625823 Average: 0.7909 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

9% | 118/1261 [05:12<50:25, 2.65s/it]

The times for each task are: [0.618728, 1.398072, 0.978378, 0.306168, 0.400256] with:

Minimum: 0.306168 Maximum: 1.398072 Average: 0.7403 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 119/1261 [05:14<50:21, 2.65s/it]

The times for each task are: [1.006967, 0.621114, 1.67853, 0.519454, 0.363136] with:

Minimum: 0.363136 Maximum: 1.67853 Average: 0.8378 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

10% | 120/1261 [05:17<50:18, 2.65s/it]

The times for each task are: [0.85535, 0.623134, 1.425931, 0.290289, 0.402089] with:

Minimum: 0.290289 Maximum: 1.425931 Average: 0.7194 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

10% | 121/1261 [05:19<50:14, 2.64s/it]

The times for each task are: [0.908725, 0.712585, 1.455737, 0.436224, 0.357534] with:

Minimum: 0.357534 Maximum: 1.455737 Average: 0.7742 seconds

Number of cars 0
Length of task list: 5

Number of processes used: 3

10% | 122/1261 [05:22<50:11, 2.64s/it]

The times for each task are: [0.916932, 1.373374, 0.6641, 0.394001, 0.282807] with:

Minimum: 0.282807 Maximum: 1.373374 Average: 0.7262 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 123/1261 [05:25<50:09, 2.64s/it]

The times for each task are: [0.618272, 0.904329, 1.515136, 0.426241, 0.378638] with:

Minimum: 0.378638 Maximum: 1.515136 Average: 0.7685 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 124/1261 [05:28<50:08, 2.65s/it]

The times for each task are: [0.656555, 1.016743, 1.553244, 0.423651, 0.471604] with:

Minimum: 0.423651 Maximum: 1.553244 Average: 0.8244 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 125/1261 [05:31<50:08, 2.65s/it]

The times for each task are: [0.633157, 1.435318, 1.066167, 0.42165, 0.392658] with:

Minimum: 0.392658 Maximum: 1.435318 Average: 0.7898 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 126/1261 [05:33<50:05, 2.65s/it]

The times for each task are: [0.887793, 0.713881, 1.393668, 0.343794, 0.394369] with:

Minimum: 0.343794 Maximum: 1.393668 Average: 0.7467 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 127/1261 [05:36<50:02, 2.65s/it]

The times for each task are: [0.878901, 0.704837, 0.412386, 1.600898, 0.293496] with:

Minimum: 0.293496 Maximum: 1.600898 Average: 0.7781 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 128/1261 [05:39<50:01, 2.65s/it]

The times for each task are: [0.967041, 1.364192, 0.646695, 0.403576, 0.344026] with:

Minimum: 0.344026 Maximum: 1.364192 Average: 0.7451 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 129/1261 [05:41<50:00, 2.65s/it]

The times for each task are: [0.852881, 0.63613, 1.655123, 0.311573, 0.404194] with:

Minimum: 0.311573 Maximum: 1.655123 Average: 0.772 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 130/1261 [05:44<49:56, 2.65s/it]

The times for each task are: [0.927361, 0.658835, 1.652883, 0.295407, 0.402601] with:

Minimum: 0.295407 Maximum: 1.652883 Average: 0.7874 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 131/1261 [05:47<49:53, 2.65s/it]

The times for each task are: [0.662498, 0.878487, 1.489348, 0.427699, 0.286671] with:

Minimum: 0.286671 Maximum: 1.489348 Average: 0.7489 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 132/1261 [05:49<49:48, 2.65s/it]

The times for each task are: [0.650592, 0.919486, 0.413295, 1.46601, 0.292308] with:

Minimum: 0.292308 Maximum: 1.46601 Average: 0.7483 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 133/1261 [05:52<49:45, 2.65s/it]

The times for each task are: [1.456612, 1.010126, 0.739717, 0.404631, 0.289037] with:

Minimum: 0.289037 Maximum: 1.456612 Average: 0.78 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 134/1261 [05:54<49:42, 2.65s/it]

The times for each task are: [1.421994, 0.64554, 0.897582, 0.408118, 0.285532] with:

Minimum: 0.285532 Maximum: 1.421994 Average: 0.7318 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 135/1261 [05:57<49:38, 2.65s/it]

The times for each task are: [0.629726, 1.405404, 0.988704, 0.321713, 0.491452] with:

Minimum: 0.321713 Maximum: 1.405404 Average: 0.7674 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 136/1261 [06:00<49:39, 2.65s/it]

The times for each task are: [1.089206, 0.630732, 1.683392, 0.42855, 0.394597] with:

Minimum: 0.394597 Maximum: 1.683392 Average: 0.8453 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 137/1261 [06:02<49:37, 2.65s/it]

The times for each task are: [0.661141, 0.910505, 1.439686, 0.456405, 0.343563] with:

Minimum: 0.343563 Maximum: 1.439686 Average: 0.7623 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 138/1261 [06:05<49:35, 2.65s/it]

The times for each task are: [0.63306, 1.088763, 1.40445, 0.350514, 0.43283] with:

Minimum: 0.350514 Maximum: 1.40445 Average: 0.7819 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 139/1261 [06:08<49:32, 2.65s/it]

The times for each task are: [0.715159, 1.473431, 0.863986, 0.353191, 0.493112] with:
Minimum: 0.353191 Maximum: 1.473431 Average: 0.7798 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 140/1261 [06:10<49:30, 2.65s/it]

The times for each task are: [0.906973, 0.728046, 1.667022, 0.51886, 0.419509] with:
Minimum: 0.419509 Maximum: 1.667022 Average: 0.8481 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 141/1261 [06:13<49:27, 2.65s/it]

The times for each task are: [0.861816, 0.625651, 1.605692, 0.399655, 0.340423] with:
Minimum: 0.340423 Maximum: 1.605692 Average: 0.7666 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 142/1261 [06:16<49:26, 2.65s/it]

The times for each task are: [0.621351, 0.914896, 1.487206, 0.33482, 0.432448] with:
Minimum: 0.33482 Maximum: 1.487206 Average: 0.7581 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 143/1261 [06:19<49:25, 2.65s/it]

The times for each task are: [0.660388, 1.450119, 1.047667, 0.422477, 0.293393] with:
Minimum: 0.293393 Maximum: 1.450119 Average: 0.7748 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 144/1261 [06:21<49:21, 2.65s/it]

The times for each task are: [1.030304, 0.661107, 1.643122, 0.338735, 0.39155] with:
Minimum: 0.338735 Maximum: 1.643122 Average: 0.813 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 145/1261 [06:24<49:18, 2.65s/it]

The times for each task are: [0.864448, 0.63585, 1.409199, 0.394396, 0.348021] with:
Minimum: 0.348021 Maximum: 1.409199 Average: 0.7304 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

12% | 146/1261 [06:27<49:15, 2.65s/it]

The times for each task are: [0.903487, 0.624593, 1.632748, 0.481054, 0.318544] with:
Minimum: 0.318544 Maximum: 1.632748 Average: 0.7921 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

12% | 147/1261 [06:29<49:15, 2.65s/it]

The times for each task are: [0.617602, 0.977613, 1.772943, 0.54463, 0.36454] with:

Minimum: 0.36454 Maximum: 1.772943 Average: 0.8555 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

12% | 148/1261 [06:32<49:12, 2.65s/it]

The times for each task are: [0.864825, 0.587072, 1.424261, 0.406969, 0.287681] with:

Minimum: 0.287681 Maximum: 1.424261 Average: 0.7142 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

12% | 149/1261 [06:35<49:07, 2.65s/it]

The times for each task are: [0.649429, 1.00743, 1.426562, 0.320763, 0.41833] with:

Minimum: 0.320763 Maximum: 1.426562 Average: 0.7645 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

12% | 150/1261 [06:37<49:04, 2.65s/it]

The times for each task are: [0.859764, 0.772689, 1.429327, 0.472994, 0.351512] with:

Minimum: 0.351512 Maximum: 1.429327 Average: 0.7773 seconds

Number of cars 0
Length of task list: 5

Number of processes used: 3

12%| 151/1261 [06:40<49:03, 2.65s/it]

The times for each task are: [0.914694, 0.714257, 0.354331, 0.513937, 1.485168] with:

Minimum: 0.354331 Maximum: 1.485168 Average: 0.7965 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12%| 152/1261 [06:43<49:00, 2.65s/it]

The times for each task are: [1.405651, 0.62585, 0.902275, 0.395688, 0.28375] with:

Minimum: 0.28375 Maximum: 1.405651 Average: 0.7226 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12%| 153/1261 [06:45<48:57, 2.65s/it]

The times for each task are: [0.629787, 0.908816, 1.483648, 0.391158, 0.291506] with:

Minimum: 0.291506 Maximum: 1.483648 Average: 0.741 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12%| 154/1261 [06:48<48:56, 2.65s/it]

The times for each task are: [0.838302, 1.655504, 0.686722, 0.463811, 0.288343] with:

Minimum: 0.288343 Maximum: 1.655504 Average: 0.7865 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12% | 155/1261 [06:51<48:52, 2.65s/it]

The times for each task are: [0.870107, 0.588248, 0.412211, 1.655701, 0.295053] with:

Minimum: 0.295053 Maximum: 1.655701 Average: 0.7643 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12% | 156/1261 [06:53<48:49, 2.65s/it]

The times for each task are: [0.628161, 0.961293, 1.54838, 0.346675, 0.396252] with:

Minimum: 0.346675 Maximum: 1.54838 Average: 0.7762 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12% | 157/1261 [06:56<48:46, 2.65s/it]

The times for each task are: [1.342464, 0.554238, 1.139185, 0.396925, 0.313646] with:

Minimum: 0.313646 Maximum: 1.342464 Average: 0.7493 seconds

Number of cars 1

bbox:car_number ((1175, 460), (1212, 498)) : 1

Length of task list: 5

Number of processes used: 3

13% | 158/1261 [06:58<48:43, 2.65s/it]

The times for each task are: [0.893399, 0.659959, 0.494973, 1.628199, 0.297051] with:

Minimum: 0.297051 Maximum: 1.628199 Average: 0.7947 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

13% | 159/1261 [07:01<48:38, 2.65s/it]

The times for each task are: [0.627266, 0.971862, 1.387325, 0.302466, 0.370271] with:

Minimum: 0.302466 Maximum: 1.387325 Average: 0.7318 seconds

Number of cars 1
bbox:car_number ((1217, 438), (1219, 459)) : 1
Length of task list: 5
Number of processes used: 3

13% | 160/1261 [07:03<48:36, 2.65s/it]

The times for each task are: [0.600072, 1.104096, 1.647479, 0.470951, 0.296033] with:

Minimum: 0.296033 Maximum: 1.647479 Average: 0.8237 seconds

Number of cars 1
bbox:car_number ((1204, 400), (1273, 478)) : 1
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478))]
Length of task list: 5
Number of processes used: 3

13% | 161/1261 [07:06<48:34, 2.65s/it]

The times for each task are: [0.861629, 0.604033, 1.624727, 0.510449, 0.28225] with:

Minimum: 0.28225 Maximum: 1.624727 Average: 0.7766 seconds

Number of cars 1
bbox:car_number ((1175, 400), (1273, 512)) : 1
The minimum distance from car: 1 is 22.02271554554524
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

13% | 162/1261 [07:09<48:32, 2.65s/it]

The times for each task are: [1.43587, 1.037788, 0.664514, 0.400676, 0.285752] with:

Minimum: 0.285752 Maximum: 1.43587 Average: 0.7649 seconds

Number of cars 1

```
bbox:car_number ((1183, 400), (1273, 497)) : 1
The minimum distance from car: 1 is 8.94427190999916
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13% | 163/1261 [07:12<48:30, 2.65s/it]

The times for each task are: [1.375047, 0.626407, 0.898301, 0.361235, 0.444632] with:
Minimum: 0.361235 Maximum: 1.375047 Average: 0.7411 seconds

```
Number of cars 1
bbox:car_number ((1175, 400), (1273, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13% | 164/1261 [07:14<48:27, 2.65s/it]

The times for each task are: [1.39128, 0.922565, 0.632664, 0.493851, 0.374148] with:
Minimum: 0.374148 Maximum: 1.39128 Average: 0.7629 seconds

```
Number of cars 1
bbox:car_number ((1175, 400), (1273, 512)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13% | 165/1261 [07:17<48:26, 2.65s/it]

The times for each task are: [0.606828, 0.983609, 1.372214, 0.445026, 0.325581] with:
Minimum: 0.325581 Maximum: 1.372214 Average: 0.7467 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1273, 512)) : 1
```

```
The minimum distance from car: 1 is 16.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13% | 166/1261 [07:20<48:23, 2.65s/it]

The times for each task are: [0.845845, 1.465522, 0.74388, 0.31437, 0.466225] with:

Minimum: 0.31437 Maximum: 1.465522 Average: 0.7672 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1257, 498)) : 1
The minimum distance from car: 1 is 10.63014581273465
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13% | 167/1261 [07:22<48:20, 2.65s/it]

The times for each task are: [0.917301, 1.581814, 0.68391, 0.480274, 0.33841] with:

Minimum: 0.33841 Maximum: 1.581814 Average: 0.8003 seconds

```
Number of cars 1
bbox:car_number ((1156, 400), (1257, 497)) : 1
The minimum distance from car: 1 is 6.082762530298219
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13% | 168/1261 [07:26<48:22, 2.66s/it]

The times for each task are: [0.725339, 0.997365, 1.475392, 0.48886, 0.367171] with:

Minimum: 0.367171 Maximum: 1.475392 Average: 0.8108 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1257, 498)) : 1
The minimum distance from car: 1 is 6.082762530298219
```

```
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13%| 169/1261 [07:28<48:21, 2.66s/it]

The times for each task are: [0.891614, 0.628349, 1.398728, 0.44244, 0.286669] with:

Minimum: 0.286669 Maximum: 1.398728 Average: 0.7296 seconds

```
Number of cars 1
bbox:car_number ((1141, 400), (1257, 507)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13%| 170/1261 [07:31<48:18, 2.66s/it]

The times for each task are: [0.941199, 0.638778, 1.722357, 0.303719, 0.449392] with:

Minimum: 0.303719 Maximum: 1.722357 Average: 0.8111 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 7.211102550927978
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14%| 171/1261 [07:34<48:14, 2.66s/it]

The times for each task are: [1.051496, 0.650565, 1.507409, 0.470783, 0.287206] with:

Minimum: 0.287206 Maximum: 1.507409 Average: 0.7935 seconds

```
Number of cars 2
bbox:car_number ((1156, 400), (1253, 498)) : 1
The minimum distance from car: 1 is 11.0
bbox:car_number ((1075, 438), (1122, 459)) : 2
```

```
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14% | 172/1261 [07:36<48:12, 2.66s/it]

The times for each task are: [1.401222, 1.03242, 0.693771, 0.400916, 0.342221] with:

Minimum: 0.342221 Maximum: 1.401222 Average: 0.7741 seconds

```
Number of cars 1
bbox:car_number ((1135, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 17.46424919657298
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14% | 173/1261 [07:39<48:10, 2.66s/it]

The times for each task are: [0.918842, 0.623815, 1.395065, 0.286188, 0.520432] with:

Minimum: 0.286188 Maximum: 1.395065 Average: 0.7489 seconds

```
Number of cars 2
bbox:car_number ((1129, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1113, 438), (1124, 497)) : 2
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14% | 174/1261 [07:42<48:07, 2.66s/it]

The times for each task are: [1.425221, 0.589097, 0.884345, 0.30683, 0.489166] with:

Minimum: 0.30683 Maximum: 1.425221 Average: 0.7389 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1242, 497)) : 1
The minimum distance from car: 1 is 11.313708498984761
```

```
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

```
14%| 175/1261 [07:44<48:03, 2.66s/it]
```

The times for each task are: [0.629905, 0.871081, 1.610003, 0.311274, 0.509045] with:

Minimum: 0.311274 Maximum: 1.610003 Average: 0.7863 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1238, 497)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

```
14%| 176/1261 [07:47<48:01, 2.66s/it]
```

The times for each task are: [0.872458, 1.414872, 0.694653, 0.421993, 0.356532] with:

Minimum: 0.356532 Maximum: 1.414872 Average: 0.7521 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1238, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

```
14%| 177/1261 [07:50<48:00, 2.66s/it]
```

The times for each task are: [0.835179, 1.347618, 0.730512, 0.400455, 0.369248] with:

Minimum: 0.369248 Maximum: 1.347618 Average: 0.7366 seconds

```
Number of cars 1
bbox:car_number ((1120, 400), (1230, 498)) : 1
The minimum distance from car: 1 is 16.0
totalCars: 1
```

```
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]  
Length of task list: 5  
Number of processes used: 3
```

```
14% | 178/1261 [07:52<47:57, 2.66s/it]
```

The times for each task are: [1.358163, 0.703427, 1.075526, 0.344536, 0.423234] with:

Minimum: 0.344536 Maximum: 1.358163 Average: 0.781 seconds

```
Number of cars 1  
bbox:car_number ((1120, 400), (1230, 507)) : 1  
The minimum distance from car: 1 is 4.0  
totalCars: 1  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]  
Length of task list: 5  
Number of processes used: 3
```

```
14% | 179/1261 [07:55<47:55, 2.66s/it]
```

The times for each task are: [0.643614, 1.004771, 1.692471, 0.543299, 0.440472] with:

Minimum: 0.440472 Maximum: 1.692471 Average: 0.8649 seconds

```
Number of cars 1  
bbox:car_number ((1113, 400), (1227, 512)) : 1  
The minimum distance from car: 1 is 5.830951894845301  
totalCars: 1  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]  
Length of task list: 5  
Number of processes used: 3
```

```
14% | 180/1261 [07:58<47:54, 2.66s/it]
```

The times for each task are: [0.982164, 0.676259, 1.706172, 0.412788, 0.319029] with:

Minimum: 0.319029 Maximum: 1.706172 Average: 0.8193 seconds

```
Number of cars 1  
bbox:car_number ((1120, 400), (1230, 512)) : 1  
The minimum distance from car: 1 is 5.0  
totalCars: 1  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
14%| 181/1261 [08:01<47:50, 2.66s/it]
```

```
The times for each task are: [0.9072, 0.619054, 1.427152, 0.431415, 0.39342] with:
```

```
Minimum: 0.39342 Maximum: 1.427152 Average: 0.7556 seconds
```

```
Number of cars 1
bbox:car_number ((1113, 400), (1230, 516)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
14%| 182/1261 [08:03<47:48, 2.66s/it]
```

```
The times for each task are: [0.699288, 0.92688, 1.392263, 0.317539, 0.401271] with:
```

```
Minimum: 0.317539 Maximum: 1.392263 Average: 0.7474 seconds
```

```
Number of cars 1
bbox:car_number ((1113, 400), (1219, 512)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
15%| 183/1261 [08:06<47:47, 2.66s/it]
```

```
The times for each task are: [0.846579, 0.732575, 0.378966, 1.317424, 0.401625] with:
```

```
Minimum: 0.378966 Maximum: 1.317424 Average: 0.7354 seconds
```

```
Number of cars 1
bbox:car_number ((1113, 400), (1219, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
```

Number of processes used: 3

15% | 184/1261 [08:09<47:44, 2.66s/it]

The times for each task are: [0.57539, 0.895291, 1.399172, 0.309455, 0.404608] with:

Minimum: 0.309455 Maximum: 1.399172 Average: 0.7168 seconds

Number of cars 1

bbox:car_number ((1129, 400), (1214, 498)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

15% | 185/1261 [08:11<47:41, 2.66s/it]

The times for each task are: [0.647483, 0.906047, 1.662757, 0.402335, 0.34839] with:

Minimum: 0.34839 Maximum: 1.662757 Average: 0.7934 seconds

Number of cars 1

bbox:car_number ((1105, 400), (1214, 512)) : 1

The minimum distance from car: 1 is 13.892443989449804

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

15% | 186/1261 [08:14<47:39, 2.66s/it]

The times for each task are: [0.641583, 0.927906, 1.648901, 0.407261, 0.383563] with:

Minimum: 0.383563 Maximum: 1.648901 Average: 0.8018 seconds

Number of cars 1

bbox:car_number ((1113, 400), (1214, 507)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

15%| 187/1261 [08:17<47:38, 2.66s/it]

The times for each task are: [0.926614, 1.568347, 0.630828, 0.517512, 0.4137] with:

Minimum: 0.4137 Maximum: 1.568347 Average: 0.8114 seconds

Number of cars 1
bbox:car_number ((1103, 400), (1212, 507)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

15%| 188/1261 [08:20<47:34, 2.66s/it]

The times for each task are: [0.636607, 0.938154, 1.567869, 0.401621, 0.354245] with:

Minimum: 0.354245 Maximum: 1.567869 Average: 0.7797 seconds

Number of cars 1
bbox:car_number ((1103, 400), (1212, 516)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

15%| 189/1261 [08:22<47:32, 2.66s/it]

The times for each task are: [0.711331, 0.940404, 1.364012, 0.417146, 0.315058] with:

Minimum: 0.315058 Maximum: 1.364012 Average: 0.7496 seconds

Number of cars 1
bbox:car_number ((1102, 400), (1214, 518)) : 1
The minimum distance from car: 1 is 1.4142135623730951
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

15% | 190/1261 [08:25<47:29, 2.66s/it]

The times for each task are: [1.377636, 0.847993, 0.761095, 0.313496, 0.413763] with:

Minimum: 0.313496 Maximum: 1.377636 Average: 0.7428 seconds

Number of cars 1
bbox:car_number ((1113, 400), (1212, 512)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1225, 475))]
Length of task list: 5
Number of processes used: 3

15% | 191/1261 [08:28<47:27, 2.66s/it]

The times for each task are: [1.431881, 0.664674, 0.926589, 0.288021, 0.404949] with:

Minimum: 0.288021 Maximum: 1.431881 Average: 0.7432 seconds

Number of cars 1
bbox:car_number ((1103, 400), (1212, 516)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1225, 475))]
Length of task list: 5
Number of processes used: 3

15% | 192/1261 [08:31<47:25, 2.66s/it]

The times for each task are: [0.874663, 0.653508, 1.661262, 0.403311, 0.346812] with:

Minimum: 0.346812 Maximum: 1.661262 Average: 0.7879 seconds

Number of cars 2
bbox:car_number ((1105, 400), (1214, 518)) : 1
The minimum distance from car: 1 is 2.23606797749979
bbox:car_number ((1225, 475), (1257, 477)) : 2
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1225, 475))]
Length of task list: 5
Number of processes used: 3

15% | 193/1261 [08:33<47:21, 2.66s/it]

The times for each task are: [0.720053, 0.939618, 0.444196, 1.552137, 0.420586] with:

Minimum: 0.420586 Maximum: 1.552137 Average: 0.8153 seconds

Number of cars 1

bbox:car_number ((1102, 400), (1211, 512)) : 1

The minimum distance from car: 1 is 4.242640687119285

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1211, 498))]

Length of task list: 5

Number of processes used: 3

15% | 194/1261 [08:36<47:19, 2.66s/it]

The times for each task are: [0.679847, 1.018399, 1.443655, 0.398672, 0.358686] with:

Minimum: 0.358686 Maximum: 1.443655 Average: 0.7799 seconds

Number of cars 1

bbox:car_number ((1113, 400), (1211, 498)) : 1

The minimum distance from car: 1 is 9.219544457292887

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1211, 498))]

Length of task list: 5

Number of processes used: 3

15% | 195/1261 [08:39<47:17, 2.66s/it]

The times for each task are: [0.844914, 1.440922, 0.711508, 0.482385, 0.302075] with:

Minimum: 0.302075 Maximum: 1.440922 Average: 0.7564 seconds

Number of cars 1

bbox:car_number ((1113, 400), (1211, 512)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1211, 498))]

Length of task list: 5

Number of processes used: 3

16% | 196/1261 [08:42<47:16, 2.66s/it]

The times for each task are: [1.031825, 1.381349, 0.667077, 0.461117, 0.289829] with:

Minimum: 0.289829 Maximum: 1.381349 Average: 0.7662 seconds

Number of cars 2

bbox:car_number ((1090, 400), (1211, 512)) : 1

The minimum distance from car: 1 is 12.0

bbox:car_number ((1225, 431), (1272, 477)) : 2

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 197/1261 [08:44<47:13, 2.66s/it]

The times for each task are: [0.592698, 1.515728, 0.903237, 0.452679, 0.3032] with:

Minimum: 0.3032 Maximum: 1.515728 Average: 0.7535 seconds

Number of cars 2

bbox:car_number ((1090, 400), (1211, 512)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1225, 430), (1272, 477)) : 2

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 198/1261 [08:47<47:11, 2.66s/it]

The times for each task are: [0.913764, 0.705659, 1.579025, 0.40726, 0.505088] with:

Minimum: 0.40726 Maximum: 1.579025 Average: 0.8222 seconds

Number of cars 1

bbox:car_number ((1113, 400), (1272, 512)) : 1

The minimum distance from car: 1 is 42.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 199/1261 [08:50<47:09, 2.66s/it]

The times for each task are: [1.029954, 0.640546, 1.637995, 0.321421, 0.482073] with:

Minimum: 0.321421 Maximum: 1.637995 Average: 0.8224 seconds

Number of cars 1

bbox:car_number ((1102, 400), (1273, 498)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 200/1261 [08:52<47:06, 2.66s/it]

The times for each task are: [1.449001, 0.887631, 0.718903, 0.418483, 0.428498] with:

Minimum: 0.418483 Maximum: 1.449001 Average: 0.7805 seconds

Number of cars 1

bbox:car_number ((1090, 400), (1273, 512)) : 1

The minimum distance from car: 1 is 9.219544457292887

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 201/1261 [08:55<47:05, 2.67s/it]

The times for each task are: [0.883378, 0.625589, 0.446185, 1.70032, 0.317316] with:

Minimum: 0.317316 Maximum: 1.70032 Average: 0.7946 seconds

Number of cars 1

bbox:car_number ((1102, 400), (1272, 498)) : 1

The minimum distance from car: 1 is 9.219544457292887

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 202/1261 [08:58<47:01, 2.66s/it]

The times for each task are: [0.924916, 1.463025, 0.597731, 0.295753, 0.491151] with:

Minimum: 0.295753 Maximum: 1.463025 Average: 0.7545 seconds

Number of cars 1
bbox:car_number ((1102, 400), (1273, 512)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

16% | 203/1261 [09:00<46:58, 2.66s/it]

The times for each task are: [0.855036, 1.416529, 0.615694, 0.390231, 0.28602] with:

Minimum: 0.28602 Maximum: 1.416529 Average: 0.7127 seconds

Number of cars 1
bbox:car_number ((1102, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

16% | 204/1261 [09:03<46:55, 2.66s/it]

The times for each task are: [0.878847, 0.692037, 1.4236, 0.327556, 0.481153] with:

Minimum: 0.327556 Maximum: 1.4236 Average: 0.7606 seconds

Number of cars 1
bbox:car_number ((1102, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

16% | 205/1261 [09:06<46:53, 2.66s/it]

The times for each task are: [1.420263, 0.70979, 0.945186, 0.352945, 0.494745] with:

Minimum: 0.352945 Maximum: 1.420263 Average: 0.7846 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1272, 518)) : 1
The minimum distance from car: 1 is 8.54400374531753
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1268, 518))]
Length of task list: 5
Number of processes used: 3
```

16% | 206/1261 [09:08<46:50, 2.66s/it]

The times for each task are: [0.932672, 0.596295, 1.498764, 0.300435, 0.484825] with:

Minimum: 0.300435 Maximum: 1.498764 Average: 0.7626 seconds

```
Number of cars 1
bbox:car_number ((1084, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1268, 518))]
Length of task list: 5
Number of processes used: 3
```

16% | 207/1261 [09:11<46:47, 2.66s/it]

The times for each task are: [0.915607, 1.470855, 0.701804, 0.397385, 0.290697] with:

Minimum: 0.290697 Maximum: 1.470855 Average: 0.7553 seconds

```
Number of cars 1
bbox:car_number ((1084, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1268, 518))]
Length of task list: 5
Number of processes used: 3
```

16% | 208/1261 [09:14<46:45, 2.66s/it]

The times for each task are: [0.990753, 1.379245, 0.616246, 0.471997, 0.367235] with:

Minimum: 0.367235 Maximum: 1.379245 Average: 0.7651 seconds

```
Number of cars 1
bbox:car_number ((1084, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17% | 209/1261 [09:16<46:42, 2.66s/it]

The times for each task are: [0.895667, 0.615733, 1.591134, 0.301115, 0.405828] with:
Minimum: 0.301115 Maximum: 1.591134 Average: 0.7619 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17% | 210/1261 [09:19<46:39, 2.66s/it]

The times for each task are: [0.879332, 1.379083, 0.667938, 0.416137, 0.305421] with:
Minimum: 0.305421 Maximum: 1.379083 Average: 0.7296 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17% | 211/1261 [09:21<46:35, 2.66s/it]

The times for each task are: [0.859309, 0.692268, 1.362162, 0.386845, 0.302299] with:
Minimum: 0.302299 Maximum: 1.362162 Average: 0.7206 seconds

Number of cars 1

```
bbox:car_number ((1090, 400), (1268, 498)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 507))]
Length of task list: 5
Number of processes used: 3
```

17% | 212/1261 [09:24<46:33, 2.66s/it]

The times for each task are: [0.572598, 1.016694, 1.388674, 0.404402, 0.451549] with:
Minimum: 0.404402 Maximum: 1.388674 Average: 0.7668 seconds

```
Number of cars 1
bbox:car_number ((1090, 400), (1268, 498)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 507))]
Length of task list: 5
Number of processes used: 3
```

17% | 213/1261 [09:27<46:30, 2.66s/it]

The times for each task are: [0.982281, 0.681588, 1.407073, 0.409986, 0.329848] with:
Minimum: 0.329848 Maximum: 1.407073 Average: 0.7622 seconds

```
Number of cars 1
bbox:car_number ((1090, 400), (1257, 507)) : 1
The minimum distance from car: 1 is 7.211102550927978
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]
Length of task list: 5
Number of processes used: 3
```

17% | 214/1261 [09:29<46:27, 2.66s/it]

The times for each task are: [0.848287, 1.628166, 0.676049, 0.381022, 0.461402] with:
Minimum: 0.381022 Maximum: 1.628166 Average: 0.799 seconds

```
Number of cars 1
bbox:car_number ((1090, 400), (1257, 512)) : 1
```

The minimum distance from car: 1 is 3.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

17% | 215/1261 [09:32<46:25, 2.66s/it]

The times for each task are: [0.728459, 0.940011, 1.468096, 0.312218, 0.434494] with:

Minimum: 0.312218 Maximum: 1.468096 Average: 0.7767 seconds

Number of cars 1

bbox:car_number ((1082, 400), (1257, 498)) : 1

The minimum distance from car: 1 is 8.06225774829855

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

17% | 216/1261 [09:35<46:22, 2.66s/it]

The times for each task are: [1.085334, 1.413905, 0.666438, 0.345455, 0.510638] with:

Minimum: 0.345455 Maximum: 1.413905 Average: 0.8044 seconds

Number of cars 1

bbox:car_number ((1082, 400), (1268, 498)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

17% | 217/1261 [09:37<46:20, 2.66s/it]

The times for each task are: [0.992497, 1.423584, 0.608948, 0.380739, 0.281688] with:

Minimum: 0.281688 Maximum: 1.423584 Average: 0.7375 seconds

Number of cars 1

bbox:car_number ((1082, 400), (1257, 512)) : 1

The minimum distance from car: 1 is 9.219544457292887

```
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17%| 218/1261 [09:40<46:18, 2.66s/it]

The times for each task are: [1.449355, 0.670693, 0.993089, 0.369682, 0.495553] with:

Minimum: 0.369682 Maximum: 1.449355 Average: 0.7957 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1268, 498)) : 1
The minimum distance from car: 1 is 9.219544457292887
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17%| 219/1261 [09:43<46:16, 2.66s/it]

The times for each task are: [0.882377, 1.371187, 0.716502, 0.39992, 0.35591] with:

Minimum: 0.35591 Maximum: 1.371187 Average: 0.7452 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17%| 220/1261 [09:46<46:14, 2.66s/it]

The times for each task are: [0.883558, 1.373082, 0.721616, 0.299627, 0.39782] with:

Minimum: 0.299627 Maximum: 1.373082 Average: 0.7351 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 13.0
totalCars: 1
```

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

18% | 221/1261 [09:48<46:10, 2.66s/it]

The times for each task are: [0.857458, 1.411404, 0.625401, 0.310991, 0.407096] with:

Minimum: 0.310991 Maximum: 1.411404 Average: 0.7225 seconds

Number of cars 1
bbox:car_number ((1082, 400), (1257, 498)) : 1
The minimum distance from car: 1 is 9.899494936611665
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

18% | 222/1261 [09:51<46:08, 2.66s/it]

The times for each task are: [1.419024, 0.721005, 0.886145, 0.349098, 0.40366] with:

Minimum: 0.349098 Maximum: 1.419024 Average: 0.7558 seconds

Number of cars 1
bbox:car_number ((1082, 400), (1257, 512)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

18% | 223/1261 [09:54<46:05, 2.66s/it]

The times for each task are: [0.872461, 0.650687, 1.365632, 0.286912, 0.39029] with:

Minimum: 0.286912 Maximum: 1.365632 Average: 0.7132 seconds

Number of cars 1
bbox:car_number ((1082, 400), (1253, 512)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

```
Length of task list: 5
Number of processes used: 3
```

```
18%|           | 224/1261 [09:56<46:02,  2.66s/it]
```

```
The times for each task are: [0.947886, 0.618993, 1.605646, 0.32771, 0.476057] with:
```

```
Minimum: 0.32771 Maximum: 1.605646 Average: 0.7953 seconds
```

```
Number of cars 1
bbox:car_number ((1075, 400), (1257, 507)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 507))]
Length of task list: 5
Number of processes used: 3
```

```
18%|           | 225/1261 [09:59<46:00,  2.66s/it]
```

```
The times for each task are: [0.680546, 0.879107, 1.412339, 0.287054, 0.409047] with:
```

```
Minimum: 0.287054 Maximum: 1.412339 Average: 0.7336 seconds
```

```
Number of cars 2
bbox:car_number ((1075, 400), (1257, 512)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1051, 431), (1070, 495)) : 2
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]
Length of task list: 5
Number of processes used: 3
```

```
18%|           | 226/1261 [10:02<45:57,  2.66s/it]
```

```
The times for each task are: [0.5783, 0.904268, 1.378873, 0.333367, 0.380587] with:
```

```
Minimum: 0.333367 Maximum: 1.378873 Average: 0.7151 seconds
```

```
Number of cars 1
bbox:car_number ((1082, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
18%|           | 227/1261 [10:04<45:55,  2.67s/it]
```

```
The times for each task are: [1.434741, 0.875181, 0.637036, 0.400213, 0.300799] with:
```

```
Minimum: 0.300799 Maximum: 1.434741 Average: 0.7296 seconds
```

```
Number of cars 1
bbox:car_number ((1075, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1242, 498))]
Length of task list: 5
Number of processes used: 3
```

```
18%|           | 228/1261 [10:07<45:52,  2.66s/it]
```

```
The times for each task are: [0.588123, 0.865717, 1.623954, 0.403252, 0.298427] with:
```

```
Minimum: 0.298427 Maximum: 1.623954 Average: 0.7559 seconds
```

```
Number of cars 1
bbox:car_number ((1075, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1242, 498))]
Length of task list: 5
Number of processes used: 3
```

```
18%|           | 229/1261 [10:10<45:49,  2.66s/it]
```

```
The times for each task are: [0.63004, 1.444268, 1.022033, 0.402958, 0.317973] with:
```

```
Minimum: 0.317973 Maximum: 1.444268 Average: 0.7635 seconds
```

```
Number of cars 1
bbox:car_number ((1065, 400), (1257, 512)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]
Length of task list: 5
```

Number of processes used: 3

18%| 230/1261 [10:12<45:46, 2.66s/it]

The times for each task are: [1.001709, 0.674319, 1.393034, 0.404916, 0.293877] with:

Minimum: 0.293877 Maximum: 1.393034 Average: 0.7536 seconds

Number of cars 1

bbox:car_number ((1065, 400), (1242, 512)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 498))]

Length of task list: 5

Number of processes used: 3

18%| 231/1261 [10:15<45:43, 2.66s/it]

The times for each task are: [0.842463, 1.368362, 0.67959, 0.452621, 0.303053] with:

Minimum: 0.303053 Maximum: 1.368362 Average: 0.7292 seconds

Number of cars 1

bbox:car_number ((1065, 400), (1241, 498)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 498))]

Length of task list: 5

Number of processes used: 3

18%| 232/1261 [10:17<45:40, 2.66s/it]

The times for each task are: [0.928704, 0.666076, 1.682121, 0.402009, 0.351759] with:

Minimum: 0.351759 Maximum: 1.682121 Average: 0.8061 seconds

Number of cars 1

bbox:car_number ((1067, 400), (1242, 498)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1242, 498))]

Length of task list: 5

Number of processes used: 3

18%| 233/1261 [10:20<45:37, 2.66s/it]

The times for each task are: [1.07358, 1.36574, 0.648897, 0.519974, 0.310794] with:

Minimum: 0.310794 Maximum: 1.36574 Average: 0.7838 seconds

Number of cars 1
bbox:car_number ((1060, 400), (1253, 498)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 498))]
Length of task list: 5
Number of processes used: 3

19%| 234/1261 [10:23<45:34, 2.66s/it]

The times for each task are: [0.847149, 0.676661, 1.62176, 0.312437, 0.375822] with:

Minimum: 0.312437 Maximum: 1.62176 Average: 0.7668 seconds

Number of cars 1
bbox:car_number ((1051, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 10.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 498))]
Length of task list: 5
Number of processes used: 3

19%| 235/1261 [10:25<45:32, 2.66s/it]

The times for each task are: [1.464095, 0.615987, 0.918985, 0.510488, 0.300353] with:

Minimum: 0.300353 Maximum: 1.464095 Average: 0.762 seconds

Number of cars 1
bbox:car_number ((1060, 400), (1238, 512)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 498))]
Length of task list: 5
Number of processes used: 3

19%| 236/1261 [10:28<45:29, 2.66s/it]

The times for each task are: [1.001151, 1.405034, 0.702251, 0.402904, 0.293602] with:

Minimum: 0.293602 Maximum: 1.405034 Average: 0.761 seconds

Number of cars 1

bbox:car_number ((1067, 400), (1242, 512)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 512))]

Length of task list: 5

Number of processes used: 3

19%| 237/1261 [10:31<45:26, 2.66s/it]

The times for each task are: [0.898372, 0.652411, 1.677816, 0.300578, 0.440901] with:

Minimum: 0.300578 Maximum: 1.677816 Average: 0.794 seconds

Number of cars 1

bbox:car_number ((1067, 400), (1253, 512)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 512))]

Length of task list: 5

Number of processes used: 3

19%| 238/1261 [10:33<45:23, 2.66s/it]

The times for each task are: [0.890838, 0.621161, 1.607491, 0.31642, 0.403587] with:

Minimum: 0.31642 Maximum: 1.607491 Average: 0.7679 seconds

Number of cars 1

bbox:car_number ((1067, 400), (1257, 512)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]

Length of task list: 5

Number of processes used: 3

19%| 239/1261 [10:36<45:21, 2.66s/it]

The times for each task are: [1.407298, 1.038258, 0.627724, 0.344888, 0.457495] with:

Minimum: 0.344888 Maximum: 1.407298 Average: 0.7751 seconds

Number of cars 1

bbox:car_number ((1051, 400), (1268, 518)) : 1

The minimum distance from car: 1 is 4.242640687119285

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

19% | 240/1261 [10:38<45:17, 2.66s/it]

The times for each task are: [0.670428, 0.90696, 1.646796, 0.309415, 0.443086] with:

Minimum: 0.309415 Maximum: 1.646796 Average: 0.7953 seconds

Number of cars 1

bbox:car_number ((1051, 400), (1257, 518)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

19% | 241/1261 [10:41<45:15, 2.66s/it]

The times for each task are: [0.867699, 0.61182, 1.612296, 0.407182, 0.297493] with:

Minimum: 0.297493 Maximum: 1.612296 Average: 0.7593 seconds

Number of cars 1

bbox:car_number ((1051, 400), (1253, 518)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

19% | 242/1261 [10:44<45:13, 2.66s/it]

The times for each task are: [0.656155, 0.930099, 1.678797, 0.315707, 0.409273] with:

Minimum: 0.315707 Maximum: 1.678797 Average: 0.798 seconds

Number of cars 1
bbox:car_number ((1067, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 15.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

19% | 243/1261 [10:46<45:09, 2.66s/it]

The times for each task are: [0.853931, 0.624241, 1.386259, 0.337852, 0.395445] with:

Minimum: 0.337852 Maximum: 1.386259 Average: 0.7195 seconds

Number of cars 1
bbox:car_number ((1051, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

19% | 244/1261 [10:49<45:06, 2.66s/it]

The times for each task are: [0.996698, 1.377867, 0.697231, 0.472155, 0.280816] with:

Minimum: 0.280816 Maximum: 1.377867 Average: 0.765 seconds

Number of cars 1
bbox:car_number ((1051, 400), (1253, 518)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

19% | 245/1261 [10:52<45:05, 2.66s/it]

The times for each task are: [0.893382, 0.622254, 1.430115, 0.384983, 0.380775] with:

Minimum: 0.380775 Maximum: 1.430115 Average: 0.7423 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

20% | 246/1261 [10:54<45:02, 2.66s/it]

The times for each task are: [0.570136, 0.905165, 1.522577, 0.38952, 0.300248] with:

Minimum: 0.300248 Maximum: 1.522577 Average: 0.7375 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1257, 518)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

20% | 247/1261 [10:57<44:59, 2.66s/it]

The times for each task are: [0.906348, 0.634292, 0.308058, 1.648384, 0.47745] with:

Minimum: 0.308058 Maximum: 1.648384 Average: 0.7949 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1242, 522)) : 1
The minimum distance from car: 1 is 8.246211251235321
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

20% | 248/1261 [10:59<44:55, 2.66s/it]

The times for each task are: [0.646307, 1.436367, 0.999789, 0.295815, 0.439749] with:

Minimum: 0.295815 Maximum: 1.436367 Average: 0.7636 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1253, 522)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 522))]
Length of task list: 5
Number of processes used: 3
```

20%| 249/1261 [11:02<44:52, 2.66s/it]

The times for each task are: [0.907414, 0.644615, 1.386984, 0.39429, 0.288862] with:

Minimum: 0.288862 Maximum: 1.386984 Average: 0.7244 seconds

```
Number of cars 1
bbox:car_number ((1046, 400), (1253, 529)) : 1
The minimum distance from car: 1 is 4.242640687119285
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 529))]
Length of task list: 5
Number of processes used: 3
```

20%| 250/1261 [11:05<44:50, 2.66s/it]

The times for each task are: [0.881021, 1.613966, 0.653938, 0.461739, 0.368018] with:

Minimum: 0.368018 Maximum: 1.613966 Average: 0.7957 seconds

```
Number of cars 1
bbox:car_number ((1046, 400), (1253, 518)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 518))]
Length of task list: 5
Number of processes used: 3
```

20%| 251/1261 [11:07<44:47, 2.66s/it]

The times for each task are: [0.861963, 0.626408, 1.491034, 0.309858, 0.423553] with:

Minimum: 0.309858 Maximum: 1.491034 Average: 0.7426 seconds

Number of cars 0

```
Length of task list: 5
Number of processes used: 3
```

```
20%|           | 252/1261 [11:10<44:44,  2.66s/it]
```

```
The times for each task are: [1.445208, 0.634034, 0.89334, 0.391917, 0.359552] with:
```

```
Minimum: 0.359552 Maximum: 1.445208 Average: 0.7448 seconds
```

```
Number of cars 2
bbox:car_number ((1136, 419), (1200, 495)) : 1
The minimum distance from car: 1 is 19.1049731745428
bbox:car_number ((1113, 427), (1133, 485)) : 2
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1144, 431))]
Length of task list: 5
Number of processes used: 3
```

```
20%|           | 253/1261 [11:13<44:41,  2.66s/it]
```

```
The times for each task are: [0.901066, 0.625026, 1.407176, 0.3786, 0.28439] with:
```

```
Minimum: 0.28439 Maximum: 1.407176 Average: 0.7193 seconds
```

```
Number of cars 2
bbox:car_number ((1075, 427), (1133, 485)) : 1
bbox:car_number ((1144, 431), (1197, 478)) : 2
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485))]
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1144, 431))]
Length of task list: 5
Number of processes used: 3
```

```
20%|           | 254/1261 [11:16<44:40,  2.66s/it]
```

```
The times for each task are: [0.886601, 1.378836, 0.710704, 0.304727, 0.435153] with:
```

```
Minimum: 0.304727 Maximum: 1.378836 Average: 0.7432 seconds
```

```
Number of cars 2
bbox:car_number ((1051, 400), (1133, 495)) : 1
```

```
The minimum distance from car: 1 is 15.0
bbox:car_number ((1141, 415), (1211, 498)) : 2
The minimum distance from car: 1 is 6.324555320336759
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495))]
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

20% | 255/1261 [11:18<44:37, 2.66s/it]

The times for each task are: [0.603981, 0.84803, 1.354082, 0.311023, 0.451349] with:

Minimum: 0.311023 Maximum: 1.354082 Average: 0.7137 seconds

```
Number of cars 2
bbox:car_number ((1051, 400), (1133, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1141, 415), (1200, 478)) : 2
The minimum distance from car: 1 is 11.661903789690601
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

20% | 256/1261 [11:21<44:35, 2.66s/it]

The times for each task are: [1.010416, 0.65295, 1.391258, 0.445658, 0.380674] with:

Minimum: 0.380674 Maximum: 1.391258 Average: 0.7762 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1211, 498)) : 1
The minimum distance from car: 1 is 39.05124837953327
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

20% | 257/1261 [11:24<44:33, 2.66s/it]

The times for each task are: [0.704322, 1.399539, 1.00628, 0.420403, 0.292972] with:

Minimum: 0.292972 Maximum: 1.399539 Average: 0.7647 seconds

Number of cars 2

bbox:car_number ((1065, 400), (1212, 498)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1225, 431), (1242, 477)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

20% | 258/1261 [11:27<44:31, 2.66s/it]

The times for each task are: [0.615357, 1.008238, 1.65012, 0.329692, 0.371644] with:

Minimum: 0.329692 Maximum: 1.65012 Average: 0.795 seconds

Number of cars 1

bbox:car_number ((1065, 400), (1242, 516)) : 1

The minimum distance from car: 1 is 17.4928556845359

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

21% | 259/1261 [11:29<44:27, 2.66s/it]

The times for each task are: [0.624655, 0.888755, 0.407438, 0.303141, 1.641263] with:

Minimum: 0.303141 Maximum: 1.641263 Average: 0.7731 seconds

Number of cars 1

bbox:car_number ((1048, 400), (1242, 512)) : 1

The minimum distance from car: 1 is 8.246211251235321

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

21% | 260/1261 [11:32<44:25, 2.66s/it]

The times for each task are: [0.62646, 0.874425, 1.404741, 0.375833, 0.301866] with:

Minimum: 0.301866 Maximum: 1.404741 Average: 0.7167 seconds

Number of cars 1

bbox:car_number ((1048, 400), (1242, 512)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

21% | 261/1261 [11:34<44:21, 2.66s/it]

The times for each task are: [0.885116, 1.362828, 0.639098, 0.289751, 0.4451] with:

Minimum: 0.289751 Maximum: 1.362828 Average: 0.7244 seconds

Number of cars 1

bbox:car_number ((1048, 400), (1219, 498)) : 1

The minimum distance from car: 1 is 13.892443989449804

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

21% | 262/1261 [11:37<44:18, 2.66s/it]

The times for each task are: [0.664736, 1.368578, 1.005109, 0.418312, 0.329815] with:

Minimum: 0.329815 Maximum: 1.368578 Average: 0.7573 seconds

Number of cars 1

bbox:car_number ((1051, 400), (1268, 518)) : 1

The minimum distance from car: 1 is 17.029386365926403

totalCars: 2

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

21% | 263/1261 [11:40<44:16, 2.66s/it]

The times for each task are: [1.47638, 0.658495, 0.976565, 0.292437, 0.405211] with:

Minimum: 0.292437 Maximum: 1.47638 Average: 0.7618 seconds

Number of cars 1
bbox:car_number ((1046, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 13.038404810405298
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21%| 264/1261 [11:42<44:13, 2.66s/it]

The times for each task are: [0.585754, 0.889088, 0.404249, 1.711117, 0.298478] with:

Minimum: 0.298478 Maximum: 1.711117 Average: 0.7777 seconds

Number of cars 2
bbox:car_number ((1044, 400), (1238, 518)) : 1
The minimum distance from car: 1 is 4.242640687119285
bbox:car_number ((1021, 431), (1032, 462)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21%| 265/1261 [11:45<44:10, 2.66s/it]

The times for each task are: [0.629327, 0.899837, 0.449325, 0.326976, 1.669091] with:

Minimum: 0.326976 Maximum: 1.669091 Average: 0.7949 seconds

Number of cars 1
bbox:car_number ((1044, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21%| 266/1261 [11:47<44:08, 2.66s/it]

The times for each task are: [0.862, 1.391073, 0.691354, 0.299089, 0.508323] with:

Minimum: 0.299089 Maximum: 1.391073 Average: 0.7504 seconds

Number of cars 1
bbox:car_number ((1044, 400), (1219, 512)) : 1
The minimum distance from car: 1 is 11.40175425099138
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21% | 267/1261 [11:50<44:05, 2.66s/it]

The times for each task are: [0.589506, 0.900901, 1.498376, 0.396194, 0.291251] with:

Minimum: 0.291251 Maximum: 1.498376 Average: 0.7352 seconds

Number of cars 1
bbox:car_number ((1044, 400), (1230, 498)) : 1
The minimum distance from car: 1 is 9.219544457292887
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21% | 268/1261 [11:53<44:02, 2.66s/it]

The times for each task are: [0.635198, 0.870236, 1.396401, 0.299258, 0.514402] with:

Minimum: 0.299258 Maximum: 1.396401 Average: 0.7431 seconds

Number of cars 1
bbox:car_number ((1027, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 10.44030650891055
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21% | 269/1261 [11:55<43:59, 2.66s/it]

The times for each task are: [0.908792, 1.34229, 0.612819, 0.49147, 0.283427] with:

Minimum: 0.283427 Maximum: 1.34229 Average: 0.7278 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1242, 522)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

21% | 270/1261 [11:58<43:57, 2.66s/it]

The times for each task are: [0.601655, 1.002226, 1.596196, 0.520316, 0.320048] with:

Minimum: 0.320048 Maximum: 1.596196 Average: 0.8081 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1242, 529)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

21% | 271/1261 [12:01<43:54, 2.66s/it]

The times for each task are: [1.357374, 0.873813, 0.719202, 0.40354, 0.328038] with:

Minimum: 0.328038 Maximum: 1.357374 Average: 0.7364 seconds

```
Number of cars 1
bbox:car_number ((1020, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22% | 272/1261 [12:03<43:50, 2.66s/it]

The times for each task are: [0.929108, 0.5821, 1.423926, 0.289044, 0.395464] with:

Minimum: 0.289044 Maximum: 1.423926 Average: 0.7239 seconds

```
Number of cars 1
bbox:car_number ((1020, 400), (1242, 522)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 273/1261 [12:06<43:49, 2.66s/it]

The times for each task are: [0.878755, 0.694468, 1.376199, 0.375864, 0.389621] with:
Minimum: 0.375864 Maximum: 1.376199 Average: 0.743 seconds

```
Number of cars 1
bbox:car_number ((1044, 400), (1214, 512)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 274/1261 [12:08<43:45, 2.66s/it]

The times for each task are: [0.938977, 1.427347, 0.712886, 0.310034, 0.477347] with:
Minimum: 0.310034 Maximum: 1.427347 Average: 0.7733 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1253, 518)) : 1
The minimum distance from car: 1 is 8.54400374531753
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 275/1261 [12:11<43:42, 2.66s/it]

The times for each task are: [0.643491, 1.416673, 0.843367, 0.413097, 0.393319] with:
Minimum: 0.393319 Maximum: 1.416673 Average: 0.742 seconds

Number of cars 1

```
bbox:car_number ((1021, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22% | 276/1261 [12:14<43:40, 2.66s/it]

The times for each task are: [1.020823, 0.621553, 1.411056, 0.460557, 0.307734] with:
Minimum: 0.307734 Maximum: 1.411056 Average: 0.7643 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22% | 277/1261 [12:16<43:37, 2.66s/it]

The times for each task are: [1.3722, 0.667648, 1.018404, 0.381612, 0.479464] with:
Minimum: 0.381612 Maximum: 1.3722 Average: 0.7839 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 522)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22% | 278/1261 [12:19<43:34, 2.66s/it]

The times for each task are: [0.886654, 0.620323, 1.533645, 0.524357, 0.32628] with:
Minimum: 0.32628 Maximum: 1.533645 Average: 0.7783 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 522)) : 1
```

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

22% | 279/1261 [12:22<43:32, 2.66s/it]

The times for each task are: [1.455737, 0.873863, 0.715596, 0.327563, 0.388873] with:

Minimum: 0.327563 Maximum: 1.455737 Average: 0.7523 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 516)) : 1

The minimum distance from car: 1 is 14.317821063276353

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

22% | 280/1261 [12:24<43:28, 2.66s/it]

The times for each task are: [0.583974, 0.905389, 1.622835, 0.369949, 0.402262] with:

Minimum: 0.369949 Maximum: 1.622835 Average: 0.7769 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 516)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

22% | 281/1261 [12:27<43:26, 2.66s/it]

The times for each task are: [0.654428, 0.987363, 1.500175, 0.471972, 0.330482] with:

Minimum: 0.330482 Maximum: 1.500175 Average: 0.7889 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1230, 518)) : 1

The minimum distance from car: 1 is 8.06225774829855

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 282/1261 [12:30<43:24, 2.66s/it]

The times for each task are: [1.545614, 0.871565, 0.665059, 0.341388, 0.462678] with:

Minimum: 0.341388 Maximum: 1.545614 Average: 0.7773 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 283/1261 [12:33<43:22, 2.66s/it]

The times for each task are: [0.915297, 0.616171, 1.370421, 0.3227, 0.465647] with:

Minimum: 0.3227 Maximum: 1.370421 Average: 0.738 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 516)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

23%| 284/1261 [12:35<43:19, 2.66s/it]

The times for each task are: [0.573707, 1.406969, 1.06171, 0.375025, 0.289265] with:

Minimum: 0.289265 Maximum: 1.406969 Average: 0.7413 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1211, 518)) : 1
Length of task list: 5
Number of processes used: 3
```

23% | 285/1261 [12:38<43:16, 2.66s/it]

The times for each task are: [0.562625, 1.357056, 0.99975, 0.298564, 0.39475] with:

Minimum: 0.298564 Maximum: 1.357056 Average: 0.7225 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 15.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1211, 518)) : 1
Length of task list: 5
Number of processes used: 3
```

23% | 286/1261 [12:40<43:12, 2.66s/it]

The times for each task are: [0.855882, 1.419192, 0.339477, 0.675493, 0.498541] with:

Minimum: 0.339477 Maximum: 1.419192 Average: 0.7577 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 512)) : 1
The minimum distance from car: 1 is 15.297058540778355
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1241, 512)) : 1
Length of task list: 5
Number of processes used: 3
```

23% | 287/1261 [12:43<43:10, 2.66s/it]

The times for each task are: [1.341002, 0.970068, 0.687933, 0.386238, 0.327637] with:

Minimum: 0.327637 Maximum: 1.341002 Average: 0.7426 seconds

```
Number of cars 1
bbox:car_number ((1020, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1242, 518)) : 1
```

```
Length of task list: 5
Number of processes used: 3
```

```
23%|      | 288/1261 [12:45<43:07, 2.66s/it]
```

```
The times for each task are: [1.309805, 0.729991, 1.001514, 0.304954, 0.44592] with:
```

```
Minimum: 0.304954 Maximum: 1.309805 Average: 0.7584 seconds
```

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
23%|      | 289/1261 [12:48<43:05, 2.66s/it]
```

```
The times for each task are: [1.367814, 1.007776, 0.693297, 0.342956, 0.390913] with:
```

```
Minimum: 0.342956 Maximum: 1.367814 Average: 0.7606 seconds
```

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
23%|      | 290/1261 [12:51<43:02, 2.66s/it]
```

```
The times for each task are: [1.039203, 0.581692, 1.382282, 0.460675, 0.352933] with:
```

```
Minimum: 0.352933 Maximum: 1.382282 Average: 0.7634 seconds
```

```
Number of cars 1
bbox:car_number ((1021, 400), (1214, 518)) : 1
The minimum distance from car: 1 is 14.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

23%| 291/1261 [12:54<43:00, 2.66s/it]

The times for each task are: [0.565924, 1.005535, 1.634847, 0.369485, 0.510428] with:

Minimum: 0.369485 Maximum: 1.634847 Average: 0.8172 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 518)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

23%| 292/1261 [12:56<42:57, 2.66s/it]

The times for each task are: [0.668802, 0.953776, 1.455478, 0.417239, 0.292949] with:

Minimum: 0.292949 Maximum: 1.455478 Average: 0.7576 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 522)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

23%| 293/1261 [12:59<42:54, 2.66s/it]

The times for each task are: [0.910591, 0.620689, 1.625821, 0.349724, 0.490771] with:

Minimum: 0.349724 Maximum: 1.625821 Average: 0.7995 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 518)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

23%| 294/1261 [13:01<42:51, 2.66s/it]

The times for each task are: [1.398067, 0.982265, 0.682019, 0.313926, 0.379815] with:

Minimum: 0.313926 Maximum: 1.398067 Average: 0.7512 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1214, 529)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

23%| 295/1261 [13:04<42:49, 2.66s/it]

The times for each task are: [0.880855, 0.692982, 1.446033, 0.40535, 0.30609] with:

Minimum: 0.30609 Maximum: 1.446033 Average: 0.7463 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1211, 522)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

23%| 296/1261 [13:07<42:46, 2.66s/it]

The times for each task are: [1.394095, 0.893117, 0.705384, 0.412776, 0.316291] with:

Minimum: 0.316291 Maximum: 1.394095 Average: 0.7443 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1214, 529)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

24% | 297/1261 [13:09<42:44, 2.66s/it]

The times for each task are: [0.977763, 0.61471, 1.413938, 0.285519, 0.462955] with:

Minimum: 0.285519 Maximum: 1.413938 Average: 0.751 seconds

Number of cars 2

bbox:car_number ((1021, 400), (1214, 529)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1008, 438), (1017, 462)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 298/1261 [13:12<42:41, 2.66s/it]

The times for each task are: [0.635272, 0.97778, 1.424884, 0.30608, 0.392427] with:

Minimum: 0.30608 Maximum: 1.424884 Average: 0.7473 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 529)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 299/1261 [13:15<42:39, 2.66s/it]

The times for each task are: [1.398875, 0.618859, 1.056458, 0.388929, 0.286129] with:

Minimum: 0.286129 Maximum: 1.398875 Average: 0.7499 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1211, 522)) : 1

The minimum distance from car: 1 is 3.1622776601683795

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24%| 300/1261 [13:18<42:37, 2.66s/it]

The times for each task are: [0.978522, 1.438604, 0.606447, 0.391037, 0.346092] with:

Minimum: 0.346092 Maximum: 1.438604 Average: 0.7521 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1211, 522)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

24%| 301/1261 [13:21<42:35, 2.66s/it]

The times for each task are: [0.90926, 0.712357, 1.422518, 0.462761, 0.324105] with:

Minimum: 0.324105 Maximum: 1.422518 Average: 0.7662 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1214, 512)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

24%| 302/1261 [13:23<42:32, 2.66s/it]

The times for each task are: [0.575133, 0.889603, 1.660172, 0.517023, 0.37081] with:

Minimum: 0.37081 Maximum: 1.660172 Average: 0.8025 seconds

Number of cars 1
bbox:car_number ((1020, 400), (1211, 516)) : 1
The minimum distance from car: 1 is 2.8284271247461903
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

24%| 303/1261 [13:26<42:29, 2.66s/it]

The times for each task are: [0.910141, 0.577098, 1.426644, 0.301095, 0.38463] with:

Minimum: 0.301095 Maximum: 1.426644 Average: 0.7199 seconds

Number of cars 1

bbox:car_number ((1020, 400), (1214, 518)) : 1

The minimum distance from car: 1 is 2.23606797749979

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 304/1261 [13:29<42:27, 2.66s/it]

The times for each task are: [0.661693, 0.85405, 0.359529, 1.614822, 0.470545] with:

Minimum: 0.359529 Maximum: 1.614822 Average: 0.7921 seconds

Number of cars 1

bbox:car_number ((1015, 400), (1214, 518)) : 1

The minimum distance from car: 1 is 3.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 305/1261 [13:31<42:24, 2.66s/it]

The times for each task are: [0.980063, 0.622141, 1.515391, 0.45008, 0.299081] with:

Minimum: 0.299081 Maximum: 1.515391 Average: 0.7734 seconds

Number of cars 1

bbox:car_number ((1008, 400), (1208, 518)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 306/1261 [13:34<42:22, 2.66s/it]

The times for each task are: [0.614742, 0.883181, 1.405384, 0.401587, 0.348763] with:

Minimum: 0.348763 Maximum: 1.405384 Average: 0.7307 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1212, 516)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

24%| 307/1261 [13:37<42:20, 2.66s/it]

The times for each task are: [0.59141, 1.387461, 1.01488, 0.444979, 0.301695] with:

Minimum: 0.301695 Maximum: 1.387461 Average: 0.7481 seconds

Number of cars 1
bbox:car_number ((994, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 14.035668847618199
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

24%| 308/1261 [13:40<42:17, 2.66s/it]

The times for each task are: [1.403351, 0.615615, 0.831738, 0.493979, 0.296009] with:

Minimum: 0.296009 Maximum: 1.403351 Average: 0.7281 seconds

Number of cars 2
bbox:car_number ((1021, 400), (1208, 512)) : 1
The minimum distance from car: 1 is 12.36931687685298
bbox:car_number ((994, 430), (1017, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25%| 309/1261 [13:42<42:14, 2.66s/it]

The times for each task are: [0.860129, 0.669439, 1.372955, 0.319978, 0.471356] with:

Minimum: 0.319978 Maximum: 1.372955 Average: 0.7388 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25%| 310/1261 [13:45<42:11, 2.66s/it]

The times for each task are: [0.862908, 0.638377, 0.408185, 1.724812, 0.311552] with:

Minimum: 0.311552 Maximum: 1.724812 Average: 0.7892 seconds

Number of cars 1
bbox:car_number ((970, 400), (1212, 518)) : 1
The minimum distance from car: 1 is 25.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25%| 311/1261 [13:47<42:09, 2.66s/it]

The times for each task are: [0.826368, 0.700203, 1.404616, 0.381154, 0.345856] with:

Minimum: 0.345856 Maximum: 1.404616 Average: 0.7316 seconds

Number of cars 2
bbox:car_number ((1027, 400), (1211, 512)) : 1
The minimum distance from car: 1 is 28.160255680657446
bbox:car_number ((994, 427), (1010, 462)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25%| 312/1261 [13:50<42:07, 2.66s/it]

The times for each task are: [0.76082, 0.914268, 0.299764, 1.444934, 0.437144] with:

Minimum: 0.299764 Maximum: 1.444934 Average: 0.7714 seconds

Number of cars 2
bbox:car_number ((1021, 400), (1025, 472)) : 1
bbox:car_number ((1027, 400), (1211, 518)) : 2
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1214, 518))]
Length of task list: 5
Number of processes used: 3

25% | 313/1261 [13:53<42:04, 2.66s/it]

The times for each task are: [0.883046, 1.419209, 0.71039, 0.3951, 0.316492] with:

Minimum: 0.316492 Maximum: 1.419209 Average: 0.7448 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1214, 518)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1214, 518))]
Length of task list: 5
Number of processes used: 3

25% | 314/1261 [13:56<42:01, 2.66s/it]

The times for each task are: [0.632029, 1.013721, 1.649646, 0.339393, 0.378295] with:

Minimum: 0.339393 Maximum: 1.649646 Average: 0.8026 seconds

Number of cars 2
bbox:car_number ((1021, 400), (1211, 529)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1008, 445), (1017, 492)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1211, 529))]
Length of task list: 5
Number of processes used: 3

25% | 315/1261 [13:58<41:59, 2.66s/it]

The times for each task are: [0.911, 0.705628, 1.532026, 0.415354, 0.31484] with:

Minimum: 0.31484 Maximum: 1.532026 Average: 0.7758 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1208, 522)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25% | 316/1261 [14:01<41:56, 2.66s/it]

The times for each task are: [0.930414, 1.411229, 0.681442, 0.297183, 0.439401] with:

Minimum: 0.297183 Maximum: 1.411229 Average: 0.7519 seconds

Number of cars 1
bbox:car_number ((1027, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25% | 317/1261 [14:04<41:54, 2.66s/it]

The times for each task are: [0.989034, 1.490866, 0.690823, 0.425516, 0.368456] with:

Minimum: 0.368456 Maximum: 1.490866 Average: 0.7929 seconds

Number of cars 1
bbox:car_number ((1008, 400), (1208, 512)) : 1
The minimum distance from car: 1 is 11.40175425099138
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25% | 318/1261 [14:07<41:51, 2.66s/it]

The times for each task are: [0.932449, 1.402529, 0.732918, 0.301093, 0.44361] with:

Minimum: 0.301093 Maximum: 1.402529 Average: 0.7625 seconds

```
Number of cars 2
bbox:car_number ((1021, 400), (1208, 518)) : 1
The minimum distance from car: 1 is 6.708203932499369
bbox:car_number ((1008, 438), (1010, 478)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

25%| 319/1261 [14:10<41:50, 2.66s/it]

The times for each task are: [0.876712, 0.633937, 1.436698, 0.421303, 0.409699] with:

Minimum: 0.409699 Maximum: 1.436698 Average: 0.7557 seconds

```
Number of cars 1
bbox:car_number ((952, 400), (1187, 518)) : 1
The minimum distance from car: 1 is 45.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

25%| 320/1261 [14:13<41:48, 2.67s/it]

The times for each task are: [0.681261, 1.487635, 0.935841, 0.410378, 0.310256] with:

Minimum: 0.310256 Maximum: 1.487635 Average: 0.7651 seconds

```
Number of cars 2
bbox:car_number ((989, 400), (1187, 522)) : 1
The minimum distance from car: 1 is 19.1049731745428
bbox:car_number ((1195, 445), (1211, 492)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

25%| 321/1261 [14:15<41:45, 2.67s/it]

The times for each task are: [0.937123, 0.663892, 1.590084, 0.40104, 0.303314] with:

Minimum: 0.303314 Maximum: 1.590084 Average: 0.7791 seconds

Number of cars 2
bbox:car_number ((989, 400), (1187, 512)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1195, 445), (1211, 492)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26% | 322/1261 [14:18<41:43, 2.67s/it]

The times for each task are: [0.902946, 0.610482, 1.677909, 0.299994, 0.388268] with:

Minimum: 0.299994 Maximum: 1.677909 Average: 0.7759 seconds

Number of cars 1
bbox:car_number ((989, 400), (1197, 507)) : 1
The minimum distance from car: 1 is 5.830951894845301
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26% | 323/1261 [14:21<41:40, 2.67s/it]

The times for each task are: [0.626951, 0.93454, 1.680925, 0.33116, 0.537539] with:

Minimum: 0.33116 Maximum: 1.680925 Average: 0.8222 seconds

Number of cars 2
bbox:car_number ((994, 400), (1187, 516)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1195, 445), (1200, 492)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26% | 324/1261 [14:23<41:38, 2.67s/it]

The times for each task are: [0.878416, 0.598673, 1.603689, 0.45579, 0.28632] with:

Minimum: 0.28632 Maximum: 1.603689 Average: 0.7646 seconds

Number of cars 1
bbox:car_number ((998, 400), (1200, 512)) : 1
The minimum distance from car: 1 is 9.219544457292887
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26%| 325/1261 [14:26<41:36, 2.67s/it]

The times for each task are: [0.635557, 0.916571, 1.408805, 0.300796, 0.463459] with:

Minimum: 0.300796 Maximum: 1.408805 Average: 0.745 seconds

Number of cars 1
bbox:car_number ((998, 400), (1187, 518)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26%| 326/1261 [14:29<41:33, 2.67s/it]

The times for each task are: [0.578772, 1.419788, 1.071082, 0.39582, 0.301288] with:

Minimum: 0.301288 Maximum: 1.419788 Average: 0.7533 seconds

Number of cars 1
bbox:car_number ((998, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 12.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26%| 327/1261 [14:31<41:30, 2.67s/it]

The times for each task are: [0.969049, 0.684627, 1.373231, 0.309362, 0.40714] with:

Minimum: 0.309362 Maximum: 1.373231 Average: 0.7487 seconds

```
Number of cars 2
bbox:car_number ((998, 400), (1187, 512)) : 1
The minimum distance from car: 1 is 12.36931687685298
bbox:car_number ((1195, 445), (1211, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

26%| 328/1261 [14:34<41:28, 2.67s/it]

The times for each task are: [0.874615, 0.62, 1.421158, 0.288873, 0.547262] with:

Minimum: 0.288873 Maximum: 1.421158 Average: 0.7504 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 4.242640687119285
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

26%| 329/1261 [14:37<41:25, 2.67s/it]

The times for each task are: [0.896381, 0.607672, 1.423745, 0.414378, 0.349052] with:

Minimum: 0.349052 Maximum: 1.423745 Average: 0.7382 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1187, 512)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

26%| 330/1261 [14:40<41:23, 2.67s/it]

The times for each task are: [0.893481, 0.72211, 1.46162, 0.42065, 0.343093] with:

Minimum: 0.343093 Maximum: 1.46162 Average: 0.7682 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1180, 518)) : 1
The minimum distance from car: 1 is 4.242640687119285
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

26% | 331/1261 [14:43<41:21, 2.67s/it]

The times for each task are: [0.881808, 0.696922, 1.59958, 0.410924, 0.3487] with:

Minimum: 0.3487 Maximum: 1.59958 Average: 0.7876 seconds

```
Number of cars 1
bbox:car_number ((998, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 17.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

26% | 332/1261 [14:46<41:19, 2.67s/it]

The times for each task are: [1.435927, 1.09745, 0.630444, 0.398117, 0.372127] with:

Minimum: 0.372127 Maximum: 1.435927 Average: 0.7868 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1187, 518)) : 1
The minimum distance from car: 1 is 14.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

26% | 333/1261 [14:48<41:16, 2.67s/it]

The times for each task are: [0.813861, 0.663787, 1.448276, 0.291513, 0.395271] with:

Minimum: 0.291513 Maximum: 1.448276 Average: 0.7225 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1181, 518)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

26%| 334/1261 [14:51<41:13, 2.67s/it]

The times for each task are: [0.597172, 0.884266, 1.33783, 0.292657, 0.487407] with:

Minimum: 0.292657 Maximum: 1.33783 Average: 0.7199 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1182, 512)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27%| 335/1261 [14:53<41:10, 2.67s/it]

The times for each task are: [0.84518, 0.648679, 1.405176, 0.489793, 0.298653] with:

Minimum: 0.298653 Maximum: 1.405176 Average: 0.7375 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27%| 336/1261 [14:56<41:08, 2.67s/it]

The times for each task are: [0.596622, 0.917373, 1.372087, 0.383459, 0.290773] with:

Minimum: 0.290773 Maximum: 1.372087 Average: 0.7121 seconds

Number of cars 1

```
bbox:car_number ((994, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1182, 518))]
Length of task list: 5
Number of processes used: 3
```

27% | 337/1261 [14:58<41:04, 2.67s/it]

The times for each task are: [0.910797, 0.689825, 1.374051, 0.396276, 0.291071] with:
Minimum: 0.291071 Maximum: 1.374051 Average: 0.7324 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1182, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1182, 518))]
Length of task list: 5
Number of processes used: 3
```

27% | 338/1261 [15:01<41:02, 2.67s/it]

The times for each task are: [0.624925, 0.843166, 1.362023, 0.319711, 0.459561] with:
Minimum: 0.319711 Maximum: 1.362023 Average: 0.7219 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1212, 518))]
Length of task list: 5
Number of processes used: 3
```

27% | 339/1261 [15:04<40:59, 2.67s/it]

The times for each task are: [0.830248, 1.323387, 0.700935, 0.408051, 0.292059] with:
Minimum: 0.292059 Maximum: 1.323387 Average: 0.7109 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1212, 518)) : 1
```

The minimum distance from car: 1 is 14.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

27% | 340/1261 [15:07<40:57, 2.67s/it]

The times for each task are: [1.376665, 0.867324, 0.750548, 0.292403, 0.450157] with:

Minimum: 0.292403 Maximum: 1.376665 Average: 0.7474 seconds

Number of cars 1

bbox:car_number ((994, 400), (1180, 518)) : 1

The minimum distance from car: 1 is 16.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

27% | 341/1261 [15:09<40:53, 2.67s/it]

The times for each task are: [0.976813, 0.690318, 1.325616, 0.364749, 0.446348] with:

Minimum: 0.364749 Maximum: 1.325616 Average: 0.7608 seconds

Number of cars 2

bbox:car_number ((994, 400), (1162, 512)) : 1

The minimum distance from car: 1 is 9.486832980505138

bbox:car_number ((1165, 431), (1180, 477)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

27% | 342/1261 [15:11<40:50, 2.67s/it]

The times for each task are: [0.643732, 1.021965, 1.33881, 0.309341, 0.426139] with:

Minimum: 0.309341 Maximum: 1.33881 Average: 0.748 seconds

Number of cars 3

bbox:car_number ((994, 400), (1162, 518)) : 1

```
The minimum distance from car: 1 is 3.0
bbox:car_number ((1165, 431), (1180, 478)) : 2
bbox:car_number ((1183, 445), (1212, 477)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27% | 343/1261 [15:14<40:47, 2.67s/it]

The times for each task are: [0.84399, 0.636688, 1.564076, 0.527583, 0.370131] with:

Minimum: 0.370131 Maximum: 1.564076 Average: 0.7885 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1230, 518)) : 1
The minimum distance from car: 1 is 34.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27% | 344/1261 [15:16<40:44, 2.67s/it]

The times for each task are: [0.907178, 0.646635, 1.689275, 0.37148, 0.436058] with:

Minimum: 0.37148 Maximum: 1.689275 Average: 0.8101 seconds

```
Number of cars 2
bbox:car_number ((994, 400), (1182, 512)) : 1
The minimum distance from car: 1 is 24.186773244895647
bbox:car_number ((1204, 419), (1219, 459)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27% | 345/1261 [15:19<40:41, 2.67s/it]

The times for each task are: [0.912998, 0.6208, 1.411593, 0.297733, 0.497586] with:

Minimum: 0.297733 Maximum: 1.411593 Average: 0.7481 seconds

```
Number of cars 2
bbox:car_number ((994, 400), (1160, 512)) : 1
The minimum distance from car: 1 is 11.0
bbox:car_number ((1165, 431), (1180, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1165, 431))]
Length of task list: 5
Number of processes used: 3
```

27% | 346/1261 [15:21<40:37, 2.66s/it]

The times for each task are: [0.907212, 0.693935, 0.397915, 1.507324, 0.336166] with:

Minimum: 0.336166 Maximum: 1.507324 Average: 0.7685 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1180, 512)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1165, 431))]
Length of task list: 5
Number of processes used: 3
```

28% | 347/1261 [15:24<40:35, 2.67s/it]

The times for each task are: [0.989226, 0.69128, 1.553872, 0.454114, 0.340765] with:

Minimum: 0.340765 Maximum: 1.553872 Average: 0.8059 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1167, 507)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1165, 431))]
Length of task list: 5
Number of processes used: 3
```

28% | 348/1261 [15:27<40:33, 2.67s/it]

The times for each task are: [0.927516, 1.410316, 0.612067, 0.394248, 0.288859] with:

Minimum: 0.288859 Maximum: 1.410316 Average: 0.7266 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1167, 507)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28%| 349/1261 [15:30<40:30, 2.67s/it]

The times for each task are: [0.922646, 0.640326, 1.686355, 0.41517, 0.281801] with:

Minimum: 0.281801 Maximum: 1.686355 Average: 0.7893 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1162, 507)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28%| 350/1261 [15:32<40:28, 2.67s/it]

The times for each task are: [1.330269, 0.954579, 0.673562, 0.303546, 0.389467] with:

Minimum: 0.303546 Maximum: 1.330269 Average: 0.7303 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1160, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28%| 351/1261 [15:35<40:25, 2.66s/it]

The times for each task are: [0.903206, 0.605875, 1.694715, 0.420081, 0.295879] with:

Minimum: 0.295879 Maximum: 1.694715 Average: 0.784 seconds

Number of cars 1

```
bbox:car_number ((989, 400), (1162, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28% | 352/1261 [15:37<40:22, 2.66s/it]

The times for each task are: [0.643073, 0.958116, 0.31311, 1.662036, 0.388075] with:

Minimum: 0.31311 Maximum: 1.662036 Average: 0.7929 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1160, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28% | 353/1261 [15:40<40:19, 2.66s/it]

The times for each task are: [0.87662, 1.352805, 0.690583, 0.405942, 0.28606] with:

Minimum: 0.28606 Maximum: 1.352805 Average: 0.7224 seconds

```
Number of cars 3
bbox:car_number ((989, 400), (1133, 498)) : 1
The minimum distance from car: 1 is 13.0
bbox:car_number ((1158, 400), (1230, 485)) : 2
The minimum distance from car: 1 is 38.91015291668744
bbox:car_number ((1144, 427), (1149, 485)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

28% | 354/1261 [15:43<40:17, 2.67s/it]

The times for each task are: [0.634453, 0.896711, 0.400568, 1.374341, 0.289474] with:

Minimum: 0.289474 Maximum: 1.374341 Average: 0.7191 seconds

Number of cars 2
bbox:car_number ((989, 400), (1149, 507)) : 1
The minimum distance from car: 1 is 8.94427190999916
bbox:car_number ((985, 430), (987, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

28% | 355/1261 [15:46<40:14, 2.67s/it]

The times for each task are: [0.631619, 0.866087, 1.420297, 0.301944, 0.442136] with:

Minimum: 0.301944 Maximum: 1.420297 Average: 0.7324 seconds

Number of cars 3
bbox:car_number ((989, 400), (1143, 498)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1195, 415), (1242, 462)) : 2
bbox:car_number ((1150, 445), (1160, 477)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

28% | 356/1261 [15:48<40:11, 2.66s/it]

The times for each task are: [0.969464, 1.459688, 0.47376, 0.75913, 0.36817] with:

Minimum: 0.36817 Maximum: 1.459688 Average: 0.806 seconds

Number of cars 1
bbox:car_number ((989, 400), (1152, 507)) : 1
The minimum distance from car: 1 is 5.656854249492381
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

28% | 357/1261 [15:51<40:09, 2.67s/it]

The times for each task are: [0.929318, 0.640659, 1.503396, 0.324178, 0.420269] with:

Minimum: 0.324178 Maximum: 1.503396 Average: 0.7636 seconds

Number of cars 1

bbox:car_number ((989, 400), (1139, 507)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

28% | 358/1261 [15:53<40:06, 2.66s/it]

The times for each task are: [1.406758, 0.61709, 1.09673, 0.401217, 0.373956] with:

Minimum: 0.373956 Maximum: 1.406758 Average: 0.7792 seconds

Number of cars 1

bbox:car_number ((985, 415), (1143, 507)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

28% | 359/1261 [15:56<40:03, 2.66s/it]

The times for each task are: [0.571889, 1.003573, 1.63321, 0.314696, 0.41981] with:

Minimum: 0.314696 Maximum: 1.63321 Average: 0.7886 seconds

Number of cars 1

bbox:car_number ((975, 400), (1143, 507)) : 1

The minimum distance from car: 1 is 9.433981132056603

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

29% | 360/1261 [15:59<40:00, 2.66s/it]

The times for each task are: [0.873999, 0.635023, 1.451236, 0.312627, 0.503237] with:

Minimum: 0.312627 Maximum: 1.451236 Average: 0.7552 seconds

Number of cars 1
bbox:car_number ((985, 400), (1200, 512)) : 1
The minimum distance from car: 1 is 33.13608305156178
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

29% | 361/1261 [16:01<39:57, 2.66s/it]

The times for each task are: [0.869478, 1.460769, 0.676866, 0.406295, 0.316914] with:

Minimum: 0.316914 Maximum: 1.460769 Average: 0.7461 seconds

Number of cars 1
bbox:car_number ((975, 400), (1160, 512)) : 1
The minimum distance from car: 1 is 25.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

29% | 362/1261 [16:04<39:55, 2.66s/it]

The times for each task are: [1.424158, 0.990295, 0.575096, 0.292122, 0.388059] with:

Minimum: 0.292122 Maximum: 1.424158 Average: 0.7339 seconds

Number of cars 1
bbox:car_number ((970, 400), (1143, 512)) : 1
The minimum distance from car: 1 is 11.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

29% | 363/1261 [16:07<39:52, 2.66s/it]

The times for each task are: [0.690516, 0.919883, 1.65534, 0.415516, 0.352489] with:

Minimum: 0.352489 Maximum: 1.65534 Average: 0.8067 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 512)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 364/1261 [16:09<39:49, 2.66s/it]

The times for each task are: [0.980024, 0.669458, 1.586756, 0.443297, 0.300866] with:

Minimum: 0.300866 Maximum: 1.586756 Average: 0.7961 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 365/1261 [16:12<39:46, 2.66s/it]

The times for each task are: [0.877783, 0.614136, 1.441478, 0.298593, 0.474882] with:

Minimum: 0.298593 Maximum: 1.441478 Average: 0.7414 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 366/1261 [16:14<39:43, 2.66s/it]

The times for each task are: [0.612328, 0.893073, 1.424698, 0.307834, 0.389503] with:

Minimum: 0.307834 Maximum: 1.424698 Average: 0.7255 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 518)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29%| 367/1261 [16:17<39:41, 2.66s/it]

The times for each task are: [0.602672, 0.976496, 1.389155, 0.438412, 0.426205] with:
Minimum: 0.426205 Maximum: 1.389155 Average: 0.7666 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 18.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29%| 368/1261 [16:20<39:39, 2.66s/it]

The times for each task are: [0.658819, 0.992681, 0.417091, 1.705943, 0.301723] with:
Minimum: 0.301723 Maximum: 1.705943 Average: 0.8153 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 512)) : 1
The minimum distance from car: 1 is 18.24828759089466
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29%| 369/1261 [16:23<39:37, 2.67s/it]

The times for each task are: [0.855454, 0.614557, 0.419768, 1.470929, 0.32974] with:
Minimum: 0.32974 Maximum: 1.470929 Average: 0.7381 seconds

Number of cars 1

```
bbox:car_number ((975, 400), (1139, 516)) : 1
The minimum distance from car: 1 is 2.8284271247461903
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 370/1261 [16:26<39:34, 2.67s/it]

The times for each task are: [0.883061, 0.616821, 1.611004, 0.295743, 0.420854] with:
Minimum: 0.295743 Maximum: 1.611004 Average: 0.7655 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1137, 516)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 371/1261 [16:28<39:31, 2.67s/it]

The times for each task are: [0.622845, 0.960055, 1.368921, 0.325066, 0.403029] with:
Minimum: 0.325066 Maximum: 1.368921 Average: 0.736 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1139, 512)) : 1
The minimum distance from car: 1 is 2.23606797749979
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

30% | 372/1261 [16:31<39:29, 2.67s/it]

The times for each task are: [0.843518, 1.41099, 0.661273, 0.48417, 0.345787] with:
Minimum: 0.345787 Maximum: 1.41099 Average: 0.7491 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1124, 507)) : 1
```

```
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1124, 498))]
Length of task list: 5
Number of processes used: 3
```

30% | 373/1261 [16:33<39:26, 2.66s/it]

The times for each task are: [0.833539, 0.709121, 1.474405, 0.485798, 0.303043] with:

Minimum: 0.303043 Maximum: 1.474405 Average: 0.7612 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1124, 498)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1124, 498))]
Length of task list: 5
Number of processes used: 3
```

30% | 374/1261 [16:36<39:23, 2.66s/it]

The times for each task are: [0.838951, 0.605858, 1.662117, 0.309475, 0.385948] with:

Minimum: 0.309475 Maximum: 1.662117 Average: 0.7605 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1122, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1122, 498))]
Length of task list: 5
Number of processes used: 3
```

30% | 375/1261 [16:39<39:20, 2.66s/it]

The times for each task are: [0.911707, 1.398757, 0.620009, 0.296422, 0.397301] with:

Minimum: 0.296422 Maximum: 1.398757 Average: 0.7248 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1124, 507)) : 1
The minimum distance from car: 1 is 4.123105625617661
```

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1118, 507))]
Length of task list: 5
Number of processes used: 3
```

30%| 376/1261 [16:42<39:18, 2.67s/it]

The times for each task are: [1.014355, 0.605636, 1.37313, 0.282638, 0.385877] with:

Minimum: 0.282638 Maximum: 1.37313 Average: 0.7323 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1124, 498)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1118, 507))]
Length of task list: 5
Number of processes used: 3
```

30%| 377/1261 [16:44<39:16, 2.67s/it]

The times for each task are: [0.617909, 0.892823, 1.628132, 0.46458, 0.366182] with:

Minimum: 0.366182 Maximum: 1.628132 Average: 0.7939 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1122, 512)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1118, 507))]
Length of task list: 5
Number of processes used: 3
```

30%| 378/1261 [16:47<39:13, 2.67s/it]

The times for each task are: [1.43276, 0.643426, 0.993023, 0.294803, 0.435076] with:

Minimum: 0.294803 Maximum: 1.43276 Average: 0.7598 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1118, 507)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1118, 512)) : 1
Length of task list: 5
Number of processes used: 3
```

30% | 379/1261 [16:50<39:10, 2.66s/it]

The times for each task are: [0.946308, 1.480167, 0.711736, 0.46357, 0.397182] with:

Minimum: 0.397182 Maximum: 1.480167 Average: 0.7998 seconds

```
Number of cars 1
bbox:car_number ((958, 400), (1118, 512)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1116, 498)) : 1
Length of task list: 5
Number of processes used: 3
```

30% | 380/1261 [16:52<39:07, 2.66s/it]

The times for each task are: [1.434195, 0.861367, 0.697917, 0.299439, 0.442153] with:

Minimum: 0.299439 Maximum: 1.434195 Average: 0.747 seconds

```
Number of cars 1
bbox:car_number ((958, 400), (1116, 498)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1116, 498)) : 1
Length of task list: 5
Number of processes used: 3
```

30% | 381/1261 [16:55<39:04, 2.66s/it]

The times for each task are: [0.896574, 0.634144, 1.688538, 0.396337, 0.388402] with:

Minimum: 0.388402 Maximum: 1.688538 Average: 0.8008 seconds

```
Number of cars 1
bbox:car_number ((955, 400), (1116, 507)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1116, 507)) : 1
```

```
Length of task list: 5
Number of processes used: 3
```

```
30%|      | 382/1261 [16:57<39:02,  2.66s/it]
```

```
The times for each task are: [0.621639, 0.903385, 1.661388, 0.420169, 0.349827] with:
```

```
Minimum: 0.349827 Maximum: 1.661388 Average: 0.7913 seconds
```

```
Number of cars 1
bbox:car_number ((955, 400), (1107, 498)) : 1
The minimum distance from car: 1 is 5.656854249492381
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
30%|      | 383/1261 [17:00<38:59,  2.66s/it]
```

```
The times for each task are: [1.000851, 0.676524, 1.362704, 0.372344, 0.430469] with:
```

```
Minimum: 0.372344 Maximum: 1.362704 Average: 0.7686 seconds
```

```
Number of cars 1
bbox:car_number ((955, 400), (1107, 507)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
30%|      | 384/1261 [17:02<38:56,  2.66s/it]
```

```
The times for each task are: [0.833563, 1.382617, 0.645334, 0.41504, 0.289928] with:
```

```
Minimum: 0.289928 Maximum: 1.382617 Average: 0.7133 seconds
```

```
Number of cars 1
bbox:car_number ((955, 400), (1107, 498)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

31%| 385/1261 [17:05<38:53, 2.66s/it]

The times for each task are: [0.834693, 1.403581, 0.511639, 0.612121, 0.296796] with:

Minimum: 0.296796 Maximum: 1.403581 Average: 0.7318 seconds

Number of cars 1

bbox:car_number ((952, 400), (1107, 498)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

31%| 386/1261 [17:08<38:51, 2.66s/it]

The times for each task are: [1.358339, 0.910707, 0.738045, 0.301165, 0.397349] with:

Minimum: 0.301165 Maximum: 1.358339 Average: 0.7411 seconds

Number of cars 1

bbox:car_number ((952, 400), (1106, 507)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

31%| 387/1261 [17:11<38:48, 2.66s/it]

The times for each task are: [0.604846, 0.914283, 1.44691, 0.391487, 0.297111] with:

Minimum: 0.297111 Maximum: 1.44691 Average: 0.7309 seconds

Number of cars 1

bbox:car_number ((951, 400), (1106, 497)) : 1

The minimum distance from car: 1 is 5.0990195135927845

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

```
31%|      | 388/1261 [17:14<38:46,  2.67s/it]
```

The times for each task are: [1.035408, 1.390761, 0.641601, 0.472848, 0.375035] with:

Minimum: 0.375035 Maximum: 1.390761 Average: 0.7831 seconds

Number of cars 1

bbox:car_number ((955, 400), (1107, 498)) : 1

The minimum distance from car: 1 is 3.1622776601683795

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

```
31%|      | 389/1261 [17:17<38:44,  2.67s/it]
```

The times for each task are: [0.827103, 0.693149, 1.635277, 0.39224, 0.344969] with:

Minimum: 0.344969 Maximum: 1.635277 Average: 0.7785 seconds

Number of cars 1

bbox:car_number ((955, 400), (1106, 498)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

```
31%|      | 390/1261 [17:19<38:42,  2.67s/it]
```

The times for each task are: [0.890299, 1.435577, 0.727685, 0.468041, 0.321602] with:

Minimum: 0.321602 Maximum: 1.435577 Average: 0.7686 seconds

Number of cars 1

bbox:car_number ((958, 400), (1105, 498)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

31% | 391/1261 [17:22<38:40, 2.67s/it]

The times for each task are: [0.907752, 1.453279, 0.706056, 0.314992, 0.406845] with:

Minimum: 0.314992 Maximum: 1.453279 Average: 0.7578 seconds

Number of cars 1

bbox:car_number ((958, 400), (1106, 497)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 392/1261 [17:25<38:37, 2.67s/it]

The times for each task are: [0.63478, 1.102683, 1.528013, 0.368865, 0.445008] with:

Minimum: 0.368865 Maximum: 1.528013 Average: 0.8159 seconds

Number of cars 1

bbox:car_number ((951, 400), (1079, 495)) : 1

The minimum distance from car: 1 is 17.029386365926403

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 393/1261 [17:28<38:35, 2.67s/it]

The times for each task are: [0.663017, 0.947481, 1.514753, 0.481477, 0.303128] with:

Minimum: 0.303128 Maximum: 1.514753 Average: 0.782 seconds

Number of cars 1

bbox:car_number ((940, 400), (1092, 498)) : 1

The minimum distance from car: 1 is 2.23606797749979

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 394/1261 [17:31<38:33, 2.67s/it]

The times for each task are: [0.673677, 0.880851, 1.624307, 0.404446, 0.342355] with:

Minimum: 0.342355 Maximum: 1.624307 Average: 0.7851 seconds

Number of cars 1

bbox:car_number ((940, 400), (1092, 497)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 395/1261 [17:34<38:30, 2.67s/it]

The times for each task are: [0.672399, 1.497293, 1.075381, 0.355147, 0.417388] with:

Minimum: 0.355147 Maximum: 1.497293 Average: 0.8035 seconds

Number of cars 1

bbox:car_number ((932, 400), (1087, 495)) : 1

The minimum distance from car: 1 is 7.0710678118654755

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 396/1261 [17:36<38:28, 2.67s/it]

The times for each task are: [0.627749, 1.012933, 1.489232, 0.43622, 0.395922] with:

Minimum: 0.395922 Maximum: 1.489232 Average: 0.7924 seconds

Number of cars 1

bbox:car_number ((932, 400), (1092, 497)) : 1

The minimum distance from car: 1 is 3.1622776601683795

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 397/1261 [17:39<38:25, 2.67s/it]

The times for each task are: [1.378026, 0.885748, 0.776537, 0.296532, 0.481881] with:

Minimum: 0.296532 Maximum: 1.378026 Average: 0.7637 seconds

Number of cars 1
bbox:car_number ((932, 400), (1087, 495)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

32% | 398/1261 [17:42<38:23, 2.67s/it]

The times for each task are: [0.943805, 0.663884, 1.400066, 0.283197, 0.387329] with:

Minimum: 0.283197 Maximum: 1.400066 Average: 0.7357 seconds

Number of cars 1
bbox:car_number ((932, 400), (1086, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

32% | 399/1261 [17:44<38:20, 2.67s/it]

The times for each task are: [0.906729, 0.616237, 1.62962, 0.293932, 0.40746] with:

Minimum: 0.293932 Maximum: 1.62962 Average: 0.7708 seconds

Number of cars 1
bbox:car_number ((940, 400), (1086, 497)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

32% | 400/1261 [17:47<38:18, 2.67s/it]

The times for each task are: [0.92943, 0.658967, 1.48244, 0.39659, 0.351361] with:

Minimum: 0.351361 Maximum: 1.48244 Average: 0.7638 seconds

```
Number of cars 1
bbox:car_number ((932, 400), (1079, 497)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 401/1261 [17:50<38:14, 2.67s/it]

The times for each task are: [1.357375, 0.697026, 0.873706, 0.353653, 0.431772] with:

Minimum: 0.353653 Maximum: 1.357375 Average: 0.7427 seconds

```
Number of cars 1
bbox:car_number ((932, 400), (1077, 497)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 402/1261 [17:52<38:12, 2.67s/it]

The times for each task are: [0.617791, 0.400885, 0.90695, 0.314756, 1.629953] with:

Minimum: 0.314756 Maximum: 1.629953 Average: 0.7741 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1079, 495)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 403/1261 [17:55<38:10, 2.67s/it]

The times for each task are: [0.902805, 0.605338, 1.798186, 0.432524, 0.300695] with:

Minimum: 0.300695 Maximum: 1.798186 Average: 0.8079 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1086, 498)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32%| 404/1261 [17:58<38:08, 2.67s/it]

The times for each task are: [0.972771, 0.67043, 0.434211, 1.722684, 0.371541] with:
Minimum: 0.371541 Maximum: 1.722684 Average: 0.8343 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1077, 495)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32%| 405/1261 [18:01<38:05, 2.67s/it]

The times for each task are: [0.638055, 1.496468, 0.92289, 0.39964, 0.324123] with:
Minimum: 0.324123 Maximum: 1.496468 Average: 0.7562 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1079, 498)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32%| 406/1261 [18:04<38:03, 2.67s/it]

The times for each task are: [0.647087, 1.424585, 1.043366, 0.385401, 0.487467] with:
Minimum: 0.385401 Maximum: 1.424585 Average: 0.7976 seconds

Number of cars 1

```
bbox:car_number ((958, 400), (1077, 498)) : 1
The minimum distance from car: 1 is 13.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 407/1261 [18:06<38:00, 2.67s/it]

The times for each task are: [0.621934, 1.430262, 0.856083, 0.303149, 0.38925] with:

Minimum: 0.303149 Maximum: 1.430262 Average: 0.7201 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1079, 498)) : 1
The minimum distance from car: 1 is 13.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 408/1261 [18:09<37:57, 2.67s/it]

The times for each task are: [0.87938, 1.478618, 0.615221, 0.406935, 0.287697] with:

Minimum: 0.287697 Maximum: 1.478618 Average: 0.7336 seconds

```
Number of cars 1
bbox:car_number ((932, 400), (1077, 498)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 409/1261 [18:11<37:54, 2.67s/it]

The times for each task are: [0.87825, 0.626935, 1.478672, 0.29149, 0.393584] with:

Minimum: 0.29149 Maximum: 1.478672 Average: 0.7338 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1070, 497)) : 1
```

```
The minimum distance from car: 1 is 1.4142135623730951
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 410/1261 [18:13<37:50, 2.67s/it]

The times for each task are: [0.643861, 0.912274, 1.393441, 0.395717, 0.295281] with:

Minimum: 0.295281 Maximum: 1.393441 Average: 0.7281 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1070, 495)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 411/1261 [18:16<37:47, 2.67s/it]

The times for each task are: [0.91688, 1.360167, 0.738866, 0.410119, 0.329161] with:

Minimum: 0.329161 Maximum: 1.360167 Average: 0.751 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1070, 497)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 412/1261 [18:19<37:44, 2.67s/it]

The times for each task are: [0.580344, 1.433939, 0.8552, 0.293599, 0.377717] with:

Minimum: 0.293599 Maximum: 1.433939 Average: 0.7082 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1070, 497)) : 1
The minimum distance from car: 1 is 7.0
```

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 413/1261 [18:21<37:42, 2.67s/it]

The times for each task are: [0.575736, 1.055597, 1.41981, 0.460068, 0.36337] with:

Minimum: 0.36337 Maximum: 1.41981 Average: 0.7749 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1070, 497)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 414/1261 [18:24<37:39, 2.67s/it]

The times for each task are: [0.960802, 0.659435, 1.508866, 0.453738, 0.356854] with:

Minimum: 0.356854 Maximum: 1.508866 Average: 0.7879 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1062, 495)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 415/1261 [18:27<37:37, 2.67s/it]

The times for each task are: [1.364475, 0.582434, 1.001712, 0.392772, 0.307252] with:

Minimum: 0.307252 Maximum: 1.364475 Average: 0.7297 seconds

```
Number of cars 1
bbox:car_number ((952, 400), (1062, 495)) : 1
The minimum distance from car: 1 is 14.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1092, 495)) : 2
Length of task list: 5
Number of processes used: 3
```

33% | 416/1261 [18:29<37:34, 2.67s/it]

The times for each task are: [0.904091, 0.664687, 1.510244, 0.498662, 0.305567] with:

Minimum: 0.305567 Maximum: 1.510244 Average: 0.7767 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1092, 495)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1092, 495)) : 2
Length of task list: 5
Number of processes used: 3
```

33% | 417/1261 [18:32<37:31, 2.67s/it]

The times for each task are: [0.681923, 0.895396, 1.529371, 0.455377, 0.355408] with:

Minimum: 0.355408 Maximum: 1.529371 Average: 0.7835 seconds

```
Number of cars 2
bbox:car_number ((925, 400), (1062, 495)) : 1
The minimum distance from car: 1 is 16.0
bbox:car_number ((1065, 415), (1079, 459)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1062, 495)) : 2
Length of task list: 5
Number of processes used: 3
```

33% | 418/1261 [18:34<37:28, 2.67s/it]

The times for each task are: [1.397257, 1.015863, 0.70859, 0.359081, 0.382252] with:

Minimum: 0.359081 Maximum: 1.397257 Average: 0.7726 seconds

```
Number of cars 2
bbox:car_number ((925, 400), (1056, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1067, 400), (1087, 462)) : 2
```

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 419/1261 [18:37<37:26, 2.67s/it]

The times for each task are: [0.875621, 1.380378, 0.708403, 0.396205, 0.287112] with:

Minimum: 0.287112 Maximum: 1.380378 Average: 0.7295 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1056, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 420/1261 [18:40<37:23, 2.67s/it]

The times for each task are: [0.641359, 0.919116, 0.303047, 1.413335, 0.499299] with:

Minimum: 0.303047 Maximum: 1.413335 Average: 0.7552 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1048, 495)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 421/1261 [18:43<37:20, 2.67s/it]

The times for each task are: [0.839181, 1.424913, 0.714474, 0.405537, 0.306141] with:

Minimum: 0.306141 Maximum: 1.424913 Average: 0.738 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1062, 495)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

33% | 422/1261 [18:45<37:18, 2.67s/it]

The times for each task are: [0.600902, 1.412488, 1.065585, 0.387833, 0.326133] with:

Minimum: 0.326133 Maximum: 1.412488 Average: 0.7586 seconds

```
Number of cars 1  
bbox:car_number ((925, 400), (1052, 495)) : 1  
The minimum distance from car: 1 is 1.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

34% | 423/1261 [18:48<37:15, 2.67s/it]

The times for each task are: [0.850875, 0.674221, 1.400403, 0.347256, 0.397746] with:

Minimum: 0.347256 Maximum: 1.400403 Average: 0.7341 seconds

```
Number of cars 1  
bbox:car_number ((913, 400), (1047, 492)) : 1  
The minimum distance from car: 1 is 8.06225774829855  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

34% | 424/1261 [18:51<37:12, 2.67s/it]

The times for each task are: [0.88816, 1.339577, 0.605251, 0.390845, 0.302328] with:

Minimum: 0.302328 Maximum: 1.339577 Average: 0.7052 seconds

```
Number of cars 1  
bbox:car_number ((913, 400), (1047, 495)) : 1  
The minimum distance from car: 1 is 1.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
34%| 425/1261 [18:53<37:10, 2.67s/it]
```

```
The times for each task are: [0.621132, 1.014209, 1.593124, 0.398021, 0.454561] with:
```

```
Minimum: 0.398021 Maximum: 1.593124 Average: 0.8162 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (1048, 498)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
34%| 426/1261 [18:56<37:08, 2.67s/it]
```

```
The times for each task are: [0.88707, 0.670255, 1.669421, 0.344173, 0.447854] with:
```

```
Minimum: 0.344173 Maximum: 1.669421 Average: 0.8038 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (1048, 485)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
34%| 427/1261 [18:59<37:05, 2.67s/it]
```

```
The times for each task are: [1.400933, 0.634034, 1.067882, 0.293509, 0.403587] with:
```

```
Minimum: 0.293509 Maximum: 1.400933 Average: 0.76 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (1052, 495)) : 1
The minimum distance from car: 1 is 9.433981132056603
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

34% | 428/1261 [19:02<37:03, 2.67s/it]

The times for each task are: [0.714865, 0.863914, 1.414872, 0.337219, 0.482582] with:

Minimum: 0.337219 Maximum: 1.414872 Average: 0.7627 seconds

Number of cars 1

bbox:car_number ((913, 400), (1047, 495)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

34% | 429/1261 [19:04<37:00, 2.67s/it]

The times for each task are: [0.586317, 0.896904, 1.496434, 0.432472, 0.38817] with:

Minimum: 0.38817 Maximum: 1.496434 Average: 0.7601 seconds

Number of cars 1

bbox:car_number ((913, 400), (1047, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

34% | 430/1261 [19:07<36:57, 2.67s/it]

The times for each task are: [1.05444, 1.500793, 0.673339, 0.484546, 0.346782] with:

Minimum: 0.346782 Maximum: 1.500793 Average: 0.812 seconds

Number of cars 1

bbox:car_number ((925, 400), (1048, 495)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

34% | 431/1261 [19:10<36:55, 2.67s/it]

The times for each task are: [0.850024, 1.488262, 0.74641, 0.428708, 0.357197] with:

Minimum: 0.357197 Maximum: 1.488262 Average: 0.7741 seconds

Number of cars 1

bbox:car_number ((913, 400), (1048, 492)) : 1

The minimum distance from car: 1 is 6.082762530298219

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

34% | 432/1261 [19:12<36:52, 2.67s/it]

The times for each task are: [0.898501, 1.424652, 0.737551, 0.420877, 0.311184] with:

Minimum: 0.311184 Maximum: 1.424652 Average: 0.7586 seconds

Number of cars 1

bbox:car_number ((913, 400), (1047, 492)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

34% | 433/1261 [19:15<36:50, 2.67s/it]

The times for each task are: [0.89727, 0.640167, 1.581812, 0.329824, 0.383284] with:

Minimum: 0.329824 Maximum: 1.581812 Average: 0.7665 seconds

Number of cars 1

bbox:car_number ((940, 400), (1048, 497)) : 1

The minimum distance from car: 1 is 14.142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

34% | 434/1261 [19:18<36:47, 2.67s/it]

The times for each task are: [0.625605, 1.412091, 0.436985, 0.387883, 1.024559] with:

Minimum: 0.387883 Maximum: 1.412091 Average: 0.7774 seconds

Number of cars 1

bbox:car_number ((925, 400), (1048, 497)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

34% | 435/1261 [19:20<36:44, 2.67s/it]

The times for each task are: [0.636792, 0.903439, 1.411141, 0.38763, 0.293179] with:

Minimum: 0.293179 Maximum: 1.411141 Average: 0.7264 seconds

Number of cars 1

bbox:car_number ((910, 400), (1047, 492)) : 1

The minimum distance from car: 1 is 8.246211251235321

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

35% | 436/1261 [19:23<36:41, 2.67s/it]

The times for each task are: [0.660531, 0.920907, 0.34383, 0.490842, 1.766268] with:

Minimum: 0.34383 Maximum: 1.766268 Average: 0.8365 seconds

Number of cars 1

bbox:car_number ((913, 400), (1047, 492)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

35% | 437/1261 [19:26<36:38, 2.67s/it]

The times for each task are: [0.839578, 0.694963, 1.464257, 0.524684, 0.313185] with:

Minimum: 0.313185 Maximum: 1.464257 Average: 0.7673 seconds

Number of cars 1

bbox:car_number ((910, 400), (1048, 495)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

35% | 438/1261 [19:28<36:35, 2.67s/it]

The times for each task are: [0.625435, 1.039262, 1.530831, 0.341191, 0.395584] with:

Minimum: 0.341191 Maximum: 1.530831 Average: 0.7865 seconds

Number of cars 1

bbox:car_number ((940, 400), (1047, 492)) : 1

The minimum distance from car: 1 is 14.035668847618199

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

35% | 439/1261 [19:31<36:32, 2.67s/it]

The times for each task are: [0.827256, 0.734964, 0.407043, 0.370627, 1.663659] with:

Minimum: 0.370627 Maximum: 1.663659 Average: 0.8007 seconds

Number of cars 1

bbox:car_number ((940, 400), (1052, 492)) : 1

The minimum distance from car: 1 is 3.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

35% | 440/1261 [19:33<36:30, 2.67s/it]

The times for each task are: [0.851282, 1.385411, 0.701596, 0.389437, 0.27774] with:

Minimum: 0.27774 Maximum: 1.385411 Average: 0.7211 seconds

Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 9.055385138137417
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

35% | 441/1261 [19:36<36:27, 2.67s/it]

The times for each task are: [0.854972, 1.484416, 0.705163, 0.306994, 0.408366] with:

Minimum: 0.306994 Maximum: 1.484416 Average: 0.752 seconds

Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

35% | 442/1261 [19:39<36:24, 2.67s/it]

The times for each task are: [0.661857, 0.868746, 1.394424, 0.472576, 0.315396] with:

Minimum: 0.315396 Maximum: 1.394424 Average: 0.7426 seconds

Number of cars 1
bbox:car_number ((913, 400), (1056, 492)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

35% | 443/1261 [19:42<36:22, 2.67s/it]

The times for each task are: [0.905495, 0.751192, 1.397379, 0.405351, 0.345632] with:

Minimum: 0.345632 Maximum: 1.397379 Average: 0.761 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 6.082762530298219
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

35% | 444/1261 [19:44<36:19, 2.67s/it]

The times for each task are: [0.634287, 0.948216, 1.439358, 0.378099, 0.450335] with:

Minimum: 0.378099 Maximum: 1.439358 Average: 0.7701 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 9.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

35% | 445/1261 [19:47<36:16, 2.67s/it]

The times for each task are: [0.650857, 1.007267, 1.594311, 0.410257, 0.30202] with:

Minimum: 0.30202 Maximum: 1.594311 Average: 0.7929 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

35% | 446/1261 [19:49<36:14, 2.67s/it]

The times for each task are: [1.366768, 0.905687, 0.608898, 0.329904, 0.525639] with:

Minimum: 0.329904 Maximum: 1.366768 Average: 0.7474 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (1029, 495)) : 1
The minimum distance from car: 1 is 10.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

35%| 447/1261 [19:52<36:11, 2.67s/it]

The times for each task are: [1.402328, 1.017371, 0.613683, 0.388888, 0.347766] with:
Minimum: 0.347766 Maximum: 1.402328 Average: 0.754 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1029, 495)) : 1
The minimum distance from car: 1 is 9.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

36%| 448/1261 [19:54<36:08, 2.67s/it]

The times for each task are: [0.915344, 0.706683, 1.449375, 0.369208, 0.385193] with:
Minimum: 0.369208 Maximum: 1.449375 Average: 0.7652 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1029, 485)) : 1
The minimum distance from car: 1 is 7.810249675906654
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

36%| 449/1261 [19:57<36:05, 2.67s/it]

The times for each task are: [0.952683, 0.649235, 1.723963, 0.496238, 0.337526] with:
Minimum: 0.337526 Maximum: 1.723963 Average: 0.8319 seconds

Number of cars 1

```
bbox:car_number ((927, 400), (1052, 495)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 450/1261 [20:00<36:02, 2.67s/it]

The times for each task are: [0.75214, 0.910812, 0.474481, 1.384203, 0.295563] with:

Minimum: 0.295563 Maximum: 1.384203 Average: 0.7634 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1056, 495)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 451/1261 [20:03<36:00, 2.67s/it]

The times for each task are: [0.905469, 0.744629, 1.460279, 0.415086, 0.342443] with:

Minimum: 0.342443 Maximum: 1.460279 Average: 0.7736 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1052, 498)) : 1
The minimum distance from car: 1 is 9.219544457292887
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 452/1261 [20:05<35:57, 2.67s/it]

The times for each task are: [0.621476, 0.908463, 1.613485, 0.474677, 0.444272] with:

Minimum: 0.444272 Maximum: 1.613485 Average: 0.8125 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1077, 498)) : 1
```

```
The minimum distance from car: 1 is 26.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 453/1261 [20:08<35:54, 2.67s/it]

The times for each task are: [0.913606, 0.630598, 1.608063, 0.287369, 0.36549] with:

Minimum: 0.287369 Maximum: 1.608063 Average: 0.761 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 28.071337695236398
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 454/1261 [20:10<35:52, 2.67s/it]

The times for each task are: [0.850995, 0.750164, 1.60655, 0.485164, 0.298725] with:

Minimum: 0.298725 Maximum: 1.60655 Average: 0.7983 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 455/1261 [20:13<35:49, 2.67s/it]

The times for each task are: [0.641242, 1.419032, 1.009493, 0.300338, 0.558883] with:

Minimum: 0.300338 Maximum: 1.419032 Average: 0.7858 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 7.0
```

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 456/1261 [20:16<35:46, 2.67s/it]

The times for each task are: [0.846118, 1.433613, 0.681667, 0.37919, 0.309676] with:

Minimum: 0.309676 Maximum: 1.433613 Average: 0.7301 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 457/1261 [20:18<35:44, 2.67s/it]

The times for each task are: [0.913392, 1.443249, 0.680237, 0.400665, 0.2804] with:

Minimum: 0.2804 Maximum: 1.443249 Average: 0.7436 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1032, 497)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 458/1261 [20:21<35:41, 2.67s/it]

The times for each task are: [1.054361, 0.594221, 1.432571, 0.330063, 0.484347] with:

Minimum: 0.330063 Maximum: 1.432571 Average: 0.7791 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (1029, 498)) : 1
The minimum distance from car: 1 is 10.04987562112089
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

36% | 459/1261 [20:24<35:38, 2.67s/it]

The times for each task are: [1.403109, 0.588421, 1.064835, 0.285047, 0.380362] with:

Minimum: 0.285047 Maximum: 1.403109 Average: 0.7444 seconds

```
Number of cars 1  
bbox:car_number ((927, 400), (1032, 497)) : 1  
The minimum distance from car: 1 is 10.04987562112089  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

36% | 460/1261 [20:26<35:36, 2.67s/it]

The times for each task are: [0.633012, 1.002408, 1.423646, 0.392015, 0.330171] with:

Minimum: 0.330171 Maximum: 1.423646 Average: 0.7563 seconds

```
Number of cars 1  
bbox:car_number ((910, 400), (1032, 495)) : 1  
The minimum distance from car: 1 is 8.06225774829855  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

37% | 461/1261 [20:29<35:33, 2.67s/it]

The times for each task are: [0.888437, 0.671489, 1.402351, 0.306459, 0.458663] with:

Minimum: 0.306459 Maximum: 1.402351 Average: 0.7455 seconds

```
Number of cars 1  
bbox:car_number ((927, 400), (1032, 495)) : 1  
The minimum distance from car: 1 is 8.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
37%|      | 462/1261 [20:32<35:31,  2.67s/it]
```

```
The times for each task are: [0.933715, 1.414868, 0.695738, 0.386911, 0.293389] with:
```

```
Minimum: 0.293389 Maximum: 1.414868 Average: 0.7449 seconds
```

```
Number of cars 1
bbox:car_number ((927, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
37%|      | 463/1261 [20:34<35:27,  2.67s/it]
```

```
The times for each task are: [0.850477, 1.407683, 0.62779, 0.414908, 0.300148] with:
```

```
Minimum: 0.300148 Maximum: 1.407683 Average: 0.7202 seconds
```

```
Number of cars 1
bbox:car_number ((896, 400), (1056, 495)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
37%|      | 464/1261 [20:36<35:24,  2.67s/it]
```

```
The times for each task are: [0.942492, 0.631807, 1.679223, 0.353592, 0.417567] with:
```

```
Minimum: 0.353592 Maximum: 1.679223 Average: 0.8049 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

37% | 465/1261 [20:39<35:21, 2.67s/it]

The times for each task are: [0.679062, 1.434502, 0.906321, 0.401949, 0.310935] with:

Minimum: 0.310935 Maximum: 1.434502 Average: 0.7466 seconds

Number of cars 1

bbox:car_number ((913, 400), (1029, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

37% | 466/1261 [20:41<35:18, 2.66s/it]

The times for each task are: [0.924115, 0.733148, 1.406526, 0.419297, 0.315603] with:

Minimum: 0.315603 Maximum: 1.406526 Average: 0.7597 seconds

Number of cars 1

bbox:car_number ((905, 400), (1029, 497)) : 1

The minimum distance from car: 1 is 4.123105625617661

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

37% | 467/1261 [20:44<35:15, 2.66s/it]

The times for each task are: [1.334704, 0.816363, 0.694617, 0.408866, 0.363495] with:

Minimum: 0.363495 Maximum: 1.334704 Average: 0.7236 seconds

Number of cars 1

bbox:car_number ((905, 400), (1029, 495)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

37% | 468/1261 [20:46<35:12, 2.66s/it]

The times for each task are: [1.023041, 0.691949, 1.433528, 0.44419, 0.336869] with:

Minimum: 0.336869 Maximum: 1.433528 Average: 0.7859 seconds

Number of cars 1
bbox:car_number ((927, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 9.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

37% | 469/1261 [20:49<35:10, 2.66s/it]

The times for each task are: [0.599389, 0.890266, 0.440344, 0.289975, 1.663033] with:

Minimum: 0.289975 Maximum: 1.663033 Average: 0.7766 seconds

Number of cars 1
bbox:car_number ((927, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

37% | 470/1261 [20:52<35:07, 2.66s/it]

The times for each task are: [1.368466, 0.68076, 1.018552, 0.41076, 0.364648] with:

Minimum: 0.364648 Maximum: 1.368466 Average: 0.7686 seconds

Number of cars 1
bbox:car_number ((913, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 8.602325267042627
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

37% | 471/1261 [20:55<35:05, 2.66s/it]

The times for each task are: [0.912877, 0.662181, 1.691966, 0.33225, 0.438978] with:

Minimum: 0.33225 Maximum: 1.691966 Average: 0.8077 seconds

Number of cars 1

bbox:car_number ((929, 400), (1024, 495)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

37% | 472/1261 [20:57<35:02, 2.66s/it]

The times for each task are: [0.649152, 1.537674, 0.92291, 0.499438, 0.395512] with:

Minimum: 0.395512 Maximum: 1.537674 Average: 0.8009 seconds

Number of cars 1

bbox:car_number ((927, 400), (1025, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 473/1261 [21:00<34:59, 2.66s/it]

The times for each task are: [0.62186, 0.952379, 1.36891, 0.552965, 0.313122] with:

Minimum: 0.313122 Maximum: 1.36891 Average: 0.7618 seconds

Number of cars 1

bbox:car_number ((913, 400), (1025, 495)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 474/1261 [21:02<34:56, 2.66s/it]

The times for each task are: [0.892702, 0.619173, 1.507361, 0.329819, 0.507893] with:

Minimum: 0.329819 Maximum: 1.507361 Average: 0.7714 seconds

Number of cars 1

bbox:car_number ((910, 400), (1024, 495)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 475/1261 [21:05<34:54, 2.66s/it]

The times for each task are: [0.895247, 0.603077, 1.686964, 0.291481, 0.379632] with:

Minimum: 0.291481 Maximum: 1.686964 Average: 0.7713 seconds

Number of cars 1

bbox:car_number ((927, 400), (1025, 485)) : 1

The minimum distance from car: 1 is 10.295630140987

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 476/1261 [21:08<34:51, 2.66s/it]

The times for each task are: [0.618662, 0.997581, 1.514862, 0.410281, 0.348685] with:

Minimum: 0.348685 Maximum: 1.514862 Average: 0.778 seconds

Number of cars 1

bbox:car_number ((910, 400), (1025, 495)) : 1

The minimum distance from car: 1 is 10.295630140987

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 477/1261 [21:11<34:49, 2.66s/it]

The times for each task are: [0.846046, 1.575124, 0.62135, 0.446082, 0.278856] with:

Minimum: 0.278856 Maximum: 1.575124 Average: 0.7535 seconds

Number of cars 2
bbox:car_number ((913, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((896, 430), (908, 462)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 478/1261 [21:13<34:46, 2.67s/it]

The times for each task are: [0.889038, 0.6101, 0.517506, 0.295862, 1.51027] with:

Minimum: 0.295862 Maximum: 1.51027 Average: 0.7646 seconds

Number of cars 1
bbox:car_number ((896, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 10.295630140987
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 479/1261 [21:16<34:44, 2.67s/it]

The times for each task are: [0.992974, 0.700857, 1.340292, 0.444603, 0.367099] with:

Minimum: 0.367099 Maximum: 1.340292 Average: 0.7692 seconds

Number of cars 1
bbox:car_number ((896, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 480/1261 [21:19<34:41, 2.66s/it]

The times for each task are: [0.918766, 1.362337, 0.619482, 0.300392, 0.400796] with:

Minimum: 0.300392 Maximum: 1.362337 Average: 0.7204 seconds

Number of cars 1
bbox:car_number ((895, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 481/1261 [21:21<34:38, 2.66s/it]

The times for each task are: [0.836967, 1.385808, 0.394767, 0.725295, 0.289408] with:

Minimum: 0.289408 Maximum: 1.385808 Average: 0.7264 seconds

Number of cars 1
bbox:car_number ((895, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 482/1261 [21:24<34:35, 2.66s/it]

The times for each task are: [0.977776, 0.669498, 1.379974, 0.472705, 0.386756] with:

Minimum: 0.386756 Maximum: 1.379974 Average: 0.7773 seconds

Number of cars 1
bbox:car_number ((895, 400), (1024, 495)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 483/1261 [21:27<34:33, 2.67s/it]

The times for each task are: [1.546811, 0.645499, 0.91266, 0.491647, 0.390394] with:

Minimum: 0.390394 Maximum: 1.546811 Average: 0.7974 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

38% | 484/1261 [21:30<34:31, 2.67s/it]

The times for each task are: [0.979805, 0.735765, 1.459592, 0.327086, 0.382095] with:

Minimum: 0.327086 Maximum: 1.459592 Average: 0.7769 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1024, 492)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

38% | 485/1261 [21:32<34:28, 2.67s/it]

The times for each task are: [0.897759, 1.365478, 0.583414, 0.298851, 0.403027] with:

Minimum: 0.298851 Maximum: 1.365478 Average: 0.7097 seconds

```
Number of cars 2
bbox:car_number ((895, 400), (1025, 492)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1225, 445), (1227, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 486/1261 [21:35<34:25, 2.67s/it]

The times for each task are: [0.838076, 0.661078, 1.391698, 0.399139, 0.350682] with:

Minimum: 0.350682 Maximum: 1.391698 Average: 0.7281 seconds

```
Number of cars 1
bbox:car_number ((895, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 487/1261 [21:37<34:22, 2.66s/it]

The times for each task are: [0.953171, 0.620728, 1.355259, 0.38289, 0.283516] with:

Minimum: 0.283516 Maximum: 1.355259 Average: 0.7191 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 488/1261 [21:40<34:20, 2.67s/it]

The times for each task are: [0.898785, 0.694849, 0.392518, 1.39269, 0.332222] with:

Minimum: 0.332222 Maximum: 1.39269 Average: 0.7422 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 489/1261 [21:42<34:16, 2.66s/it]

The times for each task are: [0.947031, 0.642639, 1.426489, 0.29294, 0.460145] with:

Minimum: 0.29294 Maximum: 1.426489 Average: 0.7538 seconds

```
Number of cars 2
bbox:car_number ((925, 400), (1024, 492)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((896, 430), (915, 459)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 490/1261 [21:45<34:13, 2.66s/it]

The times for each task are: [0.910964, 0.705692, 1.363316, 0.444544, 0.307002] with:

Minimum: 0.307002 Maximum: 1.363316 Average: 0.7463 seconds

```
Number of cars 4
bbox:car_number ((925, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((895, 415), (912, 447)) : 2
bbox:car_number ((1180, 445), (1197, 477)) : 3
bbox:car_number ((1210, 445), (1227, 477)) : 4
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 491/1261 [21:47<34:10, 2.66s/it]

The times for each task are: [1.386867, 0.567861, 1.010311, 0.409988, 0.373523] with:

Minimum: 0.373523 Maximum: 1.386867 Average: 0.7497 seconds

```
Number of cars 1
bbox:car_number ((895, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 492/1261 [21:50<34:07, 2.66s/it]

The times for each task are: [0.976672, 1.352829, 0.706391, 0.298837, 0.393705] with:

Minimum: 0.298837 Maximum: 1.352829 Average: 0.7457 seconds

Number of cars 2
bbox:car_number ((895, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 13.0
bbox:car_number ((1165, 445), (1181, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1175, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400), (1219, 492))]
Length of task list: 5
Number of processes used: 3

39% | 493/1261 [21:53<34:06, 2.66s/it]

The times for each task are: [0.883587, 0.694788, 1.343828, 0.389461, 0.388546] with:

Minimum: 0.388546 Maximum: 1.343828 Average: 0.74 seconds
Number of cars 3
bbox:car_number ((913, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 11.180339887498949
bbox:car_number ((1180, 438), (1200, 477)) : 2
bbox:car_number ((1210, 445), (1227, 477)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1175, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400), (1219, 492))]
Length of task list: 5
Number of processes used: 3

39% | 494/1261 [21:55<34:02, 2.66s/it]

The times for each task are: [0.83808, 1.380942, 0.695871, 0.472396, 0.321702] with:

Minimum: 0.321702 Maximum: 1.380942 Average: 0.7418 seconds

Number of cars 2
bbox:car_number ((896, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 10.295630140987
bbox:car_number ((1160, 438), (1219, 492)) : 2
The minimum distance from car: 1 is 23.53720459187964
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1175, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400), (1219, 492))]
Length of task list: 5
Number of processes used: 3

39% | 495/1261 [21:58<34:00, 2.66s/it]

The times for each task are: [0.955355, 0.676896, 1.518658, 0.437101, 0.373296] with:

Minimum: 0.373296 Maximum: 1.518658 Average: 0.7923 seconds

Number of cars 2

bbox:car_number ((896, 400), (1024, 495)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1179, 445), (1212, 477)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

39% | 496/1261 [22:00<33:57, 2.66s/it]

The times for each task are: [0.920996, 0.636688, 1.393064, 0.410594, 0.313399] with:

Minimum: 0.313399 Maximum: 1.393064 Average: 0.7349 seconds

Number of cars 2

bbox:car_number ((927, 400), (1010, 485)) : 1

The minimum distance from car: 1 is 9.433981132056603

bbox:car_number ((1210, 445), (1212, 477)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

39% | 497/1261 [22:03<33:54, 2.66s/it]

The times for each task are: [0.624094, 1.02533, 0.417458, 1.407613, 0.343364] with:

Minimum: 0.343364 Maximum: 1.407613 Average: 0.7636 seconds

Number of cars 1

bbox:car_number ((927, 400), (1017, 485)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

39% | 498/1261 [22:06<33:52, 2.66s/it]

The times for each task are: [0.687459, 0.833652, 1.359025, 0.30514, 0.508481] with:

Minimum: 0.30514 Maximum: 1.359025 Average: 0.7388 seconds

Number of cars 1

bbox:car_number ((913, 400), (1025, 495)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

40% | 499/1261 [22:09<33:49, 2.66s/it]

The times for each task are: [0.908081, 0.626537, 1.666298, 0.349782, 0.482717] with:

Minimum: 0.349782 Maximum: 1.666298 Average: 0.8067 seconds

Number of cars 3

bbox:car_number ((913, 400), (1025, 485)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1180, 445), (1197, 477)) : 2

bbox:car_number ((1210, 445), (1227, 477)) : 3

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

40% | 500/1261 [22:11<33:46, 2.66s/it]

The times for each task are: [0.887501, 0.613987, 1.654067, 0.303682, 0.393468] with:

Minimum: 0.303682 Maximum: 1.654067 Average: 0.7705 seconds

Number of cars 1

bbox:car_number ((940, 400), (1024, 485)) : 1

The minimum distance from car: 1 is 13.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

```
Length of task list: 5
Number of processes used: 3
```

```
40%|      | 501/1261 [22:14<33:43, 2.66s/it]
```

```
The times for each task are: [0.62095, 1.399161, 0.889838, 0.285709, 0.400183] with:
```

```
Minimum: 0.285709 Maximum: 1.399161 Average: 0.7192 seconds
```

```
Number of cars 0
Length of task list: 5
Number of processes used: 3
```

```
40%|      | 502/1261 [22:16<33:41, 2.66s/it]
```

```
The times for each task are: [0.870598, 0.613605, 1.546655, 0.390141, 0.383787] with:
```

```
Minimum: 0.383787 Maximum: 1.546655 Average: 0.761 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (998, 459)) : 1
The minimum distance from car: 1 is 29.966648127543394
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
40%|      | 503/1261 [22:19<33:38, 2.66s/it]
```

```
The times for each task are: [0.656306, 0.962193, 1.450172, 0.382033, 0.291758] with:
```

```
Minimum: 0.291758 Maximum: 1.450172 Average: 0.7485 seconds
```

```
Number of cars 3
bbox:car_number ((929, 415), (942, 459)) : 1
bbox:car_number ((951, 419), (953, 459)) : 2
bbox:car_number ((955, 419), (998, 462)) : 3
Length of task list: 5
Number of processes used: 3
```

```
40%|      | 504/1261 [22:22<33:35, 2.66s/it]
```

The times for each task are: [0.885827, 1.359634, 0.709218, 0.294961, 0.38382] with:

Minimum: 0.294961 Maximum: 1.359634 Average: 0.7267 seconds

Number of cars 2

bbox:car_number ((951, 400), (998, 459)) : 1

bbox:car_number ((929, 415), (942, 459)) : 2

Length of task list: 5

Number of processes used: 3

40% | 505/1261 [22:24<33:33, 2.66s/it]

The times for each task are: [0.933905, 0.623111, 1.520144, 0.536248, 0.322456] with:

Minimum: 0.322456 Maximum: 1.520144 Average: 0.7872 seconds

Number of cars 2

bbox:car_number ((929, 415), (942, 447)) : 1

bbox:car_number ((951, 419), (953, 459)) : 2

Length of task list: 5

Number of processes used: 3

40% | 506/1261 [22:27<33:30, 2.66s/it]

The times for each task are: [0.611225, 1.406835, 0.993825, 0.353921, 0.461426] with:

Minimum: 0.353921 Maximum: 1.406835 Average: 0.7654 seconds

Number of cars 2

bbox:car_number ((929, 400), (942, 459)) : 1

bbox:car_number ((951, 400), (991, 459)) : 2

Length of task list: 5

Number of processes used: 3

40% | 507/1261 [22:30<33:27, 2.66s/it]

The times for each task are: [0.938667, 0.641685, 1.521669, 0.349785, 0.419436] with:

Minimum: 0.349785 Maximum: 1.521669 Average: 0.7742 seconds

Number of cars 1

bbox:car_number ((913, 400), (1002, 472)) : 1

The minimum distance from car: 1 is 7.280109889280518

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (929, 400))]
Length of task list: 5
Number of processes used: 3
```

40% | 508/1261 [22:33<33:25, 2.66s/it]

The times for each task are: [0.618301, 1.079743, 1.39386, 0.44186, 0.293028] with:

Minimum: 0.293028 Maximum: 1.39386 Average: 0.7654 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (998, 472)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (929, 400))]
Length of task list: 5
Number of processes used: 3
```

40% | 509/1261 [22:35<33:22, 2.66s/it]

The times for each task are: [0.608859, 1.043821, 1.417526, 0.300505, 0.400373] with:

Minimum: 0.300505 Maximum: 1.417526 Average: 0.7542 seconds

```
Number of cars 2
bbox:car_number ((929, 415), (942, 459)) : 1
bbox:car_number ((951, 419), (953, 459)) : 2
Length of task list: 5
Number of processes used: 3
```

40% | 510/1261 [22:38<33:19, 2.66s/it]

The times for each task are: [0.615288, 1.023963, 1.412502, 0.3083, 0.403905] with:

Minimum: 0.3083 Maximum: 1.412502 Average: 0.7528 seconds

```
Number of cars 2
bbox:car_number ((932, 400), (942, 447)) : 1
bbox:car_number ((951, 400), (998, 459)) : 2
Length of task list: 5
Number of processes used: 3
```

41% | 511/1261 [22:40<33:17, 2.66s/it]

The times for each task are: [0.968767, 0.555228, 1.569419, 0.288016, 0.495846] with:

Minimum: 0.288016 Maximum: 1.569419 Average: 0.7755 seconds

Number of cars 1

bbox:car_number ((929, 400), (998, 462)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 512/1261 [22:43<33:14, 2.66s/it]

The times for each task are: [1.409018, 0.608405, 1.059421, 0.372701, 0.462101] with:

Minimum: 0.372701 Maximum: 1.409018 Average: 0.7823 seconds

Number of cars 1

bbox:car_number ((925, 400), (991, 472)) : 1

The minimum distance from car: 1 is 7.0710678118654755

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 513/1261 [22:45<33:11, 2.66s/it]

The times for each task are: [0.73774, 1.445573, 0.9248, 0.339129, 0.454055] with:

Minimum: 0.339129 Maximum: 1.445573 Average: 0.7803 seconds

Number of cars 1

bbox:car_number ((932, 400), (1001, 462)) : 1

The minimum distance from car: 1 is 9.433981132056603

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 514/1261 [22:48<33:08, 2.66s/it]

The times for each task are: [0.615155, 1.395285, 1.072851, 0.403295, 0.353582] with:

Minimum: 0.353582 Maximum: 1.395285 Average: 0.768 seconds

Number of cars 1

bbox:car_number ((925, 400), (1002, 472)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 515/1261 [22:51<33:06, 2.66s/it]

The times for each task are: [0.897467, 0.581391, 1.610738, 0.3947, 0.371984] with:

Minimum: 0.371984 Maximum: 1.610738 Average: 0.7713 seconds

Number of cars 1

bbox:car_number ((929, 400), (1001, 478)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 516/1261 [22:53<33:03, 2.66s/it]

The times for each task are: [0.587793, 1.446059, 0.984547, 0.44347, 0.336767] with:

Minimum: 0.336767 Maximum: 1.446059 Average: 0.7597 seconds

Number of cars 1

bbox:car_number ((913, 400), (1010, 485)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 517/1261 [22:56<33:00, 2.66s/it]

The times for each task are: [0.955563, 1.400452, 0.628381, 0.39936, 0.297797] with:

Minimum: 0.297797 Maximum: 1.400452 Average: 0.7363 seconds

Number of cars 1
bbox:car_number ((913, 400), (1001, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

41% | 518/1261 [22:58<32:57, 2.66s/it]

The times for each task are: [0.919525, 0.691872, 1.684875, 0.355484, 0.516491] with:

Minimum: 0.355484 Maximum: 1.684875 Average: 0.8336 seconds

Number of cars 1
bbox:car_number ((925, 400), (1001, 478)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

41% | 519/1261 [23:01<32:54, 2.66s/it]

The times for each task are: [0.664911, 0.976846, 1.523261, 0.302843, 0.402689] with:

Minimum: 0.302843 Maximum: 1.523261 Average: 0.7741 seconds

Number of cars 1
bbox:car_number ((913, 400), (1010, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

41% | 520/1261 [23:04<32:52, 2.66s/it]

The times for each task are: [1.118682, 0.739696, 1.503989, 0.349864, 0.468956] with:

Minimum: 0.349864 Maximum: 1.503989 Average: 0.8362 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (998, 478))]
Length of task list: 5
Number of processes used: 3
```

41% | 521/1261 [23:06<32:49, 2.66s/it]

The times for each task are: [0.685611, 1.63631, 0.878104, 0.534018, 0.30583] with:

Minimum: 0.30583 Maximum: 1.63631 Average: 0.808 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (998, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (998, 478))]
Length of task list: 5
Number of processes used: 3
```

41% | 522/1261 [23:09<32:47, 2.66s/it]

The times for each task are: [1.056602, 1.422267, 0.734783, 0.328219, 0.459932] with:

Minimum: 0.328219 Maximum: 1.422267 Average: 0.8004 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (998, 478))]
Length of task list: 5
Number of processes used: 3
```

41% | 523/1261 [23:12<32:44, 2.66s/it]

The times for each task are: [0.686254, 1.457792, 0.925122, 0.348311, 0.467273] with:

Minimum: 0.348311 Maximum: 1.457792 Average: 0.777 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1010, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42%| 524/1261 [23:14<32:41, 2.66s/it]

The times for each task are: [0.920401, 0.580085, 1.472777, 0.40424, 0.316205] with:
Minimum: 0.316205 Maximum: 1.472777 Average: 0.7387 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1010, 472)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42%| 525/1261 [23:17<32:39, 2.66s/it]

The times for each task are: [0.572665, 1.405444, 1.05103, 0.407371, 0.287659] with:
Minimum: 0.287659 Maximum: 1.405444 Average: 0.7448 seconds

```
Number of cars 1
bbox:car_number ((940, 415), (991, 462)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42%| 526/1261 [23:20<32:36, 2.66s/it]

The times for each task are: [0.618924, 1.417092, 0.906109, 0.411673, 0.334695] with:
Minimum: 0.334695 Maximum: 1.417092 Average: 0.7377 seconds

Number of cars 1

```
bbox:car_number ((932, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 2.23606797749979
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 527/1261 [23:23<32:34, 2.66s/it]

The times for each task are: [0.613299, 1.36629, 1.012048, 0.288995, 0.446411] with:
Minimum: 0.288995 Maximum: 1.36629 Average: 0.7454 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1010, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 528/1261 [23:25<32:31, 2.66s/it]

The times for each task are: [0.919546, 0.630876, 1.635004, 0.472619, 0.333518] with:
Minimum: 0.333518 Maximum: 1.635004 Average: 0.7983 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1010, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 529/1261 [23:28<32:28, 2.66s/it]

The times for each task are: [1.440741, 0.948642, 0.656296, 0.546438, 0.29189] with:
Minimum: 0.29189 Maximum: 1.440741 Average: 0.7768 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1010, 478)) : 1
```

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

42% | 530/1261 [23:31<32:26, 2.66s/it]

The times for each task are: [1.037805, 0.697972, 1.485828, 0.369676, 0.478876] with:

Minimum: 0.369676 Maximum: 1.485828 Average: 0.814 seconds

Number of cars 2

bbox:car_number ((925, 400), (1010, 477)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((552, 490), (567, 507)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

42% | 531/1261 [23:34<32:24, 2.66s/it]

The times for each task are: [0.944224, 0.706376, 1.588179, 0.486623, 0.312517] with:

Minimum: 0.312517 Maximum: 1.588179 Average: 0.8076 seconds

Number of cars 2

bbox:car_number ((925, 400), (998, 472)) : 1

The minimum distance from car: 1 is 6.324555320336759

bbox:car_number ((1000, 419), (1001, 462)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

42% | 532/1261 [23:36<32:21, 2.66s/it]

The times for each task are: [0.872801, 0.621416, 1.632978, 0.411966, 0.31349] with:

Minimum: 0.31349 Maximum: 1.632978 Average: 0.7705 seconds

Number of cars 1

```
bbox:car_number ((913, 400), (1010, 485)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 533/1261 [23:39<32:18, 2.66s/it]

The times for each task are: [0.866131, 1.55532, 0.676342, 0.369248, 0.483044] with:
Minimum: 0.369248 Maximum: 1.55532 Average: 0.79 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (994, 462)) : 1
The minimum distance from car: 1 is 11.180339887498949
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 534/1261 [23:42<32:16, 2.66s/it]

The times for each task are: [0.61853, 1.076319, 1.372713, 0.393436, 0.341588] with:
Minimum: 0.341588 Maximum: 1.372713 Average: 0.7605 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1010, 485)) : 1
The minimum distance from car: 1 is 12.529964086141668
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 535/1261 [23:44<32:13, 2.66s/it]

The times for each task are: [0.836665, 0.707014, 0.345703, 1.596156, 0.451832] with:
Minimum: 0.345703 Maximum: 1.596156 Average: 0.7875 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1010, 478)) : 1
```

```
The minimum distance from car: 1 is 8.54400374531753
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (932, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 536/1261 [23:47<32:10, 2.66s/it]

The times for each task are: [0.918445, 0.623973, 1.358702, 0.399429, 0.293151] with:

Minimum: 0.293151 Maximum: 1.358702 Average: 0.7187 seconds

```
Number of cars 1
bbox:car_number ((932, 400), (1001, 472)) : 1
The minimum distance from car: 1 is 5.830951894845301
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (932, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 537/1261 [23:49<32:07, 2.66s/it]

The times for each task are: [0.89065, 0.695481, 1.375427, 0.393608, 0.343824] with:

Minimum: 0.343824 Maximum: 1.375427 Average: 0.7398 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 478)) : 1
The minimum distance from car: 1 is 13.341664064126334
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 538/1261 [23:52<32:04, 2.66s/it]

The times for each task are: [0.595629, 0.888173, 1.446135, 0.347572, 0.418421] with:

Minimum: 0.347572 Maximum: 1.446135 Average: 0.7392 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1001, 472)) : 1
The minimum distance from car: 1 is 10.44030650891055
```

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 539/1261 [23:55<32:02, 2.66s/it]

The times for each task are: [0.593252, 1.354841, 0.992221, 0.347857, 0.402994] with:

Minimum: 0.347857 Maximum: 1.354841 Average: 0.7382 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 472)) : 1
The minimum distance from car: 1 is 10.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 540/1261 [23:57<31:59, 2.66s/it]

The times for each task are: [1.004753, 1.395731, 0.743017, 0.40449, 0.339699] with:

Minimum: 0.339699 Maximum: 1.395731 Average: 0.7775 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 541/1261 [24:00<31:56, 2.66s/it]

The times for each task are: [0.937158, 0.640739, 1.43021, 0.293872, 0.432725] with:

Minimum: 0.293872 Maximum: 1.43021 Average: 0.7469 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 13.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (925, 400), (1002, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 542/1261 [24:02<31:53,  2.66s/it]
```

```
The times for each task are: [0.858702, 1.394037, 0.636484, 0.352294, 0.520203] with:
```

```
Minimum: 0.352294 Maximum: 1.394037 Average: 0.7523 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (1002, 485)) : 1
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 543/1261 [24:05<31:51,  2.66s/it]
```

```
The times for each task are: [1.488647, 0.63, 1.099707, 0.410501, 0.297675] with:
```

```
Minimum: 0.297675 Maximum: 1.488647 Average: 0.7853 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (1002, 485)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (998, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 544/1261 [24:08<31:49,  2.66s/it]
```

```
The times for each task are: [0.651977, 1.145468, 0.525038, 1.452178, 0.344154] with:
```

```
Minimum: 0.344154 Maximum: 1.452178 Average: 0.8238 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (998, 478)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (998, 478)) : 1
```

```
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 545/1261 [24:11<31:47, 2.66s/it]
```

```
The times for each task are: [1.037825, 0.631877, 1.457716, 0.317141, 0.453138] with:
```

```
Minimum: 0.317141 Maximum: 1.457716 Average: 0.7795 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (998, 477)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 546/1261 [24:14<31:44, 2.66s/it]
```

```
The times for each task are: [0.88614, 0.612136, 1.64793, 0.354872, 0.445285] with:
```

```
Minimum: 0.354872 Maximum: 1.64793 Average: 0.7893 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (991, 472)) : 1
The minimum distance from car: 1 is 11.180339887498949
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 547/1261 [24:17<31:42, 2.66s/it]
```

```
The times for each task are: [0.976984, 0.700937, 1.521019, 0.397823, 0.304208] with:
```

```
Minimum: 0.304208 Maximum: 1.521019 Average: 0.7802 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (991, 472)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

43% | 548/1261 [24:20<31:39, 2.66s/it]

The times for each task are: [0.66729, 1.504089, 1.049065, 0.30812, 0.412216] with:

Minimum: 0.30812 Maximum: 1.504089 Average: 0.7882 seconds

Number of cars 1

bbox:car_number ((896, 400), (998, 478)) : 1

The minimum distance from car: 1 is 11.40175425099138

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 549/1261 [24:22<31:36, 2.66s/it]

The times for each task are: [0.854831, 0.683892, 1.562935, 0.431404, 0.376678] with:

Minimum: 0.376678 Maximum: 1.562935 Average: 0.7819 seconds

Number of cars 1

bbox:car_number ((896, 400), (994, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 550/1261 [24:25<31:34, 2.66s/it]

The times for each task are: [0.578982, 0.903982, 1.598735, 0.309309, 0.487802] with:

Minimum: 0.309309 Maximum: 1.598735 Average: 0.7758 seconds

Number of cars 1

bbox:car_number ((896, 400), (991, 472)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 551/1261 [24:27<31:31, 2.66s/it]

The times for each task are: [0.867757, 0.687839, 1.574225, 0.449239, 0.306806] with:

Minimum: 0.306806 Maximum: 1.574225 Average: 0.7772 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 472)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (989, 459))]

Length of task list: 5

Number of processes used: 3

44% | 552/1261 [24:30<31:28, 2.66s/it]

The times for each task are: [0.626183, 0.900341, 1.668744, 0.424174, 0.393845] with:

Minimum: 0.393845 Maximum: 1.668744 Average: 0.8027 seconds

Number of cars 1

bbox:car_number ((896, 400), (991, 472)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (989, 459))]

Length of task list: 5

Number of processes used: 3

44% | 553/1261 [24:33<31:25, 2.66s/it]

The times for each task are: [0.626158, 0.890933, 1.423114, 0.334399, 0.537106] with:

Minimum: 0.334399 Maximum: 1.423114 Average: 0.7623 seconds

Number of cars 2

bbox:car_number ((913, 400), (987, 472)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((989, 400), (991, 459)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (989, 459))]

Length of task list: 5

Number of processes used: 3

44% | 554/1261 [24:35<31:23, 2.66s/it]

The times for each task are: [1.483218, 0.622735, 1.014763, 0.385557, 0.462428] with:

Minimum: 0.385557 Maximum: 1.483218 Average: 0.7937 seconds

Number of cars 1

bbox:car_number ((932, 400), (987, 462)) : 1

The minimum distance from car: 1 is 10.295630140987

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 555/1261 [24:38<31:20, 2.66s/it]

The times for each task are: [0.628779, 0.901379, 1.617147, 0.294284, 0.390739] with:

Minimum: 0.294284 Maximum: 1.617147 Average: 0.7665 seconds

Number of cars 1

bbox:car_number ((896, 400), (977, 472)) : 1

The minimum distance from car: 1 is 23.53720459187964

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 556/1261 [24:40<31:17, 2.66s/it]

The times for each task are: [0.623138, 0.926809, 1.640238, 0.29817, 0.440665] with:

Minimum: 0.29817 Maximum: 1.640238 Average: 0.7858 seconds

Number of cars 2

bbox:car_number ((913, 400), (991, 472)) : 1

The minimum distance from car: 1 is 16.0

bbox:car_number ((994, 400), (994, 472)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

```
44%|      | 557/1261 [24:43<31:15,  2.66s/it]
```

The times for each task are: [0.910244, 0.682861, 1.620086, 0.409146, 0.340263] with:

Minimum: 0.340263 Maximum: 1.620086 Average: 0.7925 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (987, 462)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
44%|      | 558/1261 [24:46<31:12,  2.66s/it]
```

The times for each task are: [0.628723, 0.883672, 1.668101, 0.509426, 0.308237] with:

Minimum: 0.308237 Maximum: 1.668101 Average: 0.7996 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (991, 462)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
44%|      | 559/1261 [24:49<31:09,  2.66s/it]
```

The times for each task are: [0.705629, 0.816765, 1.560999, 0.336385, 0.476344] with:

Minimum: 0.336385 Maximum: 1.560999 Average: 0.7792 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (1001, 472)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

44% | 560/1261 [24:51<31:07, 2.66s/it]

The times for each task are: [0.91476, 0.680061, 1.583561, 0.402814, 0.374802] with:

Minimum: 0.374802 Maximum: 1.583561 Average: 0.7912 seconds

Number of cars 1

bbox:car_number ((929, 400), (987, 459)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 561/1261 [24:54<31:04, 2.66s/it]

The times for each task are: [0.61139, 1.446136, 0.991259, 0.323625, 0.441221] with:

Minimum: 0.323625 Maximum: 1.446136 Average: 0.7627 seconds

Number of cars 1

bbox:car_number ((913, 400), (1001, 459)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

45% | 562/1261 [24:57<31:02, 2.66s/it]

The times for each task are: [0.89403, 0.660329, 1.467727, 0.31398, 0.397264] with:

Minimum: 0.31398 Maximum: 1.467727 Average: 0.7467 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

45% | 563/1261 [25:00<30:59, 2.66s/it]

The times for each task are: [1.408076, 0.838959, 0.613449, 0.303117, 0.376325] with:

Minimum: 0.303117 Maximum: 1.408076 Average: 0.708 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (994, 472)) : 1
The minimum distance from car: 1 is 13.892443989449804
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (998, 477))]
Length of task list: 5
Number of processes used: 3
```

45% | 564/1261 [25:02<30:57, 2.66s/it]

The times for each task are: [0.626004, 1.035614, 1.523448, 0.354354, 0.383644] with:

Minimum: 0.354354 Maximum: 1.523448 Average: 0.7846 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (998, 477)) : 1
The minimum distance from car: 1 is 2.8284271247461903
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (998, 477))]
Length of task list: 5
Number of processes used: 3
```

45% | 565/1261 [25:05<30:54, 2.66s/it]

The times for each task are: [0.936599, 0.632967, 1.648583, 0.317771, 0.430337] with:

Minimum: 0.317771 Maximum: 1.648583 Average: 0.7933 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (998, 478)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (998, 478))]
Length of task list: 5
Number of processes used: 3
```

45% | 566/1261 [25:08<30:51, 2.66s/it]

The times for each task are: [0.891104, 0.709074, 0.437268, 1.528659, 0.281906] with:

Minimum: 0.281906 Maximum: 1.528659 Average: 0.7696 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (998, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

45%| 567/1261 [25:10<30:49, 2.66s/it]

The times for each task are: [1.470737, 0.71352, 0.900544, 0.348508, 0.397321] with:

Minimum: 0.348508 Maximum: 1.470737 Average: 0.7661 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

45%| 568/1261 [25:13<30:46, 2.66s/it]

The times for each task are: [0.584729, 1.372845, 1.070516, 0.344779, 0.425495] with:

Minimum: 0.344779 Maximum: 1.372845 Average: 0.7597 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 478)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

45%| 569/1261 [25:15<30:43, 2.66s/it]

The times for each task are: [0.63449, 0.911721, 0.41697, 1.711779, 0.291136] with:

Minimum: 0.291136 Maximum: 1.711779 Average: 0.7932 seconds

Number of cars 1

```
bbox:car_number ((913, 400), (994, 478)) : 1
```

```
The minimum distance from car: 1 is 0.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
45%|      | 570/1261 [25:18<30:41, 2.66s/it]
```

```
The times for each task are: [0.877199, 0.410547, 0.618618, 1.640634, 0.355747] with:
```

```
Minimum: 0.355747 Maximum: 1.640634 Average: 0.7805 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((913, 400), (994, 478)) : 1
```

```
The minimum distance from car: 1 is 0.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
45%|      | 571/1261 [25:21<30:38, 2.66s/it]
```

```
The times for each task are: [0.632631, 0.891789, 0.297901, 1.649857, 0.410244] with:
```

```
Minimum: 0.297901 Maximum: 1.649857 Average: 0.7765 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((905, 400), (991, 478)) : 1
```

```
The minimum distance from car: 1 is 5.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
45%|      | 572/1261 [25:23<30:35, 2.66s/it]
```

```
The times for each task are: [1.399565, 0.896445, 0.632724, 0.291511, 0.51326] with:
```

```
Minimum: 0.291511 Maximum: 1.399565 Average: 0.7467 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((913, 400), (994, 478)) : 1
```

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

45% | 573/1261 [25:26<30:32, 2.66s/it]

The times for each task are: [0.892088, 1.375014, 0.395326, 0.677163, 0.400794] with:

Minimum: 0.395326 Maximum: 1.375014 Average: 0.7481 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

46% | 574/1261 [25:29<30:30, 2.66s/it]

The times for each task are: [0.882291, 0.581591, 1.415135, 0.455125, 0.34314] with:

Minimum: 0.34314 Maximum: 1.415135 Average: 0.7355 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

46% | 575/1261 [25:31<30:27, 2.66s/it]

The times for each task are: [0.897159, 0.589379, 1.434647, 0.510216, 0.308952] with:

Minimum: 0.308952 Maximum: 1.434647 Average: 0.7481 seconds

Number of cars 1

bbox:car_number ((913, 400), (991, 472)) : 1

The minimum distance from car: 1 is 5.0

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (994, 478))]
Length of task list: 5
Number of processes used: 3
```

46%| 576/1261 [25:34<30:24, 2.66s/it]

The times for each task are: [0.88488, 0.705875, 0.412395, 1.390271, 0.38463] with:

Minimum: 0.38463 Maximum: 1.390271 Average: 0.7556 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (994, 478)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (994, 478))]
Length of task list: 5
Number of processes used: 3
```

46%| 577/1261 [25:37<30:22, 2.66s/it]

The times for each task are: [0.947592, 0.65099, 1.499323, 0.297475, 0.516791] with:

Minimum: 0.297475 Maximum: 1.499323 Average: 0.7824 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (994, 478))]
Length of task list: 5
Number of processes used: 3
```

46%| 578/1261 [25:40<30:19, 2.66s/it]

The times for each task are: [0.674804, 1.005203, 0.469787, 1.500869, 0.378254] with:

Minimum: 0.378254 Maximum: 1.500869 Average: 0.8058 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (991, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 579/1261 [25:42<30:17, 2.66s/it]
```

```
The times for each task are: [0.877424, 0.590955, 1.612922, 0.325822, 0.402857] with:
```

```
Minimum: 0.325822 Maximum: 1.612922 Average: 0.762 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400), (991, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 580/1261 [25:45<30:14, 2.66s/it]
```

```
The times for each task are: [0.944509, 1.372901, 0.69336, 0.364001, 0.439357] with:
```

```
Minimum: 0.364001 Maximum: 1.372901 Average: 0.7628 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400), (991, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 581/1261 [25:47<30:11, 2.66s/it]
```

```
The times for each task are: [0.918996, 1.391351, 0.618493, 0.361412, 0.40752] with:
```

```
Minimum: 0.361412 Maximum: 1.391351 Average: 0.7396 seconds
```

```
Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400), (991, 478)) : 1
```

```
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 582/1261 [25:50<30:08, 2.66s/it]
```

```
The times for each task are: [0.901342, 0.651373, 0.420869, 1.607463, 0.29356] with:
```

```
Minimum: 0.29356 Maximum: 1.607463 Average: 0.7749 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 583/1261 [25:52<30:05, 2.66s/it]
```

```
The times for each task are: [0.880574, 1.349811, 0.684009, 0.300555, 0.384966] with:
```

```
Minimum: 0.300555 Maximum: 1.349811 Average: 0.72 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 584/1261 [25:55<30:03, 2.66s/it]
```

```
The times for each task are: [0.88639, 0.406345, 0.764336, 1.525414, 0.353641] with:
```

```
Minimum: 0.353641 Maximum: 1.525414 Average: 0.7872 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

46% | 585/1261 [25:58<30:00, 2.66s/it]

The times for each task are: [1.093408, 0.670163, 1.553527, 0.320015, 0.390733] with:

Minimum: 0.320015 Maximum: 1.553527 Average: 0.8056 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

46% | 586/1261 [26:01<29:58, 2.66s/it]

The times for each task are: [0.599103, 0.961901, 1.663574, 0.377834, 0.476845] with:

Minimum: 0.377834 Maximum: 1.663574 Average: 0.8159 seconds

Number of cars 1

bbox:car_number ((910, 400), (991, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 587/1261 [26:04<29:55, 2.66s/it]

The times for each task are: [0.630453, 0.86722, 1.441309, 0.286613, 0.37991] with:

Minimum: 0.286613 Maximum: 1.441309 Average: 0.7211 seconds

Number of cars 1

bbox:car_number ((913, 400), (991, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 588/1261 [26:07<29:53, 2.67s/it]

The times for each task are: [0.928596, 0.732382, 1.696175, 0.361094, 0.46276] with:

Minimum: 0.361094 Maximum: 1.696175 Average: 0.8362 seconds

Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

47% | 589/1261 [26:09<29:51, 2.67s/it]

The times for each task are: [0.593831, 1.459103, 1.034933, 0.412055, 0.423255] with:

Minimum: 0.412055 Maximum: 1.459103 Average: 0.7846 seconds

Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

47% | 590/1261 [26:12<29:48, 2.67s/it]

The times for each task are: [0.976091, 0.606203, 1.40986, 0.298555, 0.40051] with:

Minimum: 0.298555 Maximum: 1.40986 Average: 0.7382 seconds

Number of cars 1
bbox:car_number ((932, 400), (991, 472)) : 1
The minimum distance from car: 1 is 13.341664064126334
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

47% | 591/1261 [26:15<29:45, 2.67s/it]

The times for each task are: [0.56098, 0.952118, 1.47125, 0.381419, 0.289295] with:

Minimum: 0.289295 Maximum: 1.47125 Average: 0.731 seconds

Number of cars 1

bbox:car_number ((913, 400), (991, 472)) : 1

The minimum distance from car: 1 is 9.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

47% | 592/1261 [26:17<29:43, 2.67s/it]

The times for each task are: [1.005718, 0.617552, 1.39467, 0.308232, 0.443273] with:

Minimum: 0.308232 Maximum: 1.39467 Average: 0.7539 seconds

Number of cars 1

bbox:car_number ((913, 400), (991, 477)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

47% | 593/1261 [26:20<29:40, 2.67s/it]

The times for each task are: [0.911448, 0.709745, 1.512358, 0.403403, 0.348147] with:

Minimum: 0.348147 Maximum: 1.512358 Average: 0.777 seconds

Number of cars 1

bbox:car_number ((905, 400), (994, 485)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

47% | 594/1261 [26:23<29:38, 2.67s/it]

The times for each task are: [0.580925, 0.893456, 1.415064, 0.313283, 0.407061] with:

Minimum: 0.313283 Maximum: 1.415064 Average: 0.722 seconds

Number of cars 2

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 3.1622776601683795

bbox:car_number ((896, 419), (896, 472)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 595/1261 [26:26<29:35, 2.67s/it]

The times for each task are: [0.844463, 0.616018, 1.703132, 0.315016, 0.471181] with:

Minimum: 0.315016 Maximum: 1.703132 Average: 0.79 seconds

Number of cars 1

bbox:car_number ((910, 400), (991, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 596/1261 [26:28<29:32, 2.67s/it]

The times for each task are: [0.87405, 0.70281, 0.453564, 1.645974, 0.362842] with:

Minimum: 0.362842 Maximum: 1.645974 Average: 0.8078 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 597/1261 [26:31<29:30, 2.67s/it]

The times for each task are: [0.902312, 0.698321, 0.419832, 1.66338, 0.3019] with:

Minimum: 0.3019 Maximum: 1.66338 Average: 0.7971 seconds

Number of cars 2

bbox:car_number ((886, 400), (896, 472)) : 1

bbox:car_number ((905, 400), (991, 478)) : 2

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 598/1261 [26:34<29:27, 2.67s/it]

The times for each task are: [0.934973, 0.653321, 1.44855, 0.410178, 0.289956] with:

Minimum: 0.289956 Maximum: 1.44855 Average: 0.7474 seconds

Number of cars 1

bbox:car_number ((882, 400), (991, 485)) : 1

The minimum distance from car: 1 is 12.36931687685298

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

48% | 599/1261 [26:37<29:25, 2.67s/it]

The times for each task are: [0.634742, 0.923189, 0.459429, 1.423617, 0.287636] with:

Minimum: 0.287636 Maximum: 1.423617 Average: 0.7457 seconds

Number of cars 1

bbox:car_number ((896, 400), (991, 485)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

48% | 600/1261 [26:40<29:23, 2.67s/it]

The times for each task are: [0.701982, 0.995701, 1.4291, 0.296713, 0.439531] with:

Minimum: 0.296713 Maximum: 1.4291 Average: 0.7726 seconds

Number of cars 1

bbox:car_number ((886, 400), (994, 495)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

48% | 601/1261 [26:42<29:20, 2.67s/it]

The times for each task are: [1.004909, 1.367342, 0.689652, 0.291149, 0.459263] with:

Minimum: 0.291149 Maximum: 1.367342 Average: 0.7625 seconds

Number of cars 1

bbox:car_number ((896, 400), (991, 485)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

48% | 602/1261 [26:45<29:17, 2.67s/it]

The times for each task are: [0.887385, 0.603141, 1.468882, 0.302622, 0.393636] with:

Minimum: 0.302622 Maximum: 1.468882 Average: 0.7311 seconds

Number of cars 1

bbox:car_number ((886, 400), (991, 485)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

48% | 603/1261 [26:48<29:14, 2.67s/it]

The times for each task are: [0.657151, 0.962474, 1.573228, 0.329669, 0.475615] with:

Minimum: 0.329669 Maximum: 1.573228 Average: 0.7996 seconds

Number of cars 1
bbox:car_number ((894, 400), (994, 485)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

48% | 604/1261 [26:50<29:12, 2.67s/it]

The times for each task are: [0.962315, 0.674675, 1.445998, 0.313229, 0.382036] with:

Minimum: 0.313229 Maximum: 1.445998 Average: 0.7557 seconds

Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

48% | 605/1261 [26:53<29:09, 2.67s/it]

The times for each task are: [0.619491, 0.894455, 0.318524, 1.643337, 0.413674] with:

Minimum: 0.318524 Maximum: 1.643337 Average: 0.7779 seconds

Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

48% | 606/1261 [26:56<29:06, 2.67s/it]

The times for each task are: [0.876068, 1.423983, 0.704006, 0.458822, 0.27519] with:

Minimum: 0.27519 Maximum: 1.423983 Average: 0.7476 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

48% | 607/1261 [26:58<29:04, 2.67s/it]

The times for each task are: [0.662835, 1.392112, 0.860198, 0.437263, 0.284831] with:

Minimum: 0.284831 Maximum: 1.392112 Average: 0.7274 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 485)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

48% | 608/1261 [27:01<29:01, 2.67s/it]

The times for each task are: [0.87283, 0.713472, 1.438984, 0.467179, 0.353294] with:

Minimum: 0.353294 Maximum: 1.438984 Average: 0.7692 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

48% | 609/1261 [27:04<28:58, 2.67s/it]

The times for each task are: [0.642135, 0.913436, 0.406321, 1.455215, 0.292756] with:

Minimum: 0.292756 Maximum: 1.455215 Average: 0.742 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 477)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

48%| 610/1261 [27:06<28:56, 2.67s/it]

The times for each task are: [0.906029, 0.593606, 1.46895, 0.397409, 0.280386] with:
Minimum: 0.280386 Maximum: 1.46895 Average: 0.7293 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

48%| 611/1261 [27:09<28:53, 2.67s/it]

The times for each task are: [0.618219, 0.930281, 0.404903, 1.661878, 0.287533] with:
Minimum: 0.287533 Maximum: 1.661878 Average: 0.7806 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

49%| 612/1261 [27:12<28:50, 2.67s/it]

The times for each task are: [0.8529, 0.57365, 0.300777, 1.530996, 0.396016] with:
Minimum: 0.300777 Maximum: 1.530996 Average: 0.7309 seconds

Number of cars 1

```
bbox:car_number ((910, 400), (987, 478)) : 1
```

```
The minimum distance from car: 1 is 2.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
49% | 613/1261 [27:14<28:48, 2.67s/it]
```

```
The times for each task are: [1.002703, 1.393108, 0.605221, 0.289559, 0.483789] with:
```

```
Minimum: 0.289559 Maximum: 1.393108 Average: 0.7549 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((910, 400), (987, 478)) : 1
```

```
The minimum distance from car: 1 is 0.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
49% | 614/1261 [27:17<28:45, 2.67s/it]
```

```
The times for each task are: [0.825163, 0.611112, 1.403089, 0.491728, 0.343713] with:
```

```
Minimum: 0.343713 Maximum: 1.403089 Average: 0.735 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((905, 400), (987, 478)) : 1
```

```
The minimum distance from car: 1 is 2.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
49% | 615/1261 [27:20<28:43, 2.67s/it]
```

```
The times for each task are: [1.347113, 0.631049, 1.017488, 0.400859, 0.365266] with:
```

```
Minimum: 0.365266 Maximum: 1.347113 Average: 0.7524 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((905, 400), (987, 478)) : 1
```

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]

Length of task list: 5

Number of processes used: 3

49% | 616/1261 [27:22<28:40, 2.67s/it]

The times for each task are: [1.396543, 1.122826, 0.62317, 0.343176, 0.439515] with:

Minimum: 0.343176 Maximum: 1.396543 Average: 0.785 seconds

Number of cars 1

bbox:car_number ((905, 400), (987, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]

Length of task list: 5

Number of processes used: 3

49% | 617/1261 [27:25<28:37, 2.67s/it]

The times for each task are: [0.926996, 1.6429, 0.680181, 0.400039, 0.350494] with:

Minimum: 0.350494 Maximum: 1.6429 Average: 0.8001 seconds

Number of cars 1

bbox:car_number ((905, 400), (987, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]

Length of task list: 5

Number of processes used: 3

49% | 618/1261 [27:28<28:35, 2.67s/it]

The times for each task are: [0.657208, 0.880735, 1.68173, 0.428913, 0.295101] with:

Minimum: 0.295101 Maximum: 1.68173 Average: 0.7887 seconds

Number of cars 1

bbox:car_number ((905, 400), (987, 478)) : 1

The minimum distance from car: 1 is 0.0

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

49% | 619/1261 [27:30<28:32, 2.67s/it]

The times for each task are: [0.901462, 0.618118, 1.459223, 0.318908, 0.49981] with:
Minimum: 0.318908 Maximum: 1.459223 Average: 0.7595 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (987, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

49% | 620/1261 [27:33<28:29, 2.67s/it]

The times for each task are: [0.868918, 0.606769, 1.426586, 0.420794, 0.288434] with:
Minimum: 0.288434 Maximum: 1.426586 Average: 0.7223 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

49% | 621/1261 [27:36<28:27, 2.67s/it]

The times for each task are: [0.745152, 0.8547, 1.532697, 0.347901, 0.4779] with:
Minimum: 0.347901 Maximum: 1.532697 Average: 0.7917 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (977, 478)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400), (977, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

```
49%| 622/1261 [27:39<28:24, 2.67s/it]
```

```
The times for each task are: [1.478777, 0.965784, 0.632394, 0.30209, 0.389975] with:
```

```
Minimum: 0.30209 Maximum: 1.478777 Average: 0.7538 seconds
```

```
Number of cars 1
bbox:car_number ((905, 400), (977, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (977, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

```
49%| 623/1261 [27:41<28:21, 2.67s/it]
```

```
The times for each task are: [0.934883, 0.698447, 1.663087, 0.380668, 0.290607] with:
```

```
Minimum: 0.290607 Maximum: 1.663087 Average: 0.7935 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (977, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (977, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

```
49%| 624/1261 [27:44<28:19, 2.67s/it]
```

```
The times for each task are: [0.719323, 0.892561, 1.366523, 0.296924, 0.465304] with:
```

```
Minimum: 0.296924 Maximum: 1.366523 Average: 0.7481 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (977, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (977, 478)) : 1
```

```
Length of task list: 5
Number of processes used: 3
```

```
50%|      | 625/1261 [27:47<28:16, 2.67s/it]
```

```
The times for each task are: [0.573384, 0.935276, 1.426846, 0.439309, 0.283561] with:
```

```
Minimum: 0.283561 Maximum: 1.426846 Average: 0.7317 seconds
```

```
Number of cars 1
bbox:car_number ((910, 415), (972, 478)) : 1
The minimum distance from car: 1 is 7.280109889280518
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

```
50%|      | 626/1261 [27:49<28:13, 2.67s/it]
```

```
The times for each task are: [0.871789, 0.698982, 1.602988, 0.33824, 0.467925] with:
```

```
Minimum: 0.33824 Maximum: 1.602988 Average: 0.796 seconds
```

```
Number of cars 1
bbox:car_number ((910, 415), (977, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

```
50%|      | 627/1261 [27:52<28:11, 2.67s/it]
```

```
The times for each task are: [0.895538, 0.74283, 1.525864, 0.385228, 0.368044] with:
```

```
Minimum: 0.368044 Maximum: 1.525864 Average: 0.7835 seconds
```

```
Number of cars 1
bbox:car_number ((905, 400), (987, 478)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
```

Number of processes used: 3

50% | 628/1261 [27:55<28:08, 2.67s/it]

The times for each task are: [0.61124, 0.849871, 1.56555, 0.424024, 0.279752] with:

Minimum: 0.279752 Maximum: 1.56555 Average: 0.7461 seconds

Number of cars 1

bbox:car_number ((910, 415), (972, 477)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

50% | 629/1261 [27:58<28:06, 2.67s/it]

The times for each task are: [1.389024, 0.909566, 0.715428, 0.413571, 0.327595] with:

Minimum: 0.327595 Maximum: 1.389024 Average: 0.751 seconds

Number of cars 1

bbox:car_number ((910, 400), (972, 477)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

50% | 630/1261 [28:01<28:03, 2.67s/it]

The times for each task are: [0.919403, 0.625576, 1.454379, 0.328978, 0.448297] with:

Minimum: 0.328978 Maximum: 1.454379 Average: 0.7553 seconds

Number of cars 1

bbox:car_number ((896, 400), (972, 472)) : 1

The minimum distance from car: 1 is 7.280109889280518

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

50%| 631/1261 [28:03<28:01, 2.67s/it]

The times for each task are: [1.106615, 0.653166, 1.424531, 0.297737, 0.448884] with:

Minimum: 0.297737 Maximum: 1.424531 Average: 0.7862 seconds

Number of cars 1
bbox:car_number ((896, 400), (972, 472)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

50%| 632/1261 [28:06<27:58, 2.67s/it]

The times for each task are: [0.967922, 0.675899, 1.385299, 0.472332, 0.409918] with:

Minimum: 0.409918 Maximum: 1.385299 Average: 0.7823 seconds

Number of cars 1
bbox:car_number ((913, 400), (972, 462)) : 1
The minimum distance from car: 1 is 9.433981132056603
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

50%| 633/1261 [28:09<27:56, 2.67s/it]

The times for each task are: [0.958299, 0.684022, 0.400264, 1.359502, 0.39021] with:

Minimum: 0.39021 Maximum: 1.359502 Average: 0.7585 seconds

Number of cars 1
bbox:car_number ((896, 400), (972, 478)) : 1
The minimum distance from car: 1 is 11.313708498984761
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

50% | 634/1261 [28:11<27:53, 2.67s/it]

The times for each task are: [1.01932, 1.435757, 0.644599, 0.295504, 0.43882] with:

Minimum: 0.295504 Maximum: 1.435757 Average: 0.7668 seconds

Number of cars 1

bbox:car_number ((895, 400), (972, 477)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

50% | 635/1261 [28:14<27:50, 2.67s/it]

The times for each task are: [0.618006, 0.883547, 0.313641, 0.397753, 1.413715] with:

Minimum: 0.313641 Maximum: 1.413715 Average: 0.7253 seconds

Number of cars 1

bbox:car_number ((894, 400), (972, 478)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

50% | 636/1261 [28:17<27:47, 2.67s/it]

The times for each task are: [0.879424, 0.690507, 1.737507, 0.342169, 0.4163] with:

Minimum: 0.342169 Maximum: 1.737507 Average: 0.8132 seconds

Number of cars 1

bbox:car_number ((895, 415), (972, 477)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

51% | 637/1261 [28:19<27:45, 2.67s/it]

The times for each task are: [0.845709, 1.418256, 0.636685, 0.449054, 0.300102] with:

Minimum: 0.300102 Maximum: 1.418256 Average: 0.73 seconds

Number of cars 1

bbox:car_number ((895, 400), (972, 492)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

51% | 638/1261 [28:22<27:42, 2.67s/it]

The times for each task are: [0.882315, 0.666046, 1.437815, 0.31311, 0.52122] with:

Minimum: 0.31311 Maximum: 1.437815 Average: 0.7641 seconds

Number of cars 1

bbox:car_number ((886, 400), (972, 495)) : 1

The minimum distance from car: 1 is 4.123105625617661

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

51% | 639/1261 [28:25<27:40, 2.67s/it]

The times for each task are: [0.903962, 1.607184, 0.672149, 0.435289, 0.300322] with:

Minimum: 0.300322 Maximum: 1.607184 Average: 0.7838 seconds

Number of cars 1

bbox:car_number ((886, 415), (972, 485)) : 1

The minimum distance from car: 1 is 3.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

51% | 640/1261 [28:28<27:37, 2.67s/it]

The times for each task are: [0.90128, 0.711635, 1.644088, 0.395377, 0.346593] with:

Minimum: 0.346593 Maximum: 1.644088 Average: 0.7998 seconds

Number of cars 1
bbox:car_number ((894, 400), (972, 478)) : 1
The minimum distance from car: 1 is 11.704699910719626
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

51% | 641/1261 [28:30<27:34, 2.67s/it]

The times for each task are: [0.619406, 0.896091, 0.348234, 0.41592, 1.669607] with:

Minimum: 0.348234 Maximum: 1.669607 Average: 0.7899 seconds

Number of cars 1
bbox:car_number ((894, 400), (972, 477)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

51% | 642/1261 [28:33<27:31, 2.67s/it]

The times for each task are: [1.326348, 0.611414, 1.041364, 0.284942, 0.462432] with:

Minimum: 0.284942 Maximum: 1.326348 Average: 0.7453 seconds

Number of cars 1
bbox:car_number ((895, 400), (972, 472)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

51% | 643/1261 [28:36<27:29, 2.67s/it]

The times for each task are: [0.982495, 1.366588, 0.609826, 0.30058, 0.489394] with:

Minimum: 0.30058 Maximum: 1.366588 Average: 0.7498 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (972, 477)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (957, 477))]
Length of task list: 5
Number of processes used: 3
```

51% | 644/1261 [28:38<27:26, 2.67s/it]

The times for each task are: [0.879478, 1.48128, 0.711496, 0.393121, 0.302316] with:

Minimum: 0.302316 Maximum: 1.48128 Average: 0.7535 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (971, 477)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (957, 477))]
Length of task list: 5
Number of processes used: 3
```

51% | 645/1261 [28:41<27:24, 2.67s/it]

The times for each task are: [0.57153, 1.47949, 1.038443, 0.447409, 0.321606] with:

Minimum: 0.321606 Maximum: 1.47949 Average: 0.7717 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (957, 477)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (957, 477))]
Length of task list: 5
Number of processes used: 3
```

51% | 646/1261 [28:44<27:21, 2.67s/it]

The times for each task are: [0.954506, 1.404737, 0.730103, 0.298141, 0.394929] with:

Minimum: 0.298141 Maximum: 1.404737 Average: 0.7565 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (963, 478)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (963, 478))]
Length of task list: 5
Number of processes used: 3
```

51%| 647/1261 [28:46<27:18, 2.67s/it]

The times for each task are: [0.934957, 1.376534, 0.734379, 0.303073, 0.47879] with:
Minimum: 0.303073 Maximum: 1.376534 Average: 0.7655 seconds

```
Number of cars 1
bbox:car_number ((886, 400), (963, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (963, 478))]
Length of task list: 5
Number of processes used: 3
```

51%| 648/1261 [28:49<27:16, 2.67s/it]

The times for each task are: [0.903779, 0.76431, 1.450486, 0.344181, 0.394177] with:
Minimum: 0.344181 Maximum: 1.450486 Average: 0.7714 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (957, 478)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (957, 478))]
Length of task list: 5
Number of processes used: 3
```

51%| 649/1261 [28:52<27:13, 2.67s/it]

The times for each task are: [1.39012, 0.85341, 0.763398, 0.446859, 0.308592] with:
Minimum: 0.308592 Maximum: 1.39012 Average: 0.7525 seconds

Number of cars 1

```
bbox:car_number ((886, 400), (957, 478)) : 1
```

```
The minimum distance from car: 1 is 4.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

52% | 650/1261 [28:54<27:10, 2.67s/it]

The times for each task are: [0.843392, 0.739224, 1.441552, 0.482678, 0.333202] with:

Minimum: 0.333202 Maximum: 1.441552 Average: 0.768 seconds

```
Number of cars 1
```

```
bbox:car_number ((894, 415), (954, 472)) : 1
```

```
The minimum distance from car: 1 is 5.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

52% | 651/1261 [28:57<27:08, 2.67s/it]

The times for each task are: [0.641157, 0.997492, 1.448095, 0.444767, 0.357302] with:

Minimum: 0.357302 Maximum: 1.448095 Average: 0.7778 seconds

```
Number of cars 1
```

```
bbox:car_number ((894, 400), (957, 472)) : 1
```

```
The minimum distance from car: 1 is 7.0710678118654755
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

52% | 652/1261 [29:00<27:05, 2.67s/it]

The times for each task are: [0.979956, 0.630882, 1.650318, 0.305882, 0.468977] with:

Minimum: 0.305882 Maximum: 1.650318 Average: 0.8072 seconds

```
Number of cars 1
```

```
bbox:car_number ((894, 400), (954, 472)) : 1
```

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

52% | 653/1261 [29:02<27:02, 2.67s/it]

The times for each task are: [0.731504, 0.850286, 1.435368, 0.471889, 0.297115] with:

Minimum: 0.297115 Maximum: 1.435368 Average: 0.7572 seconds

Number of cars 1

bbox:car_number ((886, 400), (954, 472)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

52% | 654/1261 [29:05<27:00, 2.67s/it]

The times for each task are: [0.878986, 0.729084, 1.353487, 0.500882, 0.296645] with:

Minimum: 0.296645 Maximum: 1.353487 Average: 0.7518 seconds

Number of cars 1

bbox:car_number ((894, 415), (953, 462)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

52% | 655/1261 [29:08<26:57, 2.67s/it]

The times for each task are: [0.631732, 1.507198, 1.052637, 0.415602, 0.378696] with:

Minimum: 0.378696 Maximum: 1.507198 Average: 0.7972 seconds

Number of cars 1

bbox:car_number ((894, 415), (957, 477)) : 1

The minimum distance from car: 1 is 8.246211251235321

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

52%| 656/1261 [29:11<26:55, 2.67s/it]

The times for each task are: [0.610472, 0.917566, 0.434924, 1.49919, 0.370593] with:
Minimum: 0.370593 Maximum: 1.49919 Average: 0.7665 seconds

```
Number of cars 1
bbox:car_number ((886, 400), (954, 472)) : 1
The minimum distance from car: 1 is 11.180339887498949
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

52%| 657/1261 [29:14<26:52, 2.67s/it]

The times for each task are: [0.955492, 0.747222, 1.452633, 0.409522, 0.297339] with:
Minimum: 0.297339 Maximum: 1.452633 Average: 0.7724 seconds

```
Number of cars 1
bbox:car_number ((882, 400), (954, 477)) : 1
The minimum distance from car: 1 is 2.8284271247461903
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

52%| 658/1261 [29:17<26:50, 2.67s/it]

The times for each task are: [1.168482, 1.521142, 0.663764, 0.31221, 0.464304] with:
Minimum: 0.31221 Maximum: 1.521142 Average: 0.826 seconds

```
Number of cars 1
bbox:car_number ((880, 400), (957, 477)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

52% | 659/1261 [29:19<26:47, 2.67s/it]

The times for each task are: [0.646494, 0.936497, 1.491554, 0.421412, 0.310447] with:

Minimum: 0.310447 Maximum: 1.491554 Average: 0.7613 seconds

```
Number of cars 1  
bbox:car_number ((875, 400), (957, 478)) : 1  
The minimum distance from car: 1 is 2.23606797749979  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

52% | 660/1261 [29:22<26:44, 2.67s/it]

The times for each task are: [0.851577, 1.408657, 0.685039, 0.401669, 0.285724] with:

Minimum: 0.285724 Maximum: 1.408657 Average: 0.7265 seconds

```
Number of cars 1  
bbox:car_number ((875, 400), (957, 478)) : 1  
The minimum distance from car: 1 is 0.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

52% | 661/1261 [29:24<26:41, 2.67s/it]

The times for each task are: [0.950401, 1.373619, 0.657675, 0.396279, 0.351218] with:

Minimum: 0.351218 Maximum: 1.373619 Average: 0.7458 seconds

```
Number of cars 1  
bbox:car_number ((880, 400), (957, 478)) : 1  
The minimum distance from car: 1 is 2.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
52%|     | 662/1261 [29:27<26:39,  2.67s/it]
```

```
The times for each task are: [0.615271, 0.968835, 1.612489, 0.401953, 0.302667] with:
```

```
Minimum: 0.302667 Maximum: 1.612489 Average: 0.7802 seconds
```

```
Number of cars 1
bbox:car_number ((880, 400), (957, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
53%|     | 663/1261 [29:29<26:36,  2.67s/it]
```

```
The times for each task are: [0.924305, 0.421444, 0.70182, 1.472315, 0.334703] with:
```

```
Minimum: 0.334703 Maximum: 1.472315 Average: 0.7709 seconds
```

```
Number of cars 1
bbox:car_number ((894, 400), (957, 477)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
53%|     | 664/1261 [29:32<26:33,  2.67s/it]
```

```
The times for each task are: [0.881465, 0.68056, 1.446837, 0.298562, 0.401591] with:
```

```
Minimum: 0.298562 Maximum: 1.446837 Average: 0.7418 seconds
```

```
Number of cars 1
bbox:car_number ((882, 400), (954, 477)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

53% | 665/1261 [29:34<26:30, 2.67s/it]

The times for each task are: [0.880422, 0.644576, 0.30953, 1.662231, 0.386832] with:

Minimum: 0.30953 Maximum: 1.662231 Average: 0.7767 seconds

Number of cars 1

bbox:car_number ((880, 400), (954, 478)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53% | 666/1261 [29:37<26:27, 2.67s/it]

The times for each task are: [0.90524, 1.420729, 0.750377, 0.43551, 0.317066] with:

Minimum: 0.317066 Maximum: 1.420729 Average: 0.7658 seconds

Number of cars 1

bbox:car_number ((880, 400), (954, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53% | 667/1261 [29:40<26:25, 2.67s/it]

The times for each task are: [1.445031, 0.939927, 0.782687, 0.479188, 0.348863] with:

Minimum: 0.348863 Maximum: 1.445031 Average: 0.7991 seconds

Number of cars 1

bbox:car_number ((880, 400), (954, 477)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53%| 668/1261 [29:42<26:22, 2.67s/it]

The times for each task are: [1.378202, 0.82024, 0.713085, 0.396389, 0.329867] with:

Minimum: 0.329867 Maximum: 1.378202 Average: 0.7276 seconds

Number of cars 1

bbox:car_number ((880, 400), (953, 478)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53%| 669/1261 [29:45<26:20, 2.67s/it]

The times for each task are: [0.946934, 0.62867, 0.31467, 1.694324, 0.398251] with:

Minimum: 0.31467 Maximum: 1.694324 Average: 0.7966 seconds

Number of cars 1

bbox:car_number ((875, 400), (953, 472)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53%| 670/1261 [29:48<26:17, 2.67s/it]

The times for each task are: [0.593585, 0.913326, 1.398411, 0.419601, 0.319105] with:

Minimum: 0.319105 Maximum: 1.398411 Average: 0.7288 seconds

Number of cars 1

bbox:car_number ((880, 400), (944, 472)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53% | 671/1261 [29:50<26:14, 2.67s/it]

The times for each task are: [0.574718, 0.860378, 1.660343, 0.404587, 0.306626] with:

Minimum: 0.306626 Maximum: 1.660343 Average: 0.7613 seconds

Number of cars 1

bbox:car_number ((875, 400), (953, 477)) : 1

The minimum distance from car: 1 is 2.8284271247461903

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

53% | 672/1261 [29:53<26:11, 2.67s/it]

The times for each task are: [1.056898, 0.622558, 1.399687, 0.471197, 0.309561] with:

Minimum: 0.309561 Maximum: 1.399687 Average: 0.772 seconds

Number of cars 1

bbox:car_number ((880, 400), (942, 472)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

53% | 673/1261 [29:56<26:09, 2.67s/it]

The times for each task are: [0.866414, 0.730963, 1.42987, 0.385836, 0.291585] with:

Minimum: 0.291585 Maximum: 1.42987 Average: 0.7409 seconds

Number of cars 1

bbox:car_number ((880, 400), (953, 472)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

53% | 674/1261 [29:58<26:06, 2.67s/it]

The times for each task are: [0.631547, 0.936694, 1.472759, 0.453965, 0.308551] with:

Minimum: 0.308551 Maximum: 1.472759 Average: 0.7607 seconds

Number of cars 1

bbox:car_number ((865, 400), (944, 472)) : 1

The minimum distance from car: 1 is 12.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

54% | 675/1261 [30:01<26:03, 2.67s/it]

The times for each task are: [0.736024, 0.913974, 1.447525, 0.299939, 0.476134] with:

Minimum: 0.299939 Maximum: 1.447525 Average: 0.7747 seconds

Number of cars 1

bbox:car_number ((865, 400), (944, 472)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

54% | 676/1261 [30:03<26:01, 2.67s/it]

The times for each task are: [1.500676, 0.691007, 0.922425, 0.455879, 0.366511] with:

Minimum: 0.366511 Maximum: 1.500676 Average: 0.7873 seconds

Number of cars 1

bbox:car_number ((880, 400), (944, 477)) : 1

The minimum distance from car: 1 is 8.246211251235321

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

54% | 677/1261 [30:06<25:58, 2.67s/it]

The times for each task are: [0.626293, 0.87431, 1.602181, 0.304957, 0.454471] with:

Minimum: 0.304957 Maximum: 1.602181 Average: 0.7724 seconds

```
Number of cars 1
bbox:car_number ((865, 400), (944, 477)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (953, 477))]
Length of task list: 5
Number of processes used: 3
```

54% | 678/1261 [30:09<25:55, 2.67s/it]

The times for each task are: [0.872064, 0.615546, 1.636251, 0.299069, 0.438694] with:

Minimum: 0.299069 Maximum: 1.636251 Average: 0.7723 seconds

```
Number of cars 1
bbox:car_number ((865, 400), (953, 477)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (953, 477))]
Length of task list: 5
Number of processes used: 3
```

54% | 679/1261 [30:11<25:52, 2.67s/it]

The times for each task are: [0.552566, 0.925544, 0.391867, 0.307679, 1.697282] with:

Minimum: 0.307679 Maximum: 1.697282 Average: 0.775 seconds

```
Number of cars 1
bbox:car_number ((880, 400), (944, 472)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (944, 472))]
Length of task list: 5
Number of processes used: 3
```

54% | 680/1261 [30:14<25:50, 2.67s/it]

The times for each task are: [0.680206, 1.513785, 1.026313, 0.310058, 0.451148] with:

Minimum: 0.310058 Maximum: 1.513785 Average: 0.7963 seconds

```
Number of cars 1
bbox:car_number ((880, 400), (942, 472)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54% | 681/1261 [30:17<25:47, 2.67s/it]

The times for each task are: [0.879723, 1.441684, 0.656384, 0.423627, 0.288076] with:

Minimum: 0.288076 Maximum: 1.441684 Average: 0.7379 seconds

```
Number of cars 1
bbox:car_number ((880, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54% | 682/1261 [30:19<25:45, 2.67s/it]

The times for each task are: [0.887756, 0.652585, 1.629733, 0.484165, 0.297035] with:

Minimum: 0.297035 Maximum: 1.629733 Average: 0.7903 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (944, 472)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54% | 683/1261 [30:22<25:41, 2.67s/it]

The times for each task are: [0.904205, 0.581334, 0.446149, 1.594691, 0.389718] with:

Minimum: 0.389718 Maximum: 1.594691 Average: 0.7832 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (953, 478)) : 1
The minimum distance from car: 1 is 5.830951894845301
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54%| 684/1261 [30:24<25:39, 2.67s/it]

The times for each task are: [0.616659, 1.415848, 1.056062, 0.299393, 0.393416] with:
Minimum: 0.299393 Maximum: 1.415848 Average: 0.7563 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54%| 685/1261 [30:27<25:36, 2.67s/it]

The times for each task are: [0.992325, 0.625043, 1.383575, 0.300673, 0.404416] with:
Minimum: 0.300673 Maximum: 1.383575 Average: 0.7412 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54%| 686/1261 [30:30<25:34, 2.67s/it]

The times for each task are: [0.614953, 0.889279, 1.52867, 0.421753, 0.295795] with:
Minimum: 0.295795 Maximum: 1.52867 Average: 0.7501 seconds

Number of cars 1

```
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54% | 687/1261 [30:33<25:31, 2.67s/it]

The times for each task are: [0.867097, 1.394013, 0.710725, 0.287224, 0.394239] with:
Minimum: 0.287224 Maximum: 1.394013 Average: 0.7307 seconds

```
Number of cars 1
bbox:car_number ((880, 400), (942, 472)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

55% | 688/1261 [30:36<25:29, 2.67s/it]

The times for each task are: [0.924801, 0.645826, 1.628772, 0.394262, 0.348067] with:
Minimum: 0.348067 Maximum: 1.628772 Average: 0.7883 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

55% | 689/1261 [30:38<25:26, 2.67s/it]

The times for each task are: [0.827106, 0.739553, 1.400625, 0.496423, 0.354989] with:
Minimum: 0.354989 Maximum: 1.400625 Average: 0.7637 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
```

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (875, 400))]

Length of task list: 5

Number of processes used: 3

55% | 690/1261 [30:41<25:23, 2.67s/it]

The times for each task are: [0.579958, 1.030954, 1.527766, 0.463397, 0.41616] with:

Minimum: 0.41616 Maximum: 1.527766 Average: 0.8036 seconds

Number of cars 1

bbox:car_number ((875, 400), (942, 472)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (875, 400))]

Length of task list: 5

Number of processes used: 3

55% | 691/1261 [30:44<25:21, 2.67s/it]

The times for each task are: [1.009872, 1.627487, 0.619629, 0.441103, 0.312622] with:

Minimum: 0.312622 Maximum: 1.627487 Average: 0.8021 seconds

Number of cars 1

bbox:car_number ((865, 400), (944, 472)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (875, 400))]

Length of task list: 5

Number of processes used: 3

55% | 692/1261 [30:47<25:18, 2.67s/it]

The times for each task are: [0.898391, 0.67302, 1.728984, 0.414381, 0.29051] with:

Minimum: 0.29051 Maximum: 1.728984 Average: 0.8011 seconds

Number of cars 1

bbox:car_number ((875, 400), (942, 472)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (942, 472))]
Length of task list: 5
Number of processes used: 3

55% | 693/1261 [30:49<25:16, 2.67s/it]

The times for each task are: [0.642545, 1.378623, 0.976444, 0.322448, 0.402553] with:
Minimum: 0.322448 Maximum: 1.378623 Average: 0.7445 seconds

Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
((875, 400), (942, 472)) Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (942, 472))]
Length of task list: 5
Number of processes used: 3

55% | 694/1261 [30:52<25:13, 2.67s/it]

The times for each task are: [0.870806, 1.411056, 0.714093, 0.407598, 0.293516] with:
Minimum: 0.293516 Maximum: 1.411056 Average: 0.7394 seconds

Number of cars 1
bbox:car_number ((875, 400), (944, 472)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
((875, 400), (944, 472)) Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (944, 472))]
Length of task list: 5
Number of processes used: 3

55% | 695/1261 [30:54<25:10, 2.67s/it]

The times for each task are: [1.397238, 0.95402, 0.683945, 0.471931, 0.32159] with:
Minimum: 0.32159 Maximum: 1.397238 Average: 0.7657 seconds

Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2

Length of task list: 5
Number of processes used: 3

55% | 696/1261 [30:57<25:07, 2.67s/it]

The times for each task are: [0.913236, 0.71173, 1.459971, 0.427601, 0.332065] with:

Minimum: 0.332065 Maximum: 1.459971 Average: 0.7689 seconds

Number of cars 1

bbox:car_number ((875, 400), (934, 472)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Length of task list: 5
Number of processes used: 3

55% | 697/1261 [30:59<25:04, 2.67s/it]

The times for each task are: [0.891515, 0.60849, 1.428367, 0.451798, 0.30024] with:

Minimum: 0.30024 Maximum: 1.428367 Average: 0.7361 seconds

Number of cars 1

`bbox:car number ((865, 400), (942, 478)) : 1`

The minimum distance from car: 1 is 3.1622776601683795

totalCars: 2

Length of task list: 5
Number of processes used: 3

55% | 698/1261 [31:02<25:02, 2.67s/it]

The times for each task are: [0.617095, 1.58762, 0.88963, 0.315995, 0.48542] with:

Minimum: 0.315995 Maximum: 1.58762 Average: 0.7792 seconds

Number of cars 1

`bbox:car number ((865, 400), (942, 478)) : 1`

The minimum distance from car: 1 is 0.0

totalCars: 2

(72)) ((875 4700 Num(1942 1472)in Ps((1875 4000((10924 4072))) (110375 4850))) (((914551 4790))) (((18753 4495))) (934105472 4400))

Length of task list: 5
Number of processes used: 3

55% | 699/1261 [31:05<24:59, 2.67s/it]

The times for each task are: [0.897975, 0.665776, 1.428088, 0.509217, 0.302998] with:

Minimum: 0.302998 Maximum: 1.428088 Average: 0.7608 seconds

Number of cars 1

bbox:car_number ((875, 400), (942, 477)) : 1

The minimum distance from car: 1 is 5.0990195135927845

totalCars: 2

((727)), ((875, 420, Num(18442, 14720)), Pds(187150, n400D), ((109754, 44272)),), (1(1337, 5, 48400)), , ((91405, 1, 47400)), , ((181755, 44905)), , (93(41,0 5417,2)400),

Length of task list: 5

Number of processes used: 3

56% | 700/1261 [31:07<24:56, 2.67s/it]

The times for each task are: [1.370481, 0.889827, 0.615463, 0.396176, 0.286895] with:

Minimum: 0.286895 Maximum: 1.370481 Average: 0.7118 seconds

Number of cars 2

bbox:car_number ((875, 415), (934, 472)) : 1

The minimum distance from car: 1 is 6.4031242374328485

bbox:car_number ((1181, 431), (1253, 495)) : 2

The minimum distance from car: 1 is 28.071337695236398

totalCars: 2

((-72)), ((875, 440, Num(3442, 14720), Pds(3875, 44000), ((108754, 44272)), ((110375, 44850))), (((914251, 474200)), (((187538, 44905)), ((934105472, 44000))))

totalCars: 2

Car Number: 1 Car Positions

56% | 701/1261 [31:10<24:54 2 67s/it]

The times for each task are: [0.840221 0.686871 1.372912 0.413147 0.410801] with:

Minimum: 0.410801 Maximum: 1.372912 Average: 0.7448 seconds

Number of cars ?

bbox:car number ((865 400) (944 478)) : 1

```
The minimum distance from car: 1 is 4.0
bbox:car_number ((1195, 460), (1212, 462)) : 2
totalCars: 2
72)), ((875, 400), Num(842, 472)), Pos(875, 400)((10754, 4272)), (11375, 4850)), ((9425, 4720)), (((18753, 4005)), ), (9341, 0547, 2)40
Length of task list: 5
Number of processes used: 3
```

56% | 702/1261 [31:13<24:51, 2.67s/it]

The times for each task are: [0.616023, 0.988889, 1.657373, 0.305745, 0.394017] with:
Minimum: 0.305745 Maximum: 1.657373 Average: 0.7924 seconds

```
Number of cars 1
bbox:car_number ((865, 400), (942, 477)) : 1
The minimum distance from car: 1 is 1.4142135623730951
totalCars: 2
72)), ((875, 400), Num(842, 472)), Pos(875, 400)((10754, 4272)), (11375, 4850)), ((9425, 4720)), (((18753, 4005)), ), (9341, 0547, 2)40
Length of task list: 5
Number of processes used: 3
```

56% | 703/1261 [31:15<24:48, 2.67s/it]

The times for each task are: [0.872324, 1.383863, 0.713683, 0.405933, 0.348958] with:
Minimum: 0.348958 Maximum: 1.383863 Average: 0.745 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 2
72)), ((875, 400), Num(842, 472)), Pos(875, 400)((10754, 4272)), (11375, 4850)), ((9425, 4720)), (((18753, 4005)), ), (9341, 0547, 2)40
Length of task list: 5
Number of processes used: 3
```

56% | 704/1261 [31:18<24:46, 2.67s/it]

The times for each task are: [0.632144, 0.918257, 1.45064, 0.385721, 0.428584] with:
Minimum: 0.385721 Maximum: 1.45064 Average: 0.7631 seconds

```
Number of cars 2
bbox:car_number ((875, 415), (934, 472)) : 1
```

The minimum distance from car: 1 is 8.06225774829855
 bbox:car_number ((1210, 445), (1241, 492)) : 2
 totalCars: 2
 ((875, 410), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1158, 419), (1268, 529)))
 Length of task list: 5
 Number of processes used: 3

56% | 705/1261 [31:21<24:43, 2.67s/it]

The times for each task are: [0.623171, 0.876164, 0.445962, 1.583327, 0.391693] with:
 Minimum: 0.391693 Maximum: 1.583327 Average: 0.7841 seconds

Number of cars 2
 bbox:car_number ((875, 415), (934, 472)) : 1
 The minimum distance from car: 1 is 0.0
 bbox:car_number ((1204, 427), (1257, 477)) : 2
 The minimum distance from car: 1 is 17.029386365926403
 totalCars: 2
 ((875, 410), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1158, 419), (1268, 529)))
 Length of task list: 5
 Number of processes used: 3

56% | 706/1261 [31:23<24:40, 2.67s/it]

The times for each task are: [0.654269, 0.857677, 1.429261, 0.303204, 0.39734] with:
 Minimum: 0.303204 Maximum: 1.429261 Average: 0.7284 seconds

Number of cars 2
 bbox:car_number ((875, 415), (934, 472)) : 1
 The minimum distance from car: 1 is 0.0
 bbox:car_number ((1158, 419), (1268, 529)) : 2
 The minimum distance from car: 1 is 27.80287754891569
 totalCars: 2
 ((875, 410), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1158, 419), (1268, 529)))

56% | 707/1261 [31:26<24:38, 2.67s/it]

The times for each task are: [0.876747, 0.694509, 1.418832, 0.415297, 0.361396] with:

Minimum: 0.361396 Maximum: 1.418832 Average: 0.7534 seconds

Number of cars 3

bbox:car_number ((875, 415), (934, 477)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((1175, 419), (1253, 498)) : 2

The minimum distance from car: 1 is 16.0312195418814

bbox:car_number ((1144, 481), (1160, 498)) : 3

totalCars: 2

((875, 415), (1175, 419), (1144, 481)), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1175, 400), (1273, 478)), ((1144, 481), (1160, 498))

totalCars: 2

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1175, 400), (1273, 478)), ((1144, 481), (1160, 498))]

Length of task list: 5

Number of processes used: 3

56% | 708/1261 [31:29<24:35, 2.67s/it]

The times for each task are: [0.58602, 0.990728, 0.402529, 1.717739, 0.294813] with:

Minimum: 0.294813 Maximum: 1.717739 Average: 0.7984 seconds

Number of cars 3

bbox:car_number ((1175, 400), (1257, 497)) : 1

The minimum distance from car: 1 is 10.198039027185569

bbox:car_number ((875, 415), (934, 477)) : 2

The minimum distance from car: 1 is 0.0

bbox:car_number ((1158, 445), (1167, 485)) : 3

totalCars: 2

((875, 415), (1175, 400), (1158, 445)), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1175, 400), (1273, 478)), ((1158, 445), (1167, 485))

totalCars: 2

((875, 415), (1175, 400), (1158, 445)), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1175, 400), (1273, 478)), ((1158, 445), (1167, 485))

Length of task list: 5

Number of processes used: 3

56% | 709/1261 [31:31<24:33, 2.67s/it]

The times for each task are: [0.628806, 0.894241, 1.453438, 0.407443, 0.287588] with:

Minimum: 0.287588 Maximum: 1.453438 Average: 0.7343 seconds

Number of cars 2

```
bbox:car_number ((875, 415), (934, 472)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1156, 419), (1241, 495)) : 2
The minimum distance from car: 1 is 20.12461179749811
totalCars: 2
((875, 400), (942, 472)), ((1136, 400), (1238, 495))
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 495))]
Length of task list: 5
Number of processes used: 3
```

56% | 710/1261 [31:34<24:30, 2.67s/it]

The times for each task are: [0.884646, 0.576112, 1.383307, 0.357924, 0.403515] with:
Minimum: 0.357924 Maximum: 1.383307 Average: 0.7211 seconds

```
Number of cars 2
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 8.06225774829855
bbox:car_number ((1136, 400), (1238, 495)) : 2
The minimum distance from car: 1 is 14.866068747318506
totalCars: 2
((875, 400), (942, 472)), ((1136, 400), (1238, 495))
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 495))]
Length of task list: 5
Number of processes used: 3
```

56% | 711/1261 [31:37<24:27, 2.67s/it]

The times for each task are: [1.058659, 0.630289, 1.411811, 0.296863, 0.431228] with:
Minimum: 0.296863 Maximum: 1.411811 Average: 0.7658 seconds

```
Number of cars 4
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1129, 400), (1230, 495)) : 2
The minimum distance from car: 1 is 8.0
bbox:car_number ((1113, 445), (1124, 492)) : 3
bbox:car_number ((1105, 490), (1107, 492)) : 4
totalCars: 2
((875, 400), (942, 472)), ((1129, 400), (1230, 495)), ((1113, 445), (1124, 492)), ((1105, 490), (1107, 492))
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 478))]  
Length of task list: 5  
Number of processes used: 3
```

56% | 712/1261 [31:40<24:25, 2.67s/it]

The times for each task are: [0.920317, 0.704917, 1.742561, 0.429951, 0.330326] with:

Minimum: 0.330326 Maximum: 1.742561 Average: 0.8256 seconds

```
Number of cars 2  
bbox:car_number ((875, 400), (942, 472)) : 1  
The minimum distance from car: 1 is 0.0  
bbox:car_number ((1113, 400), (1219, 495)) : 2  
The minimum distance from car: 1 is 13.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 478))]  
Length of task list: 5  
Number of processes used: 3
```

57% | 713/1261 [31:43<24:22, 2.67s/it]

The times for each task are: [0.645109, 0.936366, 0.324752, 1.6951, 0.412951] with:

Minimum: 0.324752 Maximum: 1.6951 Average: 0.8029 seconds

```
Number of cars 2  
bbox:car_number ((875, 400), (942, 472)) : 1  
The minimum distance from car: 1 is 0.0  
bbox:car_number ((1082, 400), (1214, 522)) : 2  
The minimum distance from car: 1 is 22.80350850198276  
totalCars: 2  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 478))]  
Length of task list: 5  
Number of processes used: 3
```

57% | 714/1261 [31:46<24:20, 2.67s/it]

The times for each task are: [0.610962, 0.961024, 0.427573, 0.402845, 1.692106] with:

Minimum: 0.402845 Maximum: 1.692106 Average: 0.8189 seconds

Number of cars 2
bbox:car_number ((875, 400), (934, 472)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1060, 419), (1208, 529)) : 2
The minimum distance from car: 1 is 19.1049731745428
totalCars: 2
((875, 400), (934, 472)), Pos((875, 400), ((1060, 419), (1208, 529))), ((1175, 485)), ((945, 472)), ((1875, 495)), (934, 472)
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

57% | 715/1261 [31:49<24:18, 2.67s/it]

The times for each task are: [0.674726, 1.530758, 0.991686, 0.396969, 0.471094] with:

Minimum: 0.396969 Maximum: 1.530758 Average: 0.813 seconds

Number of cars 3
bbox:car_number ((875, 400), (934, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1060, 400), (1208, 529)) : 2
The minimum distance from car: 1 is 10.0
bbox:car_number ((1150, 520), (1180, 529)) : 3
totalCars: 2
((875, 400), (934, 472)), Pos((875, 400), ((1060, 400), (1208, 529))), ((1175, 485)), ((945, 472)), ((1875, 495)), (934, 472)
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

57% | 716/1261 [31:52<24:15, 2.67s/it]

The times for each task are: [0.63537, 0.910697, 1.699845, 0.46383, 0.36625] with:

Minimum: 0.36625 Maximum: 1.699845 Average: 0.8152 seconds

Number of cars 2
bbox:car_number ((1082, 400), (1197, 512)) : 1
The minimum distance from car: 1 is 9.433981132056603
bbox:car_number ((875, 415), (934, 462)) : 2
The minimum distance from car: 1 is 2.0
totalCars: 2

```
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 472)), ((1181, 400), (1273, 512)), ((1181, 400), (1273, 472))]  
totalCars: 2  
Car Num(1412, 1472) Car Pos(1875, 400), ((1074, 4272)), ((1137, 4850)), ((945, 14720)), ((1873, 4005)), (9341, 0547, 2400)  
Length of task list: 5  
Number of processes used: 3
```

57% | 717/1261 [31:54<24:12, 2.67s/it]

The times for each task are: [0.638323, 0.883424, 1.703175, 0.323282, 0.44144] with:

Minimum: 0.323282 Maximum: 1.703175 Average: 0.7979 seconds

Number of cars 3

bbox:car_number ((875, 400), (934, 472)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((1075, 400), (1200, 512)) : 2

The minimum distance from car: 1 is 2.0

bbox:car_number ((1198, 520), (1200, 567)) : 3

totalCars: 2

400) Number,

totalCars: 2

Car Number:

Length of task list: 5

Number of cars 3

bbox:car_number ((875, 400), (934, 472))

The minimum distance from car: 1 is 0.0

bbox:car_number ((1102, 400), (1185, 495)) : 2

The minimum distance from car: 1 is 10.8166538

bbox:car_num

totalCars: 2

4500 Num(8942, 1472a)r Pos(87750, nst:00D((109741, 49772)), (10

totalCars: 2

Car Number: 1 Car Positions: [(1204, 400), (1

Length of task list: 5

Number of processes used: 3


```
Number of cars 3
bbox:car_number ((880, 400), (934, 462)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1082, 400), (1167, 485)) : 2
The minimum distance from car: 1 is 21.95449840010015
bbox:car_number ((1195, 481), (1253, 537)) : 3
totalCars: 3
((875, 400), ((1060, 400), (1160, 498))), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512)), ((1195, 481), (1253, 537))
totalCars: 3
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
totalCars: 3
Car Number: 3 Car Positions: [((1195, 481), (1253, 537))]
Length of task list: 5
Number of processes used: 3
```

57% | 722/1261 [32:08<23:59, 2.67s/it]

The times for each task are: [0.673833, 0.909147, 1.408927, 0.344613, 0.407041] with:

Minimum: 0.344613 Maximum: 1.408927 Average: 0.7487 seconds

```
Number of cars 3
bbox:car_number ((875, 400), (942, 462)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1060, 400), (1162, 498)) : 2
The minimum distance from car: 1 is 14.7648230602334
bbox:car_number ((1165, 535), (1167, 537)) : 3
totalCars: 3
((875, 400), ((1060, 400), (1160, 498))), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512)), ((1195, 481), (1253, 537))
totalCars: 3
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

57% | 723/1261 [32:11<23:57, 2.67s/it]

The times for each task are: [0.9244, 0.596336, 0.498248, 0.401951, 1.679628] with:

Minimum: 0.401951 Maximum: 1.679628 Average: 0.8201 seconds

```
Number of cars 2
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1060, 400), (1160, 498)) : 2
The minimum distance from car: 1 is 1.0
```

```
totalCars: 3
((875, 400), (942, 472)), Pos((875, 400), ((1075, 472)), (1175, 485)), ((945, 472)), ((1875, 495)), (934, 547, 400)
totalCars: 3
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 472))]
Length of task list: 5
Number of processes used: 3
```

57% | 724/1261 [32:13<23:54, 2.67s/it]

The times for each task are: [0.933702, 1.445421, 0.735935, 0.406739, 0.36522] with:

Minimum: 0.36522 Maximum: 1.445421 Average: 0.7774 seconds

```
Number of cars 3
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1067, 400), (1152, 516)) : 2
The minimum distance from car: 1 is 9.055385138137417
bbox:car_number ((1144, 505), (1200, 573)) : 3
totalCars: 4
((875, 400), (942, 472)), Pos((875, 400), ((1075, 472)), (1175, 485)), ((945, 472)), ((1875, 495)), (934, 547, 400)
totalCars: 4
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 472))]
totalCars: 4
Car Number: 3 Car Positions: [((1144, 505), (1200, 573))]
Length of task list: 5
Number of processes used: 3
```

57% | 725/1261 [32:16<23:51, 2.67s/it]

The times for each task are: [0.657821, 0.874503, 1.696398, 0.307022, 0.500392] with:

Minimum: 0.307022 Maximum: 1.696398 Average: 0.8072 seconds

```
Number of cars 2
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1075, 400), (1143, 477)) : 2
The minimum distance from car: 1 is 20.0
totalCars: 4
((875, 400), (942, 472)), Pos((875, 400), ((1075, 472)), (1175, 485)), ((945, 472)), ((1875, 495)), (934, 547, 400)
totalCars: 4
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 472))]
Length of task list: 5
Number of processes used: 3
```

58% | 726/1261 [32:19<23:49, 2.67s/it]

The times for each task are: [1.018934, 1.39928, 0.688354, 0.463891, 0.402861] with:

Minimum: 0.402861 Maximum: 1.39928 Average: 0.7947 seconds

Number of cars 2

bbox:car_number ((880, 400), (942, 462)) : 1

The minimum distance from car: 1 is 5.830951894845301

bbox:car_number ((1048, 400), (1143, 485)) : 2

The minimum distance from car: 1 is 14.560219778561036

totalCars: 4

totalCars: 4

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

58% | 727/1261 [32:22<23:46, 2.67s/it]

The times for each task are: [0.88881, 1.414952, 0.412468, 0.717185, 0.315149] with:

Minimum: 0.315149 Maximum: 1.414952 Average: 0.7497 seconds

Number of cars 5

bbox:car number ((875, 400), (944, 472)) : 1

The minimum distance from car: 1 is 5.385164807134504

bbox:car number ((1045, 400), (1143, 507)) : 2

The minimum distance from car: 1 is 11.045361017187261

bbox:car number ((1090, 505), (1107, 507)) : 3

bbox:car number ((1129, 508), (1133, 512)) : 4

bbox:car number ((1135, 508), (1137, 552)) : 5

totalCars: 4

2400 Number 42

totalCars: 4
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Car Number: _____
Length of tape _____

Number of processes used: 3

Number of processes used

58% | 728/1261 [32:24<23:43, 2.67s/it]

The times for each task are: [0.628908, 1.019338, 1.420328, 0.310631, 0.408083] with:

Minimum: 0.310631 Maximum: 1.420328 Average: 0.7575 seconds

Number of cars 8

bbox:car_number ((880, 400), (942, 462)) : 1

The minimum distance from car: 1 is 5.385164807134504

bbox:car_number ((1046, 400), (1137, 498)) : 2

The minimum distance from car: 1 is 5.0

bbox:car_number ((1113, 492), (1116, 495)) : 3

bbox:car_number ((1122, 495), (1149, 498)) : 4

bbox:car_number ((1165, 495), (1180, 498)) : 5

bbox:car_number ((1135, 505), (1137, 552)) : 6

bbox:car_number ((1165, 505), (1181, 522)) : 7

bbox:car_number ((1180, 535), (1181, 552)) : 8

totalCars: 4

((875, 400), Num(842, 1472)), Pos(875, 400), ((10754, 4272)), ((11375, 4850)), ((9425, 14720)), ((18133, 4495)), (9341, 05472)400
totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

58% | 729/1261 [32:27<23:41, 2.67s/it]

The times for each task are: [1.365151, 0.903914, 0.614169, 0.424126, 0.337372] with:

Minimum: 0.337372 Maximum: 1.365151 Average: 0.7289 seconds

Number of cars 4

bbox:car_number ((882, 400), (942, 459)) : 1

The minimum distance from car: 1 is 2.23606797749979

bbox:car_number ((1065, 415), (1133, 485)) : 2

The minimum distance from car: 1 is 8.06225774829855

bbox:car_number ((1113, 493), (1116, 495)) : 3

bbox:car_number ((1120, 495), (1181, 564)) : 4

The minimum distance from car: 3 is 24.166091947189145

totalCars: 4

((875, 400), Num(842, 1472)), Pos(875, 400), ((10754, 4272)), ((11375, 4850)), ((9425, 14720)), ((18133, 4495)), (9341, 05472)400
totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

totalCars: 4

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

58% | 730/1261 [32:30<23:38, 2.67s/it]

The times for each task are: [0.96132, 0.676263, 1.406151, 0.344425, 0.400778] with:

Minimum: 0.344425 Maximum: 1.406151 Average: 0.7578 seconds

Number of cars 5

bbox:car_number ((880, 400), (942, 462)) : 1

The minimum distance from car: 1 is 2.23606797749979

bbox:car_number ((1060, 400), (1133, 485)) : 2

The minimum distance from car: 1 is 8.54400374531753

bbox:car_number ((1102, 481), (1133, 485)) : 3

bbox:car_number ((1113, 493), (1180, 560)) : 4

The minimum distance from car: 3 is 5.0

bbox:car_number ((1183, 493), (1185, 539)) : 5

totalCars: 4

((875, 400), Num(842, 1472)), Pos(875, 400), ((10754, 4272)), , ((11375, 4850)), , ((9425, 14720)), , ((81753, 4495)), , (9341, 05472)40

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

totalCars: 4

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 493), (1185, 539))]

Length of task list: 5

Number of processes used: 3

58% | 731/1261 [32:33<23:36, 2.67s/it]

The times for each task are: [0.918862, 1.349063, 0.652345, 0.339812, 0.473104] with:

Minimum: 0.339812 Maximum: 1.349063 Average: 0.7466 seconds

Number of cars 5

bbox:car_number ((880, 400), (953, 472)) : 1

The minimum distance from car: 1 is 7.0710678118654755

bbox:car_number ((1044, 400), (1122, 485)) : 2

The minimum distance from car: 1 is 13.0

bbox:car_number ((1144, 446), (1257, 560)) : 3

The minimum distance from car: 3 is 24.73863375370596

bbox:car_number ((1082, 505), (1087, 539)) : 4

bbox:car_number ((1102, 508), (1122, 539)) : 5

totalCars: 4

((875, 400), Num(842, 1472)), Pos(875, 400), ((10754, 4272)), , ((11375, 4850)), , ((9425, 14720)), , ((81753, 4495)), , (9341, 05472)40

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

totalCars: 4

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560))]

Length of task list: 5

Number of processes used: 3

58% | 732/1261 [32:35<23:33, 2.67s/it]

The times for each task are: [1.026344, 0.763303, 1.451611, 0.344477, 0.425903] with:

Minimum: 0.344477 Maximum: 1.451611 Average: 0.8023 seconds

Number of cars 6

bbox:car_number ((882, 400), (953, 462)) : 1

The minimum distance from car: 1 is 5.0990195135927845

bbox:car_number ((1044, 400), (1124, 485)) : 2

The minimum distance from car: 1 is 1.0

bbox:car_number ((1195, 460), (1230, 516)) : 3

bbox:car_number ((1113, 514), (1182, 582)) : 4

The minimum distance from car: 3 is 22.02271554554524

bbox:car_number ((1158, 514), (1162, 518)) : 5

bbox:car_number ((1175, 514), (1185, 518)) : 6

totalCars: 4

((875, 400), Num(1842, 1472)), Pos(1875, 400), ((10754, 4272)), , ((11375, 4850)), , ((9405, 14720)), , ((18133, 4995)), , (9341, 05472)40

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 586))]

totalCars: 4

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 481), (1185, 518))]

Length of task list: 5

Number of processes used: 3

58% | 733/1261 [32:38<23:30, 2.67s/it]

The times for each task are: [0.933425, 0.627756, 1.430577, 0.381697, 0.446902] with:

Minimum: 0.381697 Maximum: 1.430577 Average: 0.7641 seconds

Number of cars 4

bbox:car_number ((882, 400), (942, 462)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1044, 400), (1116, 495)) : 2

The minimum distance from car: 1 is 6.4031242374328485

bbox:car_number ((1102, 446), (1257, 586)) : 3

The minimum distance from car: 3 is 24.698178070456937

bbox:car_number ((1090, 492), (1093, 495)) : 4

totalCars: 4

((875, 400), Num(1842, 1472)), Pos(1875, 400), ((10754, 4272)), , ((11375, 4850)), , ((9405, 14720)), , ((18133, 4995)), , (9341, 05472)40

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 586))]

totalCars: 4

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1185, 518))]

Length of task list: 5

Number of processes used: 3

58%| 734/1261 [32:40<23:27, 2.67s/it]

The times for each task are: [0.895509, 1.414089, 0.777823, 0.412406, 0.315664] with:

Minimum: 0.315664 Maximum: 1.414089 Average: 0.7631 seconds

Number of cars 4

bbox:car_number ((880, 400), (953, 477)) : 1

The minimum distance from car: 1 is 8.06225774829855

bbox:car_number ((1044, 400), (1107, 498)) : 2

The minimum distance from car: 1 is 5.385164807134504

bbox:car_number ((1105, 431), (1253, 586)) : 3

The minimum distance from car: 3 is 8.0

bbox:car_number ((1144, 493), (1149, 498)) : 4

totalCars: 4

((72)), ((875, 400) Num(1842, 1472) Car Pos(1875 on 400)((10754, 4272)), (11375, 4850)), ((9425, 14720)), (((18753, 4005)),), (9341, 05472) 400

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 560))]

totalCars: 4

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1204, 512))]

Length of task list: 5

Number of processes used: 3

58%| 735/1261 [32:43<23:25, 2.67s/it]

The times for each task are: [0.889959, 1.401552, 0.704784, 0.414755, 0.40245] with:

Minimum: 0.40245 Maximum: 1.401552 Average: 0.7627 seconds

Number of cars 3

bbox:car_number ((880, 400), (953, 477)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1021, 400), (1107, 498)) : 2

The minimum distance from car: 1 is 11.0

bbox:car_number ((1102, 446), (1242, 591)) : 3

The minimum distance from car: 3 is 12.206555615733702

totalCars: 4

((72)), ((875, 400) Num(1842, 1472) Car Pos(1875 on 400)((10754, 4272)), (11375, 4850)), ((9425, 14720)), (((18753, 4005)),), (9341, 05472) 400

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 560))]

totalCars: 4

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1204, 512))]

Length of task list: 5

Number of processes used: 3

58% | 736/1261 [32:46<23:22, 2.67s/it]

The times for each task are: [0.904399, 0.613498, 1.619321, 0.305258, 0.404539] with:

Minimum: 0.305258 Maximum: 1.619321 Average: 0.7694 seconds

Number of cars 2

bbox:car_number ((880, 400), (954, 477)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1020, 400), (1242, 592)) : 2

The minimum distance from car: 3 is 46.52956049652737

totalCars: 4

((875, 400), ((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44

Length of task list: 5

Number of processes used: 3

58% | 737/1261 [32:49<23:19, 2.67s/it]

The times for each task are: [0.901893, 0.636717, 1.524222, 0.323898, 0.535218] with:

Minimum: 0.323898 Maximum: 1.524222 Average: 0.7844 seconds

Number of cars 4

bbox:car_number ((882, 400), (953, 478)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1021, 400), (1241, 592)) : 2

The minimum distance from car: 3 is 0.0

bbox:car_number ((1210, 550), (1211, 560)) : 3

bbox:car_number ((1225, 550), (1241, 560)) : 4

totalCars: 4

((875, 400), ((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44

Length of task list: 5

Number of processes used: 3

59% | 738/1261 [32:51<23:17, 2.67s/it]

The times for each task are: [0.594538, 1.5068, 1.103262, 0.490249, 0.357201] with:

Minimum: 0.357201 Maximum: 1.5068 Average: 0.8104 seconds

Number of cars 4

bbox:car_number ((880, 400), (957, 478)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1027, 400), (1133, 498)) : 2

The minimum distance from car: 1 is 16.0

bbox:car_number ((1195, 431), (1238, 447)) : 3

bbox:car_number ((1084, 454), (1241, 586)) : 4

The minimum distance from car: 3 is 31.76476034853718

totalCars: 4

((875, 400), Num(1842, 1472), Car, Pos(1875, 400D), ((10754, 4272)), , (11137, 5, 4850)), , ((9425, 1, 4720)), , ((18733, 4495)), ,(9341, 0, 5472)400

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

totalCars: 4

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

59% | 739/1261 [32:54<23:14, 2.67s/it]

The times for each task are: [0.627584, 0.932028, 1.447831, 0.301067, 0.399757] with:

Minimum: 0.301067 Maximum: 1.447831 Average: 0.7417 seconds

Number of cars 2

bbox:car_number ((882, 400), (957, 477)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((1021, 400), (1230, 591)) : 2

The minimum distance from car: 3 is 6.082762530298219

totalCars: 4

((875, 400), Num(1842, 1472), Car, Pos(1875, 400D), ((10754, 4272)), , (11137, 5, 4850)), , ((9425, 1, 4720)), , ((18733, 4495)), ,(9341, 0, 5472)400

totalCars: 4

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1257, 560))]

Length of task list: 5

Number of processes used: 3

59% | 740/1261 [32:57<23:12, 2.67s/it]

The times for each task are: [0.986342, 0.705526, 1.44341, 0.400909, 0.323906] with:

Minimum: 0.323906 Maximum: 1.44341 Average: 0.772 seconds

Number of cars 2

bbox:car_number ((880, 400), (957, 478)) : 1

```
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1021, 400), (1242, 582)) : 2
The minimum distance from car: 3 is 7.211102550927978
totalCars: 4
((875, 400), Num(1842, 1472)ar, Pos(1875, 400D)((10754, 4072)), (11337, 4850)), ((91051, 4720)), ((18153, 4095)), (934105472)40
totalCars: 4
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44
Length of task list: 5
Number of processes used: 3
```

59% | 741/1261 [32:59<23:09, 2.67s/it]

```
The times for each task are: [0.58099, 1.440089, 1.064719, 0.410908, 0.300457] with:
Minimum: 0.300457 Maximum: 1.440089 Average: 0.7594 seconds
```

```
Number of cars 5
bbox:car_number ((882, 400), (957, 477)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1027, 400), (1086, 472)) : 2
The minimum distance from car: 1 is 27.294688127912362
bbox:car_number ((1067, 446), (1214, 586)) : 3
The minimum distance from car: 3 is 22.360679774997898
bbox:car_number ((1082, 462), (1093, 472)) : 4
bbox:car_number ((1051, 493), (1067, 497)) : 5
totalCars: 4
((875, 400), Num(1842, 1472)ar, Pos(1875, 400D)((10754, 4072)), (11337, 4850)), ((91051, 4720)), ((18153, 4095)), (934105472)40
totalCars: 4
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
totalCars: 4
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

59% | 742/1261 [33:02<23:06, 2.67s/it]

```
The times for each task are: [0.632521, 1.418945, 0.922685, 0.33336, 0.438277] with:
Minimum: 0.33336 Maximum: 1.418945 Average: 0.7492 seconds
```

```
Number of cars 2
bbox:car_number ((886, 400), (971, 512)) : 1
The minimum distance from car: 1 is 20.12461179749811
bbox:car_number ((1008, 400), (1214, 560)) : 2
The minimum distance from car: 3 is 22.825424421026653
```

```

totalCars: 4
((875, 400), Num(942, 1472), Pos(875, 400), ((10754, 4072)), (11(33, 4850)), ((9405, 14720)), (((18753, 4095)), ), (93(4, 05472)40
totalCars: 4
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44
Length of task list: 5
Number of processes used: 3

59%|    | 743/1261 [33:05<23:04, 2.67s/it]

The times for each task are: [1.33345, 0.860944, 0.58984, 0.314836, 0.405941] with:
Minimum: 0.314836 Maximum: 1.33345 Average: 0.701 seconds

Number of cars 2
bbox:car_number ((880, 400), (971, 507)) : 1
The minimum distance from car: 1 is 4.242640687119285
bbox:car_number ((998, 400), (1238, 539)) : 2
The minimum distance from car: 3 is 13.038404810405298
totalCars: 4
((875, 400), Num(942, 1472), Pos(875, 400), ((10754, 4072)), (11(33, 4850)), ((9405, 14720)), (((18753, 4095)), ), (93(4, 05472)40
totalCars: 4
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44
Length of task list: 5
Number of processes used: 3

59%|    | 744/1261 [33:08<23:01, 2.67s/it]

The times for each task are: [0.875551, 0.658171, 1.449532, 0.487035, 0.293233] with:
Minimum: 0.293233 Maximum: 1.449532 Average: 0.7527 seconds

Number of cars 3
bbox:car_number ((882, 400), (957, 478)) : 1
The minimum distance from car: 1 is 15.231546211727817
bbox:car_number ((1021, 415), (1242, 560)) : 2
The minimum distance from car: 3 is 22.20360331117452
bbox:car_number ((970, 481), (971, 485)) : 3
totalCars: 4
((875, 400), Num(942, 1472), Pos(875, 400), ((10754, 4072)), (11(33, 4850)), ((9405, 14720)), (((18753, 4095)), ), (93(4, 05472)40
totalCars: 4
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44
Length of task list: 5
Number of processes used: 3

```

59% | 745/1261 [33:10<22:58, 2.67s/it]

The times for each task are: [0.866806, 1.375131, 0.728332, 0.387242, 0.279585] with:

Minimum: 0.279585 Maximum: 1.375131 Average: 0.7274 seconds

Number of cars 4

bbox:car_number ((886, 400), (963, 485)) : 1

The minimum distance from car: 1 is 5.830951894845301

bbox:car_number ((967, 400), (1077, 560)) : 2

bbox:car_number ((1060, 431), (1253, 582)) : 3

The minimum distance from car: 3 is 18.867962264113206

bbox:car_number ((1067, 460), (1106, 485)) : 4

totalCars: 5

Car Number: 2 Car Positions: [((967, 400), (1077, 560))]

totalCars: 5

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

Length of task list: 5

Number of processes used: 3

59% | 746/1261 [33:13<22:56, 2.67s/it]

The times for each task are: [1.338382, 0.688782, 0.985337, 0.425045, 0.420579] with:

Minimum: 0.420579 Maximum: 1.338382 Average: 0.7716 seconds

Number of cars 3

bbox:car_number ((886, 400), (963, 485)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1021, 431), (1273, 566)) : 2

The minimum distance from car: 3 is 12.041594578792296

bbox:car_number ((967, 462), (971, 485)) : 3

totalCars: 5

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

Length of task list: 5

Number of processes used: 3

59% | 747/1261 [33:15<22:53, 2.67s/it]

The times for each task are: [0.722892, 0.916559, 1.448439, 0.456079, 0.29247] with:

Minimum: 0.29247 Maximum: 1.448439 Average: 0.7673 seconds

```
Number of cars 2
bbox:car_number ((886, 400), (957, 478)) : 1
The minimum distance from car: 1 is 4.242640687119285
bbox:car_number ((1015, 415), (1257, 591)) : 2
The minimum distance from car: 3 is 12.083045973594572
totalCars: 5
Car Number: 1 Pos: (886, 400) ((886, 400), (957, 478))
Car Number: 2 Pos: (1015, 415) ((1015, 415), (1257, 591))
Car Number: 3 Pos: (1144, 505) ((1144, 505), (1200, 573))
Car Number: 4 Pos: (1120, 495) ((1120, 495), (1181, 564))
Car Number: 5 Pos: (1113, 495) ((1113, 495), (1174, 564))
Length of task list: 5
Number of processes used: 3
```

59% | 748/1261 [33:18<22:50, 2.67s/it]

The times for each task are: [0.631124, 1.393217, 1.034141, 0.320497, 0.470042] with:

Minimum: 0.320497 Maximum: 1.393217 Average: 0.7698 seconds

```
Number of cars 2
bbox:car_number ((882, 400), (963, 485)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((994, 400), (1187, 567)) : 2
The minimum distance from car: 3 is 41.19465984809196
totalCars: 5
Car Number: 1 Car Position: ((882, 400), (1075, 447))
Car Number: 2 Car Position: ((994, 400), (1187, 567))
Car Number: 3 Car Position: ((1195, 481), (1253, 537))
Length of task list: 5
Number of processes used: 3
```

59% | 749/1261 [33:21<22:48, 2.67s/it]

The times for each task are: [0.937446, 0.730682, 1.48777, 0.371705, 0.463389] with:

Minimum: 0.371705 Maximum: 1.48777 Average: 0.7982 seconds

```
Number of cars 5
bbox:car_number ((894, 400), (957, 472)) : 1
The minimum distance from car: 1 is 6.708203932499369
bbox:car_number ((994, 400), (1106, 539)) : 2
The minimum distance from car: 2 is 30.083217912982647
bbox:car_number ((1141, 415), (1241, 518)) : 3
bbox:car number ((1113, 431), (1139, 472)) : 4
```

```
bbox:car_number ((1075, 508), (1122, 541)) : 5
totalCars: 6
((875, 400), (957, 472)), ((994, 400), (1257, 573))
totalCars: 6
Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539))]
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

59% | 750/1261 [33:24<22:45, 2.67s/it]

The times for each task are: [0.947015, 1.431278, 0.626162, 0.319106, 0.39776] with:
Minimum: 0.319106 Maximum: 1.431278 Average: 0.7443 seconds

```
Number of cars 2
bbox:car_number ((894, 400), (957, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((994, 400), (1257, 573)) : 2
The minimum distance from car: 3 is 20.248456731316587
totalCars: 6
((875, 400), (957, 472)), ((994, 400), (1257, 573))
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]
Length of task list: 5
Number of processes used: 3
```

60% | 751/1261 [33:26<22:42, 2.67s/it]

The times for each task are: [1.394576, 0.619148, 0.904536, 0.31244, 0.437224] with:
Minimum: 0.31244 Maximum: 1.394576 Average: 0.7336 seconds

```
Number of cars 1
bbox:car_number ((910, 415), (953, 459)) : 1
Length of task list: 5
Number of processes used: 3
```

60% | 752/1261 [33:29<22:40, 2.67s/it]

The times for each task are: [0.631005, 0.980935, 0.416186, 1.54127, 0.347872] with:

Minimum: 0.347872 Maximum: 1.54127 Average: 0.7835 seconds

Number of cars 5

bbox:car_number ((895, 400), (953, 462)) : 1

The minimum distance from car: 1 is 5.0990195135927845

bbox:car_number ((1181, 469), (1185, 472)) : 2

bbox:car_number ((1113, 514), (1122, 518)) : 3

bbox:car_number ((1113, 520), (1122, 529)) : 4

bbox:car_number ((1120, 535), (1122, 552)) : 5

totalCars: 6

((275, 460), ((1075, 400), (1075, 4472)), ((1075, 4850), ((9405, 4720), ((18755, 4005), (9341, 05472400

Length of task list: 5

Number of processes used: 3

60% | 753/1261 [33:32<22:37, 2.67s/it]

The times for each task are: [0.581902, 0.90637, 0.452158, 1.61556, 0.368484] with:

Minimum: 0.368484 Maximum: 1.61556 Average: 0.7849 seconds

Number of cars 2

bbox:car_number ((895, 415), (942, 459)) : 1

bbox:car_number ((1136, 446), (1242, 518)) : 2

The minimum distance from car: 3 is 16.1245154965971

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518))]

Length of task list: 5

Number of processes used: 3

60% | 754/1261 [33:35<22:34, 2.67s/it]

The times for each task are: [1.432551, 0.88169, 0.713097, 0.393928, 0.292952] with:

Minimum: 0.292952 Maximum: 1.432551 Average: 0.7428 seconds

Number of cars 6

bbox:car_number ((894, 400), (954, 462)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1113, 431), (1182, 485)) : 2

The minimum distance from car: 3 is 35.608987629529715

bbox:car_number ((1102, 476), (1105, 485)) : 3

bbox:car_number ((1090, 514), (1105, 518)) : 4

bbox:car_number ((1075, 520), (1105, 535)) : 5

bbox:car_number ((1120, 535), (1122, 539)) : 6

totalCars: 6

((875, 400), ((1142, 1472)), ((10754, 4472)), ((11375, 4850)), ((9425, 14720)), ((18753, 4495)), ((9341, 05472))
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1136, 446))]
Length of task list: 5
Number of processes used: 3

60% | 755/1261 [33:37<22:32, 2.67s/it]

The times for each task are: [0.970127, 0.631299, 0.413394, 1.448381, 0.2978] with:

Minimum: 0.2978 Maximum: 1.448381 Average: 0.7522 seconds

Number of cars 2
bbox:car_number ((895, 400), (954, 459)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1090, 445), (1230, 529)) : 2
The minimum distance from car: 3 is 29.427877939124322
totalCars: 6
((875, 400), ((1142, 1472)), ((10754, 4472)), ((11375, 4850)), ((9425, 14720)), ((18753, 4495)), ((9341, 05472))
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1136, 446))]
Length of task list: 5
Number of processes used: 3

60% | 756/1261 [33:39<22:29, 2.67s/it]

The times for each task are: [0.876094, 1.42514, 0.715286, 0.344078, 0.406625] with:

Minimum: 0.344078 Maximum: 1.42514 Average: 0.7534 seconds

Number of cars 4
bbox:car_number ((895, 400), (957, 462)) : 1
The minimum distance from car: 1 is 2.8284271247461903
bbox:car_number ((1000, 415), (1070, 472)) : 2
The minimum distance from car: 1 is 22.135943621178654
bbox:car_number ((1090, 431), (1214, 529)) : 3
The minimum distance from car: 3 is 10.63014581273465
bbox:car_number ((1180, 520), (1180, 529)) : 4
totalCars: 6
((875, 400), ((1142, 1472)), ((10754, 4472)), ((11375, 4850)), ((9425, 14720)), ((18753, 4495)), ((9341, 05472))
totalCars: 6
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1183, 400))]
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1136, 446))]
Length of task list: 5

Number of processes used: 3

60% | 757/1261 [33:42<22:26, 2.67s/it]

The times for each task are: [0.933703, 1.39597, 0.71093, 0.303054, 0.396699] with:

Minimum: 0.303054 Maximum: 1.39597 Average: 0.7481 seconds

Number of cars 2

bbox:car_number ((895, 400), (957, 462)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1048, 431), (1238, 552)) : 2

The minimum distance from car: 3 is 14.212670403551895

totalCars: 6

((72)), ((875, 410), Num(842, 1472)), Pos(875, 400), ((1074, 4272)), , ((1137, 4850)), , ((9425, 14720)), , ((18733, 4495)), , ((341, 05472)40

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

60% | 758/1261 [33:45<22:23, 2.67s/it]

The times for each task are: [0.992516, 1.383249, 0.631444, 0.400017, 0.362908] with:

Minimum: 0.362908 Maximum: 1.383249 Average: 0.754 seconds

Number of cars 3

bbox:car_number ((910, 400), (957, 459)) : 1

bbox:car_number ((1051, 445), (1149, 552)) : 2

The minimum distance from car: 3 is 18.027756377319946

bbox:car_number ((1165, 446), (1185, 477)) : 3

totalCars: 6

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44

Length of task list: 5

Number of processes used: 3

60% | 759/1261 [33:47<22:21, 2.67s/it]

The times for each task are: [0.608726, 1.452423, 0.864442, 0.294375, 0.384436] with:

Minimum: 0.294375 Maximum: 1.452423 Average: 0.7209 seconds

Number of cars 2

```
bbox:car_number ((895, 400), (971, 472)) : 1
The minimum distance from car: 1 is 8.602325267042627
bbox:car_number ((1084, 427), (1214, 518)) : 2
The minimum distance from car: 3 is 14.142135623730951
totalCars: 6
((875, 400), ((1075, 400), (1075, 472)), ((1037, 485), (940, 472)), ((1875, 400), (934, 472)), ((1075, 400), (1075, 472)), ((1037, 485), (940, 472)), ((1875, 400), (934, 472)))
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

60% | 760/1261 [33:50<22:18, 2.67s/it]

The times for each task are: [0.897345, 1.490001, 0.670852, 0.34998, 0.460339] with:
Minimum: 0.34998 Maximum: 1.490001 Average: 0.7737 seconds

```
Number of cars 4
bbox:car_number ((905, 400), (972, 472)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1067, 431), (1227, 518)) : 2
The minimum distance from car: 3 is 2.8284271247461903
bbox:car_number ((1021, 492), (1025, 539)) : 3
bbox:car_number ((1180, 520), (1211, 522)) : 4
totalCars: 6
((875, 400), ((1075, 400), (1075, 472)), ((1037, 485), (940, 472)), ((1875, 400), (934, 472)), ((1075, 400), (1075, 472)), ((1037, 485), (940, 472)), ((1875, 400), (934, 472)))
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

60% | 761/1261 [33:53<22:16, 2.67s/it]

The times for each task are: [0.909606, 0.687446, 1.393607, 0.434092, 0.384318] with:
Minimum: 0.384318 Maximum: 1.393607 Average: 0.7618 seconds

```
Number of cars 6
bbox:car_number ((910, 400), (957, 462)) : 1
bbox:car_number ((1030, 400), (1214, 518)) : 2
The minimum distance from car: 3 is 29.154759474226502
bbox:car_number ((1046, 415), (1047, 459)) : 3
bbox:car_number ((1020, 454), (1025, 459)) : 4
bbox:car_number ((1020, 462), (1025, 462)) : 5
bbox:car_number ((1021, 492), (1025, 498)) : 6
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

60% | 762/1261 [33:56<22:13, 2.67s/it]

The times for each task are: [0.690547, 1.412632, 1.002111, 0.369291, 0.54647] with:

Minimum: 0.369291 Maximum: 1.412632 Average: 0.8042 seconds

```
Number of cars 2
bbox:car_number ((910, 400), (972, 472)) : 1
```

The minimum distance from car: 1 is 3.0

```
bbox:car_number ((1048, 400), (1212, 566)) : 2
```

The minimum distance from car: 3 is 15.264337522473747

```
totalCars: 6
```

```
((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

61% | 763/1261 [33:59<22:11, 2.67s/it]

The times for each task are: [0.904617, 1.489104, 0.631912, 0.438765, 0.400488] with:

Minimum: 0.400488 Maximum: 1.489104 Average: 0.773 seconds

```
Number of cars 2
bbox:car_number ((905, 400), (972, 472)) : 1
```

The minimum distance from car: 1 is 3.0

```
bbox:car_number ((1020, 400), (1212, 566)) : 2
```

The minimum distance from car: 3 is 14.0

```
totalCars: 6
```

```
((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

61% | 764/1261 [34:02<22:08, 2.67s/it]

The times for each task are: [0.935285, 1.456001, 0.618749, 0.444807, 0.381924] with:

Minimum: 0.381924 Maximum: 1.456001 Average: 0.7674 seconds

Number of cars 2

bbox:car_number ((910, 400), (972, 472)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((1045, 415), (1212, 539)) : 2

The minimum distance from car: 3 is 13.416407864998739

totalCars: 6

((72)), ((875, 400), Num(1842, 1472), Pos(1875, 400), ((10754, 4272)), ((11375, 4850)), ((9405, 14720)), ((18153, 4095)), (9341, 05472)40

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

61% | 765/1261 [34:04<22:05, 2.67s/it]

The times for each task are: [0.939415, 0.670602, 0.426343, 1.720838, 0.298654] with:

Minimum: 0.298654 Maximum: 1.720838 Average: 0.8112 seconds

Number of cars 3

bbox:car_number ((910, 400), (972, 462)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1021, 400), (1211, 554)) : 2

The minimum distance from car: 3 is 12.0

bbox:car_number ((1021, 462), (1025, 462)) : 3

totalCars: 6

((72)), ((875, 400), Num(1842, 1472), Pos(1875, 400), ((10754, 4272)), ((11375, 4850)), ((9405, 14720)), ((18153, 4095)), (9341, 05472)40

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

61% | 766/1261 [34:07<22:03, 2.67s/it]

The times for each task are: [0.662186, 1.640248, 0.956619, 0.471172, 0.366188] with:

Minimum: 0.366188 Maximum: 1.640248 Average: 0.8193 seconds

Number of cars 3

bbox:car_number ((895, 400), (977, 477)) : 1

The minimum distance from car: 1 is 8.602325267042627

bbox:car_number ((985, 400), (1211, 539)) : 2

```
The minimum distance from car: 3 is 19.697715603592208
bbox:car_number ((1015, 475), (1017, 477)) : 3
totalCars: 6
72)), ((875, 400), Num(842, 1472)), Car, Pos(875, 400), ((10754, 4272)), , ((10375, 4850)), , ((9425, 14720)), , ((18753, 4495)), ,(9341, 05472)40
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

61% | 767/1261 [34:10<22:00, 2.67s/it]

The times for each task are: [1.426372, 0.603874, 0.94311, 0.358366, 0.519172] with:

Minimum: 0.358366 Maximum: 1.426372 Average: 0.7702 seconds

```
Number of cars 3
bbox:car_number ((910, 400), (972, 472)) : 1
The minimum distance from car: 1 is 5.385164807134504
bbox:car_number ((985, 400), (1208, 560)) : 2
The minimum distance from car: 3 is 11.180339887498949
bbox:car_number ((975, 454), (977, 472)) : 3
totalCars: 6
72)), ((875, 400), Num(842, 1472)), Car, Pos(875, 400), ((10754, 4272)), , ((10375, 4850)), , ((9425, 14720)), , ((18753, 4495)), ,(9341, 05472)40
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

61% | 768/1261 [34:12<21:57, 2.67s/it]

The times for each task are: [0.964877, 1.483465, 0.634182, 0.417283, 0.304504] with:

Minimum: 0.304504 Maximum: 1.483465 Average: 0.7609 seconds

```
Number of cars 3
bbox:car_number ((910, 400), (971, 462)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1030, 400), (1197, 518)) : 2
The minimum distance from car: 3 is 9.0
bbox:car_number ((1020, 446), (1032, 462)) : 3
totalCars: 6
72)), ((875, 400), Num(842, 1472)), Car, Pos(875, 400), ((10754, 4272)), , ((10375, 4850)), , ((9425, 14720)), , ((18753, 4495)), ,(9341, 05472)40
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
```

Number of processes used: 3

61% | 769/1261 [34:15<21:55, 2.67s/it]

The times for each task are: [0.657399, 0.910526, 1.503775, 0.513767, 0.325217] with:

Minimum: 0.325217 Maximum: 1.503775 Average: 0.7821 seconds

Number of cars 2

bbox:car_number ((905, 400), (972, 478)) : 1

The minimum distance from car: 1 is 8.246211251235321

bbox:car_number ((989, 431), (1197, 552)) : 2

The minimum distance from car: 3 is 9.899494936611665

totalCars: 6

((875, 400) Num(842, 1472)ar ,Pos(875ons400)((10754, 4272)),,(1(1375,4850)),, ((9405,1,4720)),, (((18753, 4005)),,(9341,05472)400

totalCars: 6

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44

Length of task list: 5

Number of processes used: 3

61% | 770/1261 [34:18<21:52, 2.67s/it]

The times for each task are: [1.431795, 0.942909, 0.660555, 0.309256, 0.386367] with:

Minimum: 0.309256 Maximum: 1.431795 Average: 0.7462 seconds

Number of cars 2

bbox:car_number ((905, 400), (972, 472)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((998, 400), (1197, 552)) : 2

The minimum distance from car: 3 is 4.123105625617661

totalCars: 6

((875, 400) Num(842, 1472)ar ,Pos(875ons400)((10754, 4272)),,(1(1375,4850)),, ((9405,1,4720)),, (((18753, 4005)),,(9341,05472)400

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

61% | 771/1261 [34:20<21:49, 2.67s/it]

The times for each task are: [0.895114, 0.706352, 1.390576, 0.339572, 0.399441] with:

Minimum: 0.339572 Maximum: 1.390576 Average: 0.7462 seconds

```
Number of cars 4
bbox:car_number ((905, 400), (971, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((998, 431), (1133, 560)) : 2
The minimum distance from car: 3 is 28.284271247461902
bbox:car_number ((1135, 445), (1137, 462)) : 3
bbox:car_number ((1150, 460), (1197, 516)) : 4
totalCars: 6
((875, 400), ((1084, 412), 1472)), Pos(1875, 400), ((1084, 412), 1472), , ((1137, 5, 48500)), , ((9425, 1, 47200)), , ((18133, 4005)), , (9341, 0, 5472), 400
totalCars: 6
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44
Length of task list: 5
Number of processes used: 3
```

61% | 772/1261 [34:23<21:47, 2.67s/it]

The times for each task are: [0.651806, 1.014267, 1.854038, 0.511469, 0.378625] with:

Minimum: 0.378625 Maximum: 1.854038 Average: 0.882 seconds

```
Number of cars 2
bbox:car_number ((910, 400), (972, 462)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1008, 419), (1162, 541)) : 2
The minimum distance from car: 3 is 12.649110640673518
totalCars: 6
((875, 400), ((1084, 412), 1472)), Pos(1875, 400), ((1084, 412), 1472), , ((1137, 5, 48500)), , ((9425, 1, 47200)), , ((18133, 4005)), , (9341, 0, 5472), 400
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

61% | 773/1261 [34:26<21:44, 2.67s/it]

The times for each task are: [0.964267, 1.511897, 0.663785, 0.455636, 0.302384] with:

Minimum: 0.302384 Maximum: 1.511897 Average: 0.7796 seconds

```
Number of cars 2
bbox:car_number ((910, 400), (972, 462)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((994, 415), (1180, 560)) : 2
The minimum distance from car: 3 is 7.280109889280518
totalCars: 6
```

```
72)), ((875, 400) Num(842, 1472) car, Pos(875 on 400)((10754, 4272)), (1(13375, 4850)), ((9405, 1, 47200)), (((18153, 4005)), ), (9341, 0, 5472) 40  
totalCars: 6  
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44  
Length of task list: 5  
Number of processes used: 3
```

61% | 774/1261 [34:28<21:41, 2.67s/it]

The times for each task are: [1.084499, 0.723692, 1.568883, 0.400641, 0.328489] with:

Minimum: 0.328489 Maximum: 1.568883 Average: 0.8212 seconds

```
Number of cars 2  
bbox:car_number ((910, 400), (972, 462)) : 1  
The minimum distance from car: 1 is 0.0  
bbox:car_number ((989, 415), (1180, 554)) : 2  
The minimum distance from car: 3 is 4.242640687119285  
totalCars: 6  
72)), ((875, 400) Num(842, 1472) car, Pos(875 on 400)((10754, 4272)), (1(13375, 4850)), ((9405, 1, 47200)), (((18153, 4005)), ), (9341, 0, 5472) 40  
totalCars: 6  
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44  
Length of task list: 5  
Number of processes used: 3
```

61% | 775/1261 [34:31<21:39, 2.67s/it]

The times for each task are: [0.645452, 1.047188, 1.481328, 0.360869, 0.447499] with:

Minimum: 0.360869 Maximum: 1.481328 Average: 0.7965 seconds

```
Number of cars 2  
bbox:car_number ((910, 400), (972, 462)) : 1  
The minimum distance from car: 1 is 0.0  
bbox:car_number ((998, 400), (1182, 554)) : 2  
The minimum distance from car: 3 is 9.219544457292887  
totalCars: 6  
72)), ((875, 400) Num(842, 1472) car, Pos(875 on 400)((10754, 4272)), (1(13375, 4850)), ((9405, 1, 47200)), (((18153, 4005)), ), (9341, 0, 5472) 40  
totalCars: 6  
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44  
Length of task list: 5  
Number of processes used: 3
```

62% | 776/1261 [34:34<21:36, 2.67s/it]

The times for each task are: [0.912237, 0.717681, 1.463051, 0.407388, 0.35775] with:

Minimum: 0.35775 Maximum: 1.463051 Average: 0.7716 seconds

Number of cars 3

bbox:car_number ((910, 400), (972, 472)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1008, 400), (1182, 560)) : 2

The minimum distance from car: 3 is 5.830951894845301

bbox:car_number ((989, 462), (1001, 462)) : 3

totalCars: 6

((875, 400), ((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

62% | 777/1261 [34:37<21:33, 2.67s/it]

The times for each task are: [1.465153, 0.892768, 0.695399, 0.445679, 0.3884] with:

Minimum: 0.3884 Maximum: 1.465153 Average: 0.7775 seconds

Number of cars 2

bbox:car_number ((913, 415), (977, 472)) : 1

The minimum distance from car: 1 is 8.06225774829855

bbox:car_number ((985, 415), (1181, 554)) : 2

The minimum distance from car: 3 is 12.649110640673518

totalCars: 6

((875, 400), ((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

62% | 778/1261 [34:40<21:31, 2.67s/it]

The times for each task are: [1.395494, 1.078136, 0.624368, 0.322891, 0.378148] with:

Minimum: 0.322891 Maximum: 1.395494 Average: 0.7598 seconds

Number of cars 2

bbox:car_number ((910, 400), (977, 472)) : 1

The minimum distance from car: 1 is 7.280109889280518

bbox:car_number ((985, 427), (1167, 539)) : 2

62%| 781/1261 [34:48<21:23, 2.67s/it]

The times for each task are: [0.868322, 1.615134, 0.686776, 0.359006, 0.432111] with:

Minimum: 0.359006 Maximum: 1.615134 Average: 0.7923 seconds

Number of cars 5

bbox:car_number ((985, 400), (1167, 560)) : 1

The minimum distance from car: 3 is 3.1622776601683795

bbox:car_number ((970, 419), (972, 447)) : 2

bbox:car_number ((970, 457), (972, 459)) : 3

bbox:car_number ((989, 462), (991, 478)) : 4

bbox:car_number ((989, 490), (991, 498)) : 5

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

62%| 782/1261 [34:50<21:20, 2.67s/it]

The times for each task are: [0.905673, 0.646295, 0.45509, 1.660052, 0.366824] with:

Minimum: 0.366824 Maximum: 1.660052 Average: 0.8068 seconds

Number of cars 2

bbox:car_number ((970, 400), (1149, 554)) : 1

The minimum distance from car: 2 is 12.041594578792296

bbox:car_number ((913, 415), (957, 447)) : 2

totalCars: 6

Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539)), ((970, 400),

Length of task list: 5

Number of processes used: 3

62%| 783/1261 [34:53<21:18, 2.67s/it]

The times for each task are: [0.923836, 1.400025, 0.636122, 0.420648, 0.352496] with:

Minimum: 0.352496 Maximum: 1.400025 Average: 0.7466 seconds

Number of cars 2

bbox:car_number ((910, 400), (957, 459)) : 1

bbox:car_number ((989, 400), (1152, 560)) : 2

The minimum distance from car: 3 is 6.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

```
Length of task list: 5
Number of processes used: 3
```

```
62%| 784/1261 [34:56<21:15, 2.67s/it]
```

```
The times for each task are: [0.614309, 0.400144, 0.909048, 0.29541, 1.651252] with:
```

```
Minimum: 0.29541 Maximum: 1.651252 Average: 0.774 seconds
```

```
Number of cars 2
bbox:car_number ((989, 400), (1149, 529)) : 1
The minimum distance from car: 3 is 16.0312195418814
bbox:car_number ((970, 415), (977, 447)) : 2
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
62%| 785/1261 [34:59<21:12, 2.67s/it]
```

```
The times for each task are: [0.933201, 0.619942, 1.622636, 0.397409, 0.314662] with:
```

```
Minimum: 0.314662 Maximum: 1.622636 Average: 0.7776 seconds
```

```
Number of cars 2
bbox:car_number ((975, 400), (1160, 537)) : 1
The minimum distance from car: 3 is 4.47213595499958
bbox:car_number ((970, 427), (972, 447)) : 2
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
62%| 786/1261 [35:01<21:10, 2.67s/it]
```

```
The times for each task are: [0.960414, 0.697464, 0.446818, 1.653178, 0.327166] with:
```

```
Minimum: 0.327166 Maximum: 1.653178 Average: 0.817 seconds
```

```
Number of cars 1
bbox:car_number ((967, 400), (1152, 529)) : 1
The minimum distance from car: 3 is 8.94427190999916
totalCars: 6
```

```
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44  
Length of task list: 5  
Number of processes used: 3
```

```
62%| 787/1261 [35:04<21:07, 2.67s/it]
```

```
The times for each task are: [0.896883, 1.42571, 0.644861, 0.465451, 0.360078] with:
```

```
Minimum: 0.360078 Maximum: 1.42571 Average: 0.7586 seconds
```

```
Number of cars 2  
bbox:car_number ((970, 400), (1152, 552)) : 1  
The minimum distance from car: 2 is 2.23606797749979  
bbox:car_number ((565, 505), (597, 522)) : 2  
totalCars: 6  
Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539)), ((970, 400),  
Length of task list: 5  
Number of processes used: 3
```

```
62%| 788/1261 [35:07<21:04, 2.67s/it]
```

```
The times for each task are: [0.908793, 1.406313, 0.638435, 0.294848, 0.427998] with:
```

```
Minimum: 0.294848 Maximum: 1.406313 Average: 0.7353 seconds
```

```
Number of cars 1  
bbox:car_number ((951, 400), (1137, 529)) : 1  
The minimum distance from car: 3 is 15.0  
totalCars: 6  
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44  
Length of task list: 5  
Number of processes used: 3
```

```
63%| 789/1261 [35:10<21:02, 2.67s/it]
```

```
The times for each task are: [0.8629, 1.384487, 0.640809, 0.402718, 0.345454] with:
```

```
Minimum: 0.345454 Maximum: 1.384487 Average: 0.7273 seconds
```

```
Number of cars 2  
bbox:car_number ((989, 431), (1143, 529)) : 1  
The minimum distance from car: 2 is 6.4031242374328485  
bbox:car_number ((989, 460), (1010, 478)) : 2
```

```
totalCars: 6
Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539)), ((970, 400),
Length of task list: 5
Number of processes used: 3
```

63%| 790/1261 [35:12<20:59, 2.67s/it]

The times for each task are: [1.392993, 0.883243, 0.621093, 0.300734, 0.447127] with:

Minimum: 0.300734 Maximum: 1.392993 Average: 0.729 seconds

```
Number of cars 2
bbox:car_number ((951, 415), (957, 447)) : 1
bbox:car_number ((985, 427), (1143, 552)) : 2
The minimum distance from car: 3 is 6.082762530298219
totalCars: 6
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44
Length of task list: 5
Number of processes used: 3
```

63%| 791/1261 [35:15<20:56, 2.67s/it]

The times for each task are: [0.911609, 1.412881, 0.616847, 0.295399, 0.402646] with:

Minimum: 0.295399 Maximum: 1.412881 Average: 0.7279 seconds

```
Number of cars 1
bbox:car_number ((967, 415), (1143, 529)) : 1
The minimum distance from car: 2 is 13.601470508735444
totalCars: 6
Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539)), ((970, 400),
Length of task list: 5
Number of processes used: 3
```

63%| 792/1261 [35:18<20:54, 2.67s/it]

The times for each task are: [0.892529, 0.682028, 1.383994, 0.431305, 0.391413] with:

Minimum: 0.391413 Maximum: 1.383994 Average: 0.7563 seconds

```
Number of cars 1
bbox:car_number ((958, 400), (1139, 529)) : 1
The minimum distance from car: 3 is 4.0
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1137, 535))]
Length of task list: 5
Number of processes used: 3
```

63%| 793/1261 [35:20<20:51, 2.67s/it]

The times for each task are: [1.344291, 0.557303, 0.424099, 1.017571, 0.307566] with:
Minimum: 0.307566 Maximum: 1.344291 Average: 0.7302 seconds

```
Number of cars 1
bbox:car_number ((955, 400), (1137, 535)) : 1
The minimum distance from car: 3 is 3.605551275463989
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1137, 535))]
Length of task list: 5
Number of processes used: 3
```

63%| 794/1261 [35:22<20:48, 2.67s/it]

The times for each task are: [0.984072, 0.596879, 1.690156, 0.348998, 0.41329] with:
Minimum: 0.348998 Maximum: 1.690156 Average: 0.8067 seconds

```
Number of cars 2
bbox:car_number ((955, 400), (1137, 535)) : 1
The minimum distance from car: 3 is 0.0
bbox:car_number ((910, 446), (927, 462)) : 2
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1137, 535))]
Length of task list: 5
Number of processes used: 3
```

63%| 795/1261 [35:25<20:45, 2.67s/it]

The times for each task are: [0.966253, 0.66359, 1.508543, 0.435834, 0.382254] with:
Minimum: 0.382254 Maximum: 1.508543 Average: 0.7913 seconds

```
Number of cars 1
bbox:car_number ((951, 400), (1133, 529)) : 1
The minimum distance from car: 3 is 5.0
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

63%| 796/1261 [35:28<20:43, 2.67s/it]

The times for each task are: [0.695025, 0.950467, 1.407252, 0.361649, 0.493335] with:

Minimum: 0.361649 Maximum: 1.407252 Average: 0.7815 seconds

```
Number of cars 2
bbox:car_number ((940, 400), (1122, 539)) : 1
The minimum distance from car: 3 is 12.083045973594572
bbox:car_number ((913, 457), (927, 462)) : 2
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

63%| 797/1261 [35:31<20:40, 2.67s/it]

The times for each task are: [0.861871, 0.617839, 1.414942, 0.460037, 0.341529] with:

Minimum: 0.341529 Maximum: 1.414942 Average: 0.7392 seconds

```
Number of cars 1
bbox:car_number ((955, 400), (1118, 539)) : 1
The minimum distance from car: 3 is 5.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

63%| 798/1261 [35:34<20:38, 2.67s/it]

The times for each task are: [0.632539, 0.877994, 1.627217, 0.299561, 0.463174] with:

Minimum: 0.299561 Maximum: 1.627217 Average: 0.7801 seconds

```
Number of cars 1
bbox:car_number ((958, 400), (1118, 552)) : 1
The minimum distance from car: 3 is 7.280109889280518
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

63%| 799/1261 [35:36<20:35, 2.67s/it]

The times for each task are: [0.895384, 0.61933, 0.30813, 1.628318, 0.393337] with:

Minimum: 0.30813 Maximum: 1.628318 Average: 0.7689 seconds

```
Number of cars 1
bbox:car_number ((955, 400), (1122, 537)) : 1
The minimum distance from car: 3 is 8.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

63%| 800/1261 [35:39<20:32, 2.67s/it]

The times for each task are: [0.977452, 1.380546, 0.607531, 0.294384, 0.464397] with:

Minimum: 0.294384 Maximum: 1.380546 Average: 0.7449 seconds

```
Number of cars 1
bbox:car_number ((952, 419), (1122, 529)) : 1
The minimum distance from car: 3 is 6.082762530298219
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 801/1261 [35:41<20:29, 2.67s/it]

The times for each task are: [0.582159, 0.921688, 1.419347, 0.544412, 0.298892] with:

Minimum: 0.298892 Maximum: 1.419347 Average: 0.7533 seconds

```
Number of cars 2
bbox:car_number ((951, 415), (1122, 535)) : 1
The minimum distance from car: 3 is 1.4142135623730951
bbox:car_number ((940, 454), (942, 462)) : 2
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 802/1261 [35:44<20:27, 2.67s/it]

The times for each task are: [0.657849, 1.385842, 1.004849, 0.311007, 0.444524] with:

Minimum: 0.311007 Maximum: 1.385842 Average: 0.7608 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1093, 537)) : 1
The minimum distance from car: 3 is 21.18962010041709
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 803/1261 [35:46<20:24, 2.67s/it]

The times for each task are: [0.659841, 0.873577, 1.451817, 0.300149, 0.388374] with:

Minimum: 0.300149 Maximum: 1.451817 Average: 0.7348 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1107, 535)) : 1
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 804/1261 [35:49<20:21, 2.67s/it]

The times for each task are: [1.041127, 1.426587, 0.616438, 0.309709, 0.464443] with:

Minimum: 0.309709 Maximum: 1.426587 Average: 0.7717 seconds

```
Number of cars 1
bbox:car_number ((940, 415), (1107, 529)) : 1
The minimum distance from car: 3 is 5.0
totalCars: 6
```

```
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 805/1261 [35:52<20:19, 2.67s/it]

The times for each task are: [0.930517, 0.651969, 1.622591, 0.443946, 0.285669] with:

Minimum: 0.285669 Maximum: 1.622591 Average: 0.7869 seconds

```
Number of cars 1
bbox:car_number ((932, 400), (1107, 535)) : 1
The minimum distance from car: 3 is 6.4031242374328485
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 806/1261 [35:54<20:16, 2.67s/it]

The times for each task are: [0.982754, 1.379014, 0.624661, 0.357695, 0.465522] with:

Minimum: 0.357695 Maximum: 1.379014 Average: 0.7619 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1107, 539)) : 1
The minimum distance from car: 3 is 9.219544457292887
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 807/1261 [35:57<20:13, 2.67s/it]

The times for each task are: [0.719058, 0.930018, 1.461375, 0.543278, 0.356764] with:

Minimum: 0.356764 Maximum: 1.461375 Average: 0.8021 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1106, 539)) : 1
The minimum distance from car: 3 is 5.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
64%| 808/1261 [35:59<20:10, 2.67s/it]
```

```
The times for each task are: [0.664218, 0.890426, 1.483575, 0.46484, 0.299337] with:
```

```
Minimum: 0.299337 Maximum: 1.483575 Average: 0.7605 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (1106, 539)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
64%| 809/1261 [36:02<20:08, 2.67s/it]
```

```
The times for each task are: [0.887961, 0.615882, 0.397097, 0.32371, 1.702352] with:
```

```
Minimum: 0.32371 Maximum: 1.702352 Average: 0.7854 seconds
```

```
Number of cars 1
bbox:car_number ((940, 400), (1105, 529)) : 1
The minimum distance from car: 3 is 8.602325267042627
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
64%| 810/1261 [36:05<20:05, 2.67s/it]
```

```
The times for each task are: [0.677853, 0.898326, 1.486943, 0.439747, 0.310627] with:
```

```
Minimum: 0.310627 Maximum: 1.486943 Average: 0.7627 seconds
```

```
Number of cars 1
bbox:car_number ((940, 400), (1092, 535)) : 1
The minimum distance from car: 3 is 6.708203932499369
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
```

Number of processes used: 3

64%| 811/1261 [36:07<20:02, 2.67s/it]

The times for each task are: [0.842156, 0.609957, 0.39963, 0.317731, 1.451411] with:

Minimum: 0.317731 Maximum: 1.451411 Average: 0.7242 seconds

Number of cars 1

bbox:car_number ((927, 400), (1092, 535)) : 1

The minimum distance from car: 3 is 7.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

64%| 812/1261 [36:10<20:00, 2.67s/it]

The times for each task are: [0.865059, 1.701042, 0.667762, 0.39953, 0.451605] with:

Minimum: 0.39953 Maximum: 1.701042 Average: 0.817 seconds

Number of cars 2

bbox:car_number ((951, 400), (1092, 535)) : 1

The minimum distance from car: 3 is 12.0

bbox:car_number ((940, 460), (942, 507)) : 2

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

64%| 813/1261 [36:13<19:57, 2.67s/it]

The times for each task are: [1.466136, 0.64483, 0.926038, 0.289931, 0.400357] with:

Minimum: 0.289931 Maximum: 1.466136 Average: 0.7455 seconds

Number of cars 1

bbox:car_number ((925, 400), (1087, 529)) : 1

The minimum distance from car: 3 is 15.297058540778355

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 814/1261 [36:15<19:54, 2.67s/it]

The times for each task are: [0.629988, 0.856516, 1.39592, 0.309306, 0.414404] with:

Minimum: 0.309306 Maximum: 1.39592 Average: 0.7212 seconds

Number of cars 1

bbox:car_number ((927, 400), (1087, 535)) : 1

The minimum distance from car: 3 is 3.1622776601683795

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 815/1261 [36:18<19:52, 2.67s/it]

The times for each task are: [1.04419, 0.684051, 1.445833, 0.475471, 0.357786] with:

Minimum: 0.357786 Maximum: 1.445833 Average: 0.8015 seconds

Number of cars 1

bbox:car_number ((925, 400), (1087, 529)) : 1

The minimum distance from car: 3 is 3.1622776601683795

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 816/1261 [36:21<19:49, 2.67s/it]

The times for each task are: [0.89636, 0.679343, 1.634802, 0.30812, 0.424345] with:

Minimum: 0.30812 Maximum: 1.634802 Average: 0.7886 seconds

Number of cars 1

bbox:car_number ((940, 400), (1077, 529)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 817/1261 [36:23<19:46, 2.67s/it]

The times for each task are: [0.964022, 1.598886, 0.600837, 0.367224, 0.411635] with:

Minimum: 0.367224 Maximum: 1.598886 Average: 0.7885 seconds

Number of cars 1

bbox:car_number ((925, 400), (1086, 535)) : 1

The minimum distance from car: 3 is 4.242640687119285

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 818/1261 [36:26<19:44, 2.67s/it]

The times for each task are: [0.885209, 0.58059, 0.337963, 1.470237, 0.48826] with:

Minimum: 0.337963 Maximum: 1.470237 Average: 0.7525 seconds

Number of cars 1

bbox:car_number ((927, 400), (1086, 529)) : 1

The minimum distance from car: 3 is 3.1622776601683795

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 819/1261 [36:29<19:41, 2.67s/it]

The times for each task are: [0.972349, 0.555211, 1.566065, 0.344527, 0.450068] with:

Minimum: 0.344527 Maximum: 1.566065 Average: 0.7776 seconds

Number of cars 1

bbox:car_number ((927, 400), (1079, 529)) : 1

The minimum distance from car: 3 is 3.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 820/1261 [36:32<19:38, 2.67s/it]

The times for each task are: [0.596879, 0.978068, 0.548935, 0.389903, 1.609467] with:

Minimum: 0.389903 Maximum: 1.609467 Average: 0.8247 seconds

Number of cars 1

bbox:car_number ((925, 400), (1079, 539)) : 1

The minimum distance from car: 3 is 5.0990195135927845

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 821/1261 [36:34<19:36, 2.67s/it]

The times for each task are: [0.88994, 0.686187, 1.391827, 0.386595, 0.400639] with:

Minimum: 0.386595 Maximum: 1.391827 Average: 0.751 seconds

Number of cars 2

bbox:car_number ((929, 400), (1077, 518)) : 1

The minimum distance from car: 3 is 10.04987562112089

bbox:car_number ((927, 460), (927, 462)) : 2

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 822/1261 [36:37<19:33, 2.67s/it]

The times for each task are: [0.626934, 1.095524, 0.412449, 1.619916, 0.362468] with:

Minimum: 0.362468 Maximum: 1.619916 Average: 0.8235 seconds

Number of cars 1

bbox:car_number ((927, 400), (1077, 522)) : 1

The minimum distance from car: 3 is 2.23606797749979

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 823/1261 [36:39<19:30, 2.67s/it]

The times for each task are: [0.864992, 1.37408, 0.700251, 0.357001, 0.442336] with:

Minimum: 0.357001 Maximum: 1.37408 Average: 0.7477 seconds

Number of cars 1

bbox:car_number ((927, 400), (1077, 529)) : 1

The minimum distance from car: 3 is 3.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 824/1261 [36:42<19:27, 2.67s/it]

The times for each task are: [0.99042, 0.613023, 0.326115, 0.465194, 1.710262] with:

Minimum: 0.326115 Maximum: 1.710262 Average: 0.821 seconds

Number of cars 1

bbox:car_number ((913, 400), (1067, 522)) : 1

The minimum distance from car: 3 is 12.36931687685298

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 825/1261 [36:44<19:25, 2.67s/it]

The times for each task are: [0.900319, 0.615678, 1.589474, 0.399138, 0.280173] with:

Minimum: 0.280173 Maximum: 1.589474 Average: 0.757 seconds

Number of cars 1

bbox:car_number ((913, 400), (1056, 518)) : 1

The minimum distance from car: 3 is 6.324555320336759

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

66%| 826/1261 [36:47<19:22, 2.67s/it]

The times for each task are: [0.893928, 0.660193, 1.596787, 0.422186, 0.327425] with:

Minimum: 0.327425 Maximum: 1.596787 Average: 0.7801 seconds

Number of cars 1

bbox:car_number ((927, 400), (1052, 512)) : 1

The minimum distance from car: 3 is 5.830951894845301

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

66%| 827/1261 [36:50<19:19, 2.67s/it]

The times for each task are: [0.653499, 0.871451, 1.399778, 0.428947, 0.525546] with:

Minimum: 0.428947 Maximum: 1.399778 Average: 0.7758 seconds

Number of cars 1

bbox:car_number ((895, 400), (1067, 522)) : 1

The minimum distance from car: 3 is 9.433981132056603

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

66%| 828/1261 [36:52<19:17, 2.67s/it]

The times for each task are: [0.853315, 0.710748, 1.447078, 0.445807, 0.336086] with:

Minimum: 0.336086 Maximum: 1.447078 Average: 0.7586 seconds

Number of cars 1

bbox:car_number ((913, 400), (1062, 535)) : 1

The minimum distance from car: 3 is 8.48528137423857

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

66%| 829/1261 [36:55<19:14, 2.67s/it]

The times for each task are: [1.34393, 0.636761, 1.022727, 0.300362, 0.407261] with:

Minimum: 0.300362 Maximum: 1.34393 Average: 0.7422 seconds

Number of cars 1
bbox:car_number ((913, 400), (1056, 529)) : 1
The minimum distance from car: 3 is 4.242640687119285
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

66%| 830/1261 [36:57<19:11, 2.67s/it]

The times for each task are: [0.913236, 1.468079, 0.686195, 0.353781, 0.414167] with:

Minimum: 0.353781 Maximum: 1.468079 Average: 0.7671 seconds

Number of cars 1
bbox:car_number ((905, 400), (1056, 529)) : 1
The minimum distance from car: 3 is 4.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

66%| 831/1261 [37:00<19:08, 2.67s/it]

The times for each task are: [1.360193, 0.958963, 0.705246, 0.400285, 0.330796] with:

Minimum: 0.330796 Maximum: 1.360193 Average: 0.7511 seconds

Number of cars 1
bbox:car_number ((896, 400), (1052, 529)) : 1
The minimum distance from car: 3 is 6.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

66%| 832/1261 [37:02<19:06, 2.67s/it]

The times for each task are: [0.669261, 1.600254, 0.909716, 0.353086, 0.395425] with:

Minimum: 0.353086 Maximum: 1.600254 Average: 0.7855 seconds

```
Number of cars 1
bbox:car_number ((896, 419), (1025, 529)) : 1
The minimum distance from car: 3 is 17.204650534085253
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

66%| 833/1261 [37:05<19:03, 2.67s/it]

The times for each task are: [0.971408, 0.627196, 1.44788, 0.418631, 0.317742] with:

Minimum: 0.317742 Maximum: 1.44788 Average: 0.7566 seconds

```
Number of cars 1
bbox:car_number ((896, 415), (1047, 529)) : 1
The minimum distance from car: 3 is 11.180339887498949
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

66%| 834/1261 [37:07<19:00, 2.67s/it]

The times for each task are: [1.013319, 1.401588, 0.707482, 0.417658, 0.540152] with:

Minimum: 0.417658 Maximum: 1.401588 Average: 0.816 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1025, 529)) : 1
The minimum distance from car: 3 is 13.601470508735444
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

66%| 835/1261 [37:10<18:57, 2.67s/it]

The times for each task are: [0.886173, 1.462702, 0.673484, 0.439571, 0.388026] with:

Minimum: 0.388026 Maximum: 1.462702 Average: 0.77 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1052, 529)) : 1
The minimum distance from car: 3 is 14.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

66%| 836/1261 [37:13<18:55, 2.67s/it]

The times for each task are: [0.910658, 0.710168, 1.660206, 0.344135, 0.466381] with:
Minimum: 0.344135 Maximum: 1.660206 Average: 0.8183 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (1048, 522)) : 1
The minimum distance from car: 3 is 5.830951894845301
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

66%| 837/1261 [37:16<18:52, 2.67s/it]

The times for each task are: [1.373945, 0.613554, 1.012155, 0.409852, 0.308651] with:
Minimum: 0.308651 Maximum: 1.373945 Average: 0.7436 seconds

```
Number of cars 1
bbox:car_number ((910, 419), (1047, 518)) : 1
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

66%| 838/1261 [37:18<18:50, 2.67s/it]

The times for each task are: [0.892283, 1.389383, 0.466753, 0.689681, 0.297049] with:
Minimum: 0.297049 Maximum: 1.389383 Average: 0.747 seconds

Number of cars 1

```
bbox:car_number ((896, 423), (1047, 518)) : 1
The minimum distance from car: 3 is 7.280109889280518
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

67%| 839/1261 [37:21<18:47, 2.67s/it]

The times for each task are: [1.443606, 0.619864, 0.963275, 0.32998, 0.403957] with:

Minimum: 0.32998 Maximum: 1.443606 Average: 0.7521 seconds

```
Number of cars 1
bbox:car_number ((896, 419), (1025, 516)) : 1
The minimum distance from car: 3 is 11.40175425099138
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

67%| 840/1261 [37:24<18:44, 2.67s/it]

The times for each task are: [0.617136, 1.424788, 0.896583, 0.4926, 0.374212] with:

Minimum: 0.374212 Maximum: 1.424788 Average: 0.7611 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 8.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

67%| 841/1261 [37:26<18:41, 2.67s/it]

The times for each task are: [0.888689, 0.734381, 1.500483, 0.451314, 0.354595] with:

Minimum: 0.354595 Maximum: 1.500483 Average: 0.7859 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (1025, 512)) : 1
```

```
The minimum distance from car: 3 is 3.1622776601683795
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

67%| 842/1261 [37:28<18:39, 2.67s/it]

The times for each task are: [0.948673, 1.517375, 0.636876, 0.326542, 0.458982] with:

Minimum: 0.326542 Maximum: 1.517375 Average: 0.7777 seconds

```
Number of cars 1
bbox:car_number ((895, 400), (1025, 512)) : 1
The minimum distance from car: 3 is 1.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

67%| 843/1261 [37:31<18:36, 2.67s/it]

The times for each task are: [0.629821, 1.40799, 1.044828, 0.286095, 0.462203] with:

Minimum: 0.286095 Maximum: 1.40799 Average: 0.7662 seconds

```
Number of cars 1
bbox:car_number ((896, 419), (1025, 512)) : 1
The minimum distance from car: 3 is 9.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

67%| 844/1261 [37:34<18:33, 2.67s/it]

The times for each task are: [0.62939, 1.584455, 0.886829, 0.479801, 0.344165] with:

Minimum: 0.344165 Maximum: 1.584455 Average: 0.7849 seconds

```
Number of cars 1
bbox:car_number ((895, 400), (1025, 512)) : 1
The minimum distance from car: 3 is 9.0
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 845/1261 [37:37<18:31, 2.67s/it]

The times for each task are: [0.631075, 0.91782, 1.411524, 0.383334, 0.442765] with:

Minimum: 0.383334 Maximum: 1.411524 Average: 0.7573 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 3.1622776601683795
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 846/1261 [37:39<18:28, 2.67s/it]

The times for each task are: [0.910044, 1.39927, 0.615738, 0.41221, 0.301711] with:

Minimum: 0.301711 Maximum: 1.39927 Average: 0.7278 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1025, 498)) : 1
The minimum distance from car: 3 is 10.04987562112089
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 847/1261 [37:42<18:25, 2.67s/it]

The times for each task are: [0.646744, 0.922708, 1.492777, 0.377774, 0.424255] with:

Minimum: 0.377774 Maximum: 1.492777 Average: 0.7729 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1024, 512)) : 1
The minimum distance from car: 3 is 7.0
totalCars: 6
```

```
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 848/1261 [37:44<18:23, 2.67s/it]

The times for each task are: [0.93641, 0.649591, 1.641752, 0.321348, 0.426739] with:

Minimum: 0.321348 Maximum: 1.641752 Average: 0.7952 seconds

```
Number of cars 1
bbox:car_number ((886, 400), (1025, 512)) : 1
The minimum distance from car: 3 is 5.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 849/1261 [37:47<18:20, 2.67s/it]

The times for each task are: [0.974724, 0.60636, 1.591217, 0.374583, 0.470922] with:

Minimum: 0.374583 Maximum: 1.591217 Average: 0.8036 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 5.830951894845301
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 850/1261 [37:50<18:17, 2.67s/it]

The times for each task are: [0.921416, 0.665081, 1.496892, 0.36142, 0.402416] with:

Minimum: 0.36142 Maximum: 1.496892 Average: 0.7694 seconds

```
Number of cars 1
bbox:car_number ((882, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
67%| 851/1261 [37:52<18:15, 2.67s/it]
```

```
The times for each task are: [0.904829, 1.410482, 0.617522, 0.346905, 0.406559] with:
```

```
Minimum: 0.346905 Maximum: 1.410482 Average: 0.7373 seconds
```

```
Number of cars 1
bbox:car_number ((882, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
68%| 852/1261 [37:55<18:12, 2.67s/it]
```

```
The times for each task are: [0.983632, 0.643851, 1.682514, 0.405424, 0.307945] with:
```

```
Minimum: 0.307945 Maximum: 1.682514 Average: 0.8047 seconds
```

```
Number of cars 1
bbox:car_number ((886, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
68%| 853/1261 [37:58<18:09, 2.67s/it]
```

```
The times for each task are: [0.912286, 0.66576, 1.655089, 0.426289, 0.382372] with:
```

```
Minimum: 0.382372 Maximum: 1.655089 Average: 0.8084 seconds
```

```
Number of cars 1
bbox:car_number ((886, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
```

Number of processes used: 3

68%| 854/1261 [38:01<18:07, 2.67s/it]

The times for each task are: [0.697047, 0.918424, 0.462738, 0.392133, 1.59169] with:

Minimum: 0.392133 Maximum: 1.59169 Average: 0.8124 seconds

Number of cars 1

bbox:car_number ((880, 400), (1025, 518)) : 1

The minimum distance from car: 3 is 3.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 855/1261 [38:03<18:04, 2.67s/it]

The times for each task are: [0.898408, 0.591961, 0.4299, 1.460874, 0.394987] with:

Minimum: 0.394987 Maximum: 1.460874 Average: 0.7552 seconds

Number of cars 1

bbox:car_number ((886, 400), (1024, 518)) : 1

The minimum distance from car: 3 is 3.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 856/1261 [38:06<18:01, 2.67s/it]

The times for each task are: [0.925832, 0.622736, 1.489458, 0.502505, 0.386848] with:

Minimum: 0.386848 Maximum: 1.489458 Average: 0.7855 seconds

Number of cars 1

bbox:car_number ((875, 400), (1032, 522)) : 1

The minimum distance from car: 3 is 2.8284271247461903

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 857/1261 [38:08<17:58, 2.67s/it]

The times for each task are: [0.640399, 0.996968, 1.41574, 0.492209, 0.293502] with:

Minimum: 0.293502 Maximum: 1.41574 Average: 0.7678 seconds

Number of cars 1

bbox:car_number ((875, 400), (1047, 522)) : 1

The minimum distance from car: 3 is 8.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 858/1261 [38:11<17:56, 2.67s/it]

The times for each task are: [0.653665, 0.931804, 1.42712, 0.309903, 0.384148] with:

Minimum: 0.309903 Maximum: 1.42712 Average: 0.7413 seconds

Number of cars 1

bbox:car_number ((875, 400), (1047, 522)) : 1

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 859/1261 [38:14<17:53, 2.67s/it]

The times for each task are: [0.732982, 1.022405, 1.403813, 0.381385, 0.498512] with:

Minimum: 0.381385 Maximum: 1.403813 Average: 0.8078 seconds

Number of cars 1

bbox:car_number ((875, 400), (1047, 522)) : 1

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 860/1261 [38:16<17:51, 2.67s/it]

The times for each task are: [0.614662, 1.059952, 1.38718, 0.403812, 0.334301] with:

Minimum: 0.334301 Maximum: 1.38718 Average: 0.76 seconds

Number of cars 1

bbox:car_number ((875, 400), (1048, 518)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 861/1261 [38:19<17:48, 2.67s/it]

The times for each task are: [0.722772, 1.415708, 0.891756, 0.314017, 0.533684] with:

Minimum: 0.314017 Maximum: 1.415708 Average: 0.7756 seconds

Number of cars 1

bbox:car_number ((865, 400), (1048, 512)) : 1

The minimum distance from car: 3 is 5.830951894845301

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 862/1261 [38:22<17:45, 2.67s/it]

The times for each task are: [0.894222, 1.443771, 0.652328, 0.340446, 0.40126] with:

Minimum: 0.340446 Maximum: 1.443771 Average: 0.7464 seconds

Number of cars 1

bbox:car_number ((859, 400), (1047, 518)) : 1

The minimum distance from car: 3 is 4.242640687119285

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 863/1261 [38:24<17:42, 2.67s/it]

The times for each task are: [0.868784, 0.576316, 1.394499, 0.392232, 0.346291] with:

Minimum: 0.346291 Maximum: 1.394499 Average: 0.7156 seconds

Number of cars 1

bbox:car_number ((859, 400), (1048, 518)) : 1

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

69%| 864/1261 [38:27<17:40, 2.67s/it]

The times for each task are: [0.633734, 0.934314, 1.435099, 0.293588, 0.504368] with:

Minimum: 0.293588 Maximum: 1.435099 Average: 0.7602 seconds

Number of cars 1

bbox:car_number ((859, 400), (1048, 518)) : 1

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

69%| 865/1261 [38:30<17:37, 2.67s/it]

The times for each task are: [0.718049, 0.898705, 1.541882, 0.445335, 0.285586] with:

Minimum: 0.285586 Maximum: 1.541882 Average: 0.7779 seconds

Number of cars 1

bbox:car_number ((859, 400), (1052, 518)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

69%| 866/1261 [38:32<17:34, 2.67s/it]

The times for each task are: [0.953393, 0.751913, 1.368124, 0.399959, 0.337767] with:

Minimum: 0.337767 Maximum: 1.368124 Average: 0.7622 seconds

Number of cars 1
bbox:car_number ((865, 400), (1052, 518)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

69%| 867/1261 [38:35<17:32, 2.67s/it]

The times for each task are: [0.927536, 0.672015, 1.41837, 0.404566, 0.307012] with:

Minimum: 0.307012 Maximum: 1.41837 Average: 0.7459 seconds

Number of cars 1
bbox:car_number ((865, 400), (1052, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

69%| 868/1261 [38:37<17:29, 2.67s/it]

The times for each task are: [0.891896, 0.578441, 1.401756, 0.30099, 0.533522] with:

Minimum: 0.30099 Maximum: 1.401756 Average: 0.7413 seconds

Number of cars 1
bbox:car_number ((865, 400), (1056, 518)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

69%| 869/1261 [38:40<17:26, 2.67s/it]

The times for each task are: [0.837773, 1.387034, 0.659157, 0.306129, 0.40016] with:

Minimum: 0.306129 Maximum: 1.387034 Average: 0.7181 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 518)) : 1
The minimum distance from car: 3 is 4.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

69%| 870/1261 [38:42<17:23, 2.67s/it]

The times for each task are: [0.588825, 1.366743, 1.059998, 0.290426, 0.382586] with:

Minimum: 0.290426 Maximum: 1.366743 Average: 0.7377 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1056, 518)) : 1
The minimum distance from car: 3 is 7.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

69%| 871/1261 [38:45<17:21, 2.67s/it]

The times for each task are: [1.355898, 0.726821, 0.90751, 0.349641, 0.504037] with:

Minimum: 0.349641 Maximum: 1.355898 Average: 0.7688 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1056, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

69%| 872/1261 [38:48<17:18, 2.67s/it]

The times for each task are: [0.615077, 1.353648, 0.909319, 0.406685, 0.283497] with:

Minimum: 0.283497 Maximum: 1.353648 Average: 0.7136 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1056, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

69%| 873/1261 [38:50<17:15, 2.67s/it]

The times for each task are: [0.914328, 0.583476, 1.65763, 0.309856, 0.511027] with:
Minimum: 0.309856 Maximum: 1.65763 Average: 0.7953 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1067, 518)) : 1
The minimum distance from car: 3 is 6.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

69%| 874/1261 [38:53<17:13, 2.67s/it]

The times for each task are: [0.869596, 0.714082, 1.46531, 0.429507, 0.576844] with:
Minimum: 0.429507 Maximum: 1.46531 Average: 0.8111 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1067, 516)) : 1
The minimum distance from car: 3 is 2.23606797749979
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

69%| 875/1261 [38:56<17:10, 2.67s/it]

The times for each task are: [0.668149, 1.030145, 1.655297, 0.355969, 0.400756] with:
Minimum: 0.355969 Maximum: 1.655297 Average: 0.8221 seconds

Number of cars 1

```
bbox:car_number ((856, 400), (1067, 516)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

69%| 876/1261 [38:59<17:08, 2.67s/it]

The times for each task are: [0.940942, 0.624055, 1.773509, 0.413848, 0.31066] with:

Minimum: 0.31066 Maximum: 1.773509 Average: 0.8126 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

70%| 877/1261 [39:01<17:05, 2.67s/it]

The times for each task are: [0.942308, 0.653947, 1.472898, 0.378068, 0.445618] with:

Minimum: 0.378068 Maximum: 1.472898 Average: 0.7786 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

70%| 878/1261 [39:04<17:02, 2.67s/it]

The times for each task are: [0.859996, 1.586739, 0.750549, 0.417543, 0.314274] with:

Minimum: 0.314274 Maximum: 1.586739 Average: 0.7858 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 516)) : 1
```

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

70%| 879/1261 [39:07<17:00, 2.67s/it]

The times for each task are: [0.897145, 1.649209, 0.624317, 0.409219, 0.339962] with:

Minimum: 0.339962 Maximum: 1.649209 Average: 0.784 seconds

Number of cars 1

bbox:car_number ((856, 400), (1067, 516)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

70%| 880/1261 [39:10<16:57, 2.67s/it]

The times for each task are: [1.347503, 0.913037, 0.70457, 0.46646, 0.314026] with:

Minimum: 0.314026 Maximum: 1.347503 Average: 0.7491 seconds

Number of cars 1

bbox:car_number ((859, 400), (1070, 512)) : 1

The minimum distance from car: 3 is 3.605551275463989

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

70%| 881/1261 [39:13<16:54, 2.67s/it]

The times for each task are: [0.912473, 0.623051, 0.412027, 1.533342, 0.295007] with:

Minimum: 0.295007 Maximum: 1.533342 Average: 0.7552 seconds

Number of cars 1

bbox:car_number ((856, 400), (1070, 512)) : 1

The minimum distance from car: 3 is 1.0

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 882/1261 [39:15<16:52, 2.67s/it]

The times for each task are: [1.359619, 0.872255, 0.727689, 0.308745, 0.393263] with:

Minimum: 0.308745 Maximum: 1.359619 Average: 0.7323 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 512)) : 1
The minimum distance from car: 3 is 1.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 883/1261 [39:18<16:49, 2.67s/it]

The times for each task are: [0.684315, 0.87438, 1.441542, 0.351609, 0.413345] with:

Minimum: 0.351609 Maximum: 1.441542 Average: 0.753 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 518)) : 1
The minimum distance from car: 3 is 3.1622776601683795
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 884/1261 [39:20<16:46, 2.67s/it]

The times for each task are: [0.675686, 0.824573, 1.4242, 0.329829, 0.499647] with:

Minimum: 0.329829 Maximum: 1.4242 Average: 0.7508 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 512)) : 1
The minimum distance from car: 3 is 3.1622776601683795
totalCars: 6
```

```
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 885/1261 [39:23<16:44, 2.67s/it]

The times for each task are: [0.715662, 1.602708, 1.124667, 0.496083, 0.331683] with:

Minimum: 0.331683 Maximum: 1.602708 Average: 0.8542 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 886/1261 [39:26<16:41, 2.67s/it]

The times for each task are: [0.954319, 0.65041, 1.488342, 0.477212, 0.318871] with:

Minimum: 0.318871 Maximum: 1.488342 Average: 0.7778 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 1.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 887/1261 [39:28<16:38, 2.67s/it]

The times for each task are: [0.686458, 0.951862, 1.474604, 0.476377, 0.308826] with:

Minimum: 0.308826 Maximum: 1.474604 Average: 0.7796 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 518)) : 1
The minimum distance from car: 3 is 1.4142135623730951
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
70%| 888/1261 [39:31<16:36, 2.67s/it]
```

```
The times for each task are: [0.924225, 0.648107, 0.407998, 0.311311, 1.692299] with:
```

```
Minimum: 0.311311 Maximum: 1.692299 Average: 0.7968 seconds
```

```
Number of cars 1
bbox:car_number ((856, 400), (1077, 518)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
70%| 889/1261 [39:34<16:33, 2.67s/it]
```

```
The times for each task are: [1.012658, 0.660515, 1.569227, 0.412252, 0.305345] with:
```

```
Minimum: 0.305345 Maximum: 1.569227 Average: 0.792 seconds
```

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 2.23606797749979
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
71%| 890/1261 [39:36<16:30, 2.67s/it]
```

```
The times for each task are: [0.639108, 0.931926, 1.448533, 0.422197, 0.304881] with:
```

```
Minimum: 0.304881 Maximum: 1.448533 Average: 0.7493 seconds
```

```
Number of cars 1
bbox:car_number ((865, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
```

Number of processes used: 3

71%| 891/1261 [39:39<16:28, 2.67s/it]

The times for each task are: [1.467655, 0.936945, 0.746438, 0.437258, 0.337013] with:

Minimum: 0.337013 Maximum: 1.467655 Average: 0.7851 seconds

Number of cars 1

bbox:car_number ((859, 400), (1079, 516)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 892/1261 [39:42<16:25, 2.67s/it]

The times for each task are: [0.670645, 1.072426, 1.46533, 0.416006, 0.457946] with:

Minimum: 0.416006 Maximum: 1.46533 Average: 0.8165 seconds

Number of cars 1

bbox:car_number ((859, 400), (1077, 516)) : 1

The minimum distance from car: 3 is 1.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 893/1261 [39:45<16:23, 2.67s/it]

The times for each task are: [0.642533, 1.00293, 1.666507, 0.448161, 0.361509] with:

Minimum: 0.361509 Maximum: 1.666507 Average: 0.8243 seconds

Number of cars 1

bbox:car_number ((865, 400), (1079, 516)) : 1

The minimum distance from car: 3 is 4.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 894/1261 [39:47<16:20, 2.67s/it]

The times for each task are: [0.909364, 0.583967, 1.437785, 0.410389, 0.303229] with:

Minimum: 0.303229 Maximum: 1.437785 Average: 0.7289 seconds

Number of cars 1
bbox:car_number ((859, 400), (1086, 516)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

71%| 895/1261 [39:50<16:17, 2.67s/it]

The times for each task are: [0.949394, 0.664009, 1.513609, 0.409483, 0.299905] with:

Minimum: 0.299905 Maximum: 1.513609 Average: 0.7673 seconds

Number of cars 1
bbox:car_number ((865, 400), (1086, 516)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

71%| 896/1261 [39:53<16:14, 2.67s/it]

The times for each task are: [0.909908, 0.755297, 1.398537, 0.405004, 0.357669] with:

Minimum: 0.357669 Maximum: 1.398537 Average: 0.7653 seconds

Number of cars 1
bbox:car_number ((859, 400), (1087, 516)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

71%| 897/1261 [39:55<16:12, 2.67s/it]

The times for each task are: [0.579274, 0.948759, 1.438251, 0.532772, 0.356812] with:

Minimum: 0.356812 Maximum: 1.438251 Average: 0.7712 seconds

Number of cars 1

bbox:car_number ((850, 400), (1087, 516)) : 1

The minimum distance from car: 3 is 5.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 898/1261 [39:58<16:09, 2.67s/it]

The times for each task are: [1.016181, 1.543582, 0.711738, 0.435406, 0.320515] with:

Minimum: 0.320515 Maximum: 1.543582 Average: 0.8055 seconds

Number of cars 1

bbox:car_number ((865, 400), (1086, 512)) : 1

The minimum distance from car: 3 is 7.280109889280518

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 899/1261 [40:01<16:06, 2.67s/it]

The times for each task are: [0.645983, 0.905804, 1.524711, 0.324248, 0.4287] with:

Minimum: 0.324248 Maximum: 1.524711 Average: 0.7659 seconds

Number of cars 1

bbox:car_number ((850, 400), (1086, 516)) : 1

The minimum distance from car: 3 is 7.280109889280518

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 900/1261 [40:04<16:04, 2.67s/it]

The times for each task are: [0.939587, 0.716211, 1.777714, 0.470215, 0.369565] with:

Minimum: 0.369565 Maximum: 1.777714 Average: 0.8547 seconds

Number of cars 1

bbox:car_number ((850, 400), (1087, 516)) : 1

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71% | 901/1261 [40:06<16:01, 2.67s/it]

The times for each task are: [0.678212, 1.020171, 1.453994, 0.389241, 0.430468] with:

Minimum: 0.389241 Maximum: 1.453994 Average: 0.7944 seconds

Number of cars 1

bbox:car_number ((859, 400), (1087, 516)) : 1

The minimum distance from car: 3 is 5.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

72% | 902/1261 [40:09<15:59, 2.67s/it]

The times for each task are: [1.060342, 0.636511, 1.448577, 0.331789, 0.484776] with:

Minimum: 0.331789 Maximum: 1.448577 Average: 0.7924 seconds

Number of cars 1

bbox:car_number ((834, 400), (1087, 512)) : 1

The minimum distance from car: 3 is 13.152946437965905

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

72% | 903/1261 [40:12<15:56, 2.67s/it]

The times for each task are: [0.599926, 0.881216, 1.508175, 0.313339, 0.407719] with:

Minimum: 0.313339 Maximum: 1.508175 Average: 0.7421 seconds

Number of cars 1
bbox:car_number ((834, 400), (1093, 512)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3

72%| 904/1261 [40:14<15:53, 2.67s/it]

The times for each task are: [0.617159, 1.438032, 1.074664, 0.445705, 0.362997] with:

Minimum: 0.362997 Maximum: 1.438032 Average: 0.7877 seconds

Number of cars 1
bbox:car_number ((835, 400), (1087, 512)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3

72%| 905/1261 [40:17<15:51, 2.67s/it]

The times for each task are: [0.860778, 1.428138, 0.764905, 0.429328, 0.370492] with:

Minimum: 0.370492 Maximum: 1.428138 Average: 0.7707 seconds

Number of cars 1
bbox:car_number ((834, 400), (1093, 512)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3

72%| 906/1261 [40:20<15:48, 2.67s/it]

The times for each task are: [0.911381, 0.651614, 1.427639, 0.418074, 0.402864] with:

Minimum: 0.402864 Maximum: 1.427639 Average: 0.7623 seconds

```
Number of cars 1
bbox:car_number ((834, 400), (1093, 512)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

72%| 907/1261 [40:23<15:45, 2.67s/it]

The times for each task are: [1.378543, 0.743035, 0.842139, 0.422369, 0.416331] with:

Minimum: 0.416331 Maximum: 1.378543 Average: 0.7605 seconds

```
Number of cars 1
bbox:car_number ((834, 400), (1093, 512)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

72%| 908/1261 [40:25<15:43, 2.67s/it]

The times for each task are: [0.892011, 0.637812, 1.664271, 0.403886, 0.30937] with:

Minimum: 0.30937 Maximum: 1.664271 Average: 0.7815 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (963, 512)) : 1
bbox:car_number ((994, 400), (1093, 495)) : 2
The minimum distance from car: 1 is 8.94427190999916
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

72%| 909/1261 [40:28<15:40, 2.67s/it]

The times for each task are: [0.663672, 1.414194, 0.894877, 0.298464, 0.376194] with:

Minimum: 0.298464 Maximum: 1.414194 Average: 0.7295 seconds

Number of cars 1
bbox:car_number ((834, 400), (1106, 512)) : 1
The minimum distance from car: 3 is 7.0
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

72%| | 910/1261 [40:31<15:37, 2.67s/it]

The times for each task are: [0.638694, 0.9208, 1.431948, 0.415639, 0.290889] with:

Minimum: 0.290889 Maximum: 1.431948 Average: 0.7396 seconds

Number of cars 1
bbox:car_number ((834, 400), (1107, 512)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

72%| | 911/1261 [40:34<15:35, 2.67s/it]

The times for each task are: [0.8803, 1.364672, 0.673965, 0.421468, 0.326974] with:

Minimum: 0.326974 Maximum: 1.364672 Average: 0.7335 seconds

Number of cars 1
bbox:car_number ((835, 400), (1106, 507)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

72%| | 912/1261 [40:37<15:32, 2.67s/it]

The times for each task are: [0.662222, 1.509168, 1.153442, 0.393928, 0.451087] with:

Minimum: 0.393928 Maximum: 1.509168 Average: 0.834 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (963, 512)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((970, 400), (1106, 498)) : 2
The minimum distance from car: 1 is 5.385164807134504
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

72%| | 913/1261 [40:39<15:29, 2.67s/it]

The times for each task are: [0.959647, 0.648355, 0.365462, 1.727146, 0.41636] with:
Minimum: 0.365462 Maximum: 1.727146 Average: 0.8234 seconds

```
Number of cars 1
bbox:car_number ((834, 400), (1107, 512)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

72%| | 914/1261 [40:42<15:27, 2.67s/it]

The times for each task are: [0.658389, 0.914589, 1.460468, 0.324881, 0.430607] with:
Minimum: 0.324881 Maximum: 1.460468 Average: 0.7578 seconds

```
Number of cars 1
bbox:car_number ((834, 400), (1105, 516)) : 1
The minimum distance from car: 3 is 2.23606797749979
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

73%| | 915/1261 [40:45<15:24, 2.67s/it]

The times for each task are: [0.926562, 1.421781, 0.626412, 0.404706, 0.435084] with:

Minimum: 0.404706 Maximum: 1.421781 Average: 0.7629 seconds

Number of cars 2

bbox:car_number ((834, 400), (963, 512)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((989, 400), (1116, 498)) : 2

The minimum distance from car: 1 is 14.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

73%| 916/1261 [40:47<15:21, 2.67s/it]

The times for each task are: [0.925629, 0.640162, 1.46401, 0.41863, 0.302585] with:

Minimum: 0.302585 Maximum: 1.46401 Average: 0.7502 seconds

Number of cars 1

bbox:car_number ((834, 400), (1116, 512)) : 1

The minimum distance from car: 3 is 6.324555320336759

totalCars: 7

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1242, 518))]

Length of task list: 5

Number of processes used: 3

73%| 917/1261 [40:50<15:19, 2.67s/it]

The times for each task are: [0.65825, 0.902347, 1.681288, 0.490312, 0.350803] with:

Minimum: 0.350803 Maximum: 1.681288 Average: 0.8166 seconds

Number of cars 1

bbox:car_number ((834, 400), (1107, 512)) : 1

The minimum distance from car: 3 is 5.0

totalCars: 7

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1242, 518))]

Length of task list: 5

Number of processes used: 3

73%| | 918/1261 [40:53<15:16, 2.67s/it]

The times for each task are: [1.378897, 0.742234, 1.042722, 0.424913, 0.405974] with:

Minimum: 0.405974 Maximum: 1.378897 Average: 0.7989 seconds

Number of cars 1

bbox:car_number ((834, 400), (1116, 512)) : 1

The minimum distance from car: 3 is 5.0

totalCars: 7

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

73%| | 919/1261 [40:55<15:13, 2.67s/it]

The times for each task are: [1.087729, 0.703612, 1.410062, 0.442867, 0.40283] with:

Minimum: 0.40283 Maximum: 1.410062 Average: 0.8094 seconds

Number of cars 1

bbox:car_number ((834, 400), (1118, 512)) : 1

The minimum distance from car: 3 is 1.0

totalCars: 7

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

73%| | 920/1261 [40:58<15:11, 2.67s/it]

The times for each task are: [0.599476, 1.380326, 1.068162, 0.447583, 0.29926] with:

Minimum: 0.29926 Maximum: 1.380326 Average: 0.759 seconds

Number of cars 2

bbox:car_number ((834, 400), (963, 512)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((970, 400), (1116, 497)) : 2

The minimum distance from car: 1 is 9.055385138137417

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

73%| | 921/1261 [41:01<15:08, 2.67s/it]

The times for each task are: [0.681761, 0.919, 1.430844, 0.301725, 0.537633] with:

Minimum: 0.301725 Maximum: 1.430844 Average: 0.7742 seconds

Number of cars 2

bbox:car_number ((834, 400), (963, 512)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((967, 400), (1118, 497)) : 2

The minimum distance from car: 1 is 1.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

73%| | 922/1261 [41:04<15:05, 2.67s/it]

The times for each task are: [0.878746, 1.433604, 0.749388, 0.453185, 0.383194] with:

Minimum: 0.383194 Maximum: 1.433604 Average: 0.7796 seconds

Number of cars 2

bbox:car_number ((834, 400), (954, 512)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((970, 400), (1116, 497)) : 2

The minimum distance from car: 1 is 1.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

73%| | 923/1261 [41:06<15:03, 2.67s/it]

The times for each task are: [0.767154, 0.95947, 1.527807, 0.473171, 0.343111] with:

Minimum: 0.343111 Maximum: 1.527807 Average: 0.8141 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (963, 512)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((970, 400), (1118, 498)) : 2
The minimum distance from car: 1 is 1.4142135623730951
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

73%| | 924/1261 [41:09<15:00, 2.67s/it]

The times for each task are: [0.930314, 1.378068, 0.659063, 0.316116, 0.442029] with:

Minimum: 0.316116 Maximum: 1.378068 Average: 0.7451 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (944, 512)) : 1
The minimum distance from car: 1 is 9.0
bbox:car_number ((970, 400), (1118, 495)) : 2
The minimum distance from car: 1 is 2.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

73%| | 925/1261 [41:12<14:58, 2.67s/it]

The times for each task are: [0.668393, 1.515642, 0.961095, 0.407044, 0.44161] with:

Minimum: 0.407044 Maximum: 1.515642 Average: 0.7988 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (953, 512)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((970, 400), (1124, 498)) : 2
The minimum distance from car: 1 is 3.605551275463989
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
```

```
Length of task list: 5
Number of processes used: 3
```

```
73%| | 926/1261 [41:15<14:55, 2.67s/it]
```

```
The times for each task are: [0.935022, 0.655036, 1.455449, 0.404072, 0.313499] with:
```

```
Minimum: 0.313499 Maximum: 1.455449 Average: 0.7526 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (944, 512)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((970, 400), (1122, 507)) : 2
The minimum distance from car: 1 is 4.123105625617661
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
74%| | 927/1261 [41:17<14:52, 2.67s/it]
```

```
The times for each task are: [0.919783, 0.685825, 1.670983, 0.43916, 0.364436] with:
```

```
Minimum: 0.364436 Maximum: 1.670983 Average: 0.816 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (944, 512)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((970, 400), (1124, 498)) : 2
The minimum distance from car: 1 is 4.123105625617661
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
74%| | 928/1261 [41:20<14:50, 2.67s/it]
```

```
The times for each task are: [0.964252, 1.372077, 0.721504, 0.402504, 0.284232] with:
```

Minimum: 0.284232 Maximum: 1.372077 Average: 0.7489 seconds

Number of cars 2
bbox:car_number ((834, 400), (944, 507)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((970, 400), (1133, 512)) : 2
The minimum distance from car: 1 is 8.06225774829855
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

74%| | 929/1261 [41:23<14:47, 2.67s/it]

The times for each task are: [1.028321, 0.679995, 0.319735, 0.495083, 1.466769] with:

Minimum: 0.319735 Maximum: 1.466769 Average: 0.798 seconds

Number of cars 2
bbox:car_number ((834, 400), (944, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((970, 400), (1133, 512)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

74%| | 930/1261 [41:26<14:44, 2.67s/it]

The times for each task are: [0.643405, 1.443988, 1.092825, 0.311888, 0.404641] with:

Minimum: 0.311888 Maximum: 1.443988 Average: 0.7793 seconds

Number of cars 2
bbox:car_number ((975, 400), (1137, 507)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((834, 415), (932, 507)) : 2
The minimum distance from car: 1 is 10.0
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

```
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

74%| | 931/1261 [41:29<14:42, 2.67s/it]

The times for each task are: [0.872548, 0.674433, 1.469201, 0.398586, 0.423533] with:

Minimum: 0.398586 Maximum: 1.469201 Average: 0.7677 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (953, 512)) : 1
The minimum distance from car: 1 is 11.180339887498949
bbox:car_number ((975, 400), (1139, 507)) : 2
The minimum distance from car: 1 is 1.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| | 932/1261 [41:31<14:39, 2.67s/it]

The times for each task are: [1.394476, 0.632841, 0.90932, 0.41092, 0.304854] with:

Minimum: 0.304854 Maximum: 1.394476 Average: 0.7305 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (944, 512)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((975, 400), (1139, 507)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| | 933/1261 [41:34<14:36, 2.67s/it]

The times for each task are: [0.702661, 0.91722, 0.44792, 1.694772, 0.299507] with:

Minimum: 0.299507 Maximum: 1.694772 Average: 0.8124 seconds

Number of cars 2

bbox:car_number ((834, 400), (944, 512)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1000, 400), (1139, 507)) : 2

The minimum distance from car: 1 is 12.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (934, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1139, 507))]

Length of task list: 5

Number of processes used: 3

74%| | 934/1261 [41:36<14:34, 2.67s/it]

The times for each task are: [0.889014, 1.443682, 0.769271, 0.453801, 0.426352] with:

Minimum: 0.426352 Maximum: 1.443682 Average: 0.7964 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 512)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((989, 400), (1139, 507)) : 2

The minimum distance from car: 1 is 5.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (934, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1139, 507))]

Length of task list: 5

Number of processes used: 3

74%| | 935/1261 [41:39<14:31, 2.67s/it]

The times for each task are: [1.364661, 0.913105, 0.684911, 0.292738, 0.394715] with:

Minimum: 0.292738 Maximum: 1.364661 Average: 0.73 seconds

Number of cars 2

bbox:car_number ((834, 400), (942, 512)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((989, 400), (1139, 512)) : 2

The minimum distance from car: 1 is 3.0

```
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| 936/1261 [41:42<14:28, 2.67s/it]

The times for each task are: [1.390963, 0.611814, 0.997144, 0.39759, 0.325718] with:

Minimum: 0.325718 Maximum: 1.390963 Average: 0.7446 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 507)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((989, 400), (1139, 512)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| 937/1261 [41:44<14:26, 2.67s/it]

The times for each task are: [0.621569, 0.939632, 1.388306, 0.352241, 0.509057] with:

Minimum: 0.352241 Maximum: 1.388306 Average: 0.7622 seconds

```
Number of cars 4
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((989, 400), (1139, 507)) : 2
The minimum distance from car: 1 is 3.0
bbox:car_number ((940, 430), (942, 462)) : 3
bbox:car_number ((985, 445), (987, 462)) : 4
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| | 938/1261 [41:47<14:23, 2.67s/it]

The times for each task are: [0.638474, 0.896043, 1.587655, 0.363155, 0.401809] with:

Minimum: 0.363155 Maximum: 1.587655 Average: 0.7774 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((989, 400), (1143, 512)) : 2

The minimum distance from car: 1 is 3.605551275463989

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

74%| | 939/1261 [41:50<14:20, 2.67s/it]

The times for each task are: [0.941817, 1.359656, 0.752532, 0.455868, 0.33025] with:

Minimum: 0.33025 Maximum: 1.359656 Average: 0.768 seconds

Number of cars 2

bbox:car_number ((834, 400), (942, 512)) : 1

The minimum distance from car: 1 is 8.06225774829855

bbox:car_number ((989, 400), (1149, 518)) : 2

The minimum distance from car: 1 is 4.242640687119285

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

75%| | 940/1261 [41:52<14:18, 2.67s/it]

The times for each task are: [0.901348, 1.516262, 0.680614, 0.380088, 0.282605] with:

Minimum: 0.282605 Maximum: 1.516262 Average: 0.7522 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((989, 400), (1149, 512)) : 2
The minimum distance from car: 1 is 3.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

75%| 941/1261 [41:55<14:15, 2.67s/it]

The times for each task are: [0.658109, 1.016762, 1.476617, 0.381598, 0.355808] with:

Minimum: 0.355808 Maximum: 1.476617 Average: 0.7778 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((989, 400), (1149, 512)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

75%| 942/1261 [41:58<14:12, 2.67s/it]

The times for each task are: [0.706267, 1.03336, 1.480017, 0.417251, 0.394707] with:

Minimum: 0.394707 Maximum: 1.480017 Average: 0.8063 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((998, 400), (1160, 498)) : 2
The minimum distance from car: 1 is 12.206555615733702
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
```

```
Length of task list: 5
Number of processes used: 3
```

```
75%| | 943/1261 [42:01<14:10, 2.67s/it]
```

```
The times for each task are: [0.888457, 0.699986, 1.758551, 0.373857, 0.446852] with:
```

```
Minimum: 0.373857 Maximum: 1.758551 Average: 0.8335 seconds
```

```
Number of cars 2
bbox:car_number ((832, 400), (934, 512)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((994, 400), (1160, 512)) : 2
The minimum distance from car: 1 is 7.280109889280518
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
75%| | 944/1261 [42:03<14:07, 2.67s/it]
```

```
The times for each task are: [0.654481, 0.89725, 0.439874, 1.501398, 0.305041] with:
```

```
Minimum: 0.305041 Maximum: 1.501398 Average: 0.7596 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (934, 498)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((989, 400), (1162, 512)) : 2
The minimum distance from car: 1 is 2.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
75%| | 945/1261 [42:06<14:04, 2.67s/it]
```

```
The times for each task are: [0.684641, 0.87564, 1.491712, 0.467259, 0.308727] with:
```

Minimum: 0.308727 Maximum: 1.491712 Average: 0.7656 seconds

Number of cars 3
bbox:car_number ((834, 400), (934, 498)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((994, 400), (1160, 512)) : 2
The minimum distance from car: 1 is 2.0
bbox:car_number ((940, 430), (942, 477)) : 3
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

75%| 946/1261 [42:08<14:02, 2.67s/it]

The times for each task are: [0.637129, 1.052665, 1.46013, 0.483982, 0.342436] with:

Minimum: 0.342436 Maximum: 1.46013 Average: 0.7953 seconds
Number of cars 2
bbox:car_number ((832, 400), (942, 512)) : 1
The minimum distance from car: 1 is 7.615773105863909
bbox:car_number ((998, 400), (1162, 512)) : 2
The minimum distance from car: 1 is 3.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

75%| 947/1261 [42:11<13:59, 2.67s/it]

The times for each task are: [0.663178, 0.870635, 1.423569, 0.309282, 0.477683] with:

Minimum: 0.309282 Maximum: 1.423569 Average: 0.7489 seconds
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((989, 400), (1162, 516)) : 2
The minimum distance from car: 1 is 5.385164807134504
totalCars: 7

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 7  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))  
Length of task list: 5  
Number of processes used: 3
```

75%| | 948/1261 [42:13<13:56, 2.67s/it]

The times for each task are: [0.867605, 1.461483, 0.698926, 0.607158, 0.414369] with:

Minimum: 0.414369 Maximum: 1.461483 Average: 0.8099 seconds

```
Number of cars 2  
bbox:car_number ((832, 400), (942, 512)) : 1  
The minimum distance from car: 1 is 3.1622776601683795  
bbox:car_number ((998, 400), (1162, 512)) : 2  
The minimum distance from car: 1 is 5.385164807134504  
totalCars: 7  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 7  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))  
Length of task list: 5  
Number of processes used: 3
```

75%| | 949/1261 [42:16<13:53, 2.67s/it]

The times for each task are: [0.925734, 0.602615, 1.655734, 0.495096, 0.310267] with:

Minimum: 0.310267 Maximum: 1.655734 Average: 0.7979 seconds

```
Number of cars 2  
bbox:car_number ((834, 400), (934, 507)) : 1  
The minimum distance from car: 1 is 4.242640687119285  
bbox:car_number ((994, 400), (1162, 498)) : 2  
The minimum distance from car: 1 is 7.280109889280518  
totalCars: 7  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 7  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))  
Length of task list: 5  
Number of processes used: 3
```

75%| | 950/1261 [42:19<13:51, 2.67s/it]

The times for each task are: [0.975906, 0.619348, 1.507864, 0.459008, 0.307193] with:

Minimum: 0.307193 Maximum: 1.507864 Average: 0.7739 seconds

Number of cars 2

bbox:car_number ((834, 400), (932, 507)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((998, 400), (1160, 512)) : 2

The minimum distance from car: 1 is 7.0710678118654755

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

75%| | 951/1261 [42:21<13:48, 2.67s/it]

The times for each task are: [0.88865, 0.676254, 1.527149, 0.381214, 0.290742] with:

Minimum: 0.290742 Maximum: 1.527149 Average: 0.7528 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 4.123105625617661

bbox:car_number ((998, 400), (1162, 512)) : 2

The minimum distance from car: 1 is 1.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

75%| | 952/1261 [42:24<13:45, 2.67s/it]

The times for each task are: [0.625109, 0.918711, 1.678646, 0.404763, 0.309346] with:

Minimum: 0.309346 Maximum: 1.678646 Average: 0.7873 seconds

Number of cars 2

bbox:car_number ((832, 400), (942, 498)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((998, 400), (1162, 512)) : 2

The minimum distance from car: 1 is 0.0

```
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

76%| | 953/1261 [42:27<13:43, 2.67s/it]

The times for each task are: [1.401186, 0.860114, 0.76223, 0.331295, 0.483119] with:

Minimum: 0.331295 Maximum: 1.401186 Average: 0.7676 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1000, 400), (1162, 512)) : 2
The minimum distance from car: 1 is 1.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

76%| | 954/1261 [42:30<13:40, 2.67s/it]

The times for each task are: [0.900746, 1.407393, 0.691253, 0.414076, 0.322244] with:

Minimum: 0.322244 Maximum: 1.407393 Average: 0.7471 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (931, 498)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((994, 400), (1167, 512)) : 2
The minimum distance from car: 1 is 1.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

76%| | 955/1261 [42:32<13:38, 2.67s/it]

The times for each task are: [1.012445, 1.385128, 0.675902, 0.461668, 0.284581] with:

Minimum: 0.284581 Maximum: 1.385128 Average: 0.7639 seconds

Number of cars 2

bbox:car_number ((834, 400), (932, 498)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((998, 400), (1180, 512)) : 2

The minimum distance from car: 1 is 9.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

76%| | 956/1261 [42:35<13:35, 2.67s/it]

The times for each task are: [1.021648, 0.60275, 1.407777, 0.514404, 0.411927] with:

Minimum: 0.411927 Maximum: 1.407777 Average: 0.7917 seconds

Number of cars 2

bbox:car_number ((832, 400), (931, 498)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((1000, 400), (1180, 512)) : 2

The minimum distance from car: 1 is 1.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

76%| | 957/1261 [42:38<13:32, 2.67s/it]

The times for each task are: [0.884968, 0.666876, 1.40536, 0.451332, 0.340934] with:

Minimum: 0.340934 Maximum: 1.40536 Average: 0.7499 seconds

Number of cars 2

bbox:car_number ((998, 400), (1180, 512)) : 1

The minimum distance from car: 1 is 1.0

```
bbox:car_number ((834, 415), (934, 498)) : 2
The minimum distance from car: 1 is 7.615773105863909
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

76%| | 958/1261 [42:41<13:30, 2.67s/it]

The times for each task are: [0.712417, 0.847584, 0.297254, 0.540647, 1.421777] with:

Minimum: 0.297254 Maximum: 1.421777 Average: 0.7639 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1030, 400), (1180, 512)) : 2
The minimum distance from car: 3 is 8.54400374531753
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

76%| | 959/1261 [42:44<13:27, 2.67s/it]

The times for each task are: [0.897006, 0.714143, 1.647156, 0.408586, 0.38021] with:

Minimum: 0.38021 Maximum: 1.647156 Average: 0.8094 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 498)) : 1
The minimum distance from car: 1 is 5.656854249492381
bbox:car_number ((998, 400), (1181, 512)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

76%| | 960/1261 [42:47<13:24, 2.67s/it]

The times for each task are: [0.71803, 0.928097, 1.572716, 0.419893, 0.327009] with:

Minimum: 0.327009 Maximum: 1.572716 Average: 0.7931 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((1015, 400), (1181, 498)) : 2

The minimum distance from car: 3 is 9.899494936611665

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

76%| | 961/1261 [42:49<13:22, 2.67s/it]

The times for each task are: [0.87801, 0.597941, 1.419845, 0.395474, 0.383803] with:

Minimum: 0.383803 Maximum: 1.419845 Average: 0.735 seconds

Number of cars 2

bbox:car_number ((832, 400), (932, 512)) : 1

The minimum distance from car: 1 is 7.280109889280518

bbox:car_number ((1015, 400), (1185, 518)) : 2

The minimum distance from car: 3 is 10.198039027185569

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

76%| | 962/1261 [42:52<13:19, 2.67s/it]

The times for each task are: [0.919911, 0.62692, 1.419768, 0.394704, 0.300859] with:

Minimum: 0.300859 Maximum: 1.419768 Average: 0.7324 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 507)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((1021, 400), (1185, 512)) : 2
The minimum distance from car: 3 is 4.242640687119285
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

76%| | 963/1261 [42:55<13:16, 2.67s/it]

The times for each task are: [0.859714, 0.713611, 0.399731, 1.372749, 0.338537] with:

Minimum: 0.338537 Maximum: 1.372749 Average: 0.7369 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1021, 400), (1185, 512)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

76%| | 964/1261 [42:57<13:14, 2.67s/it]

The times for each task are: [0.872547, 1.588459, 0.711926, 0.484117, 0.32724] with:

Minimum: 0.32724 Maximum: 1.588459 Average: 0.7969 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1021, 400), (1185, 516)) : 2
The minimum distance from car: 3 is 2.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
```

```
Length of task list: 5
Number of processes used: 3
```

```
77%| | 965/1261 [43:00<13:11, 2.67s/it]
```

```
The times for each task are: [0.868568, 1.400412, 0.741923, 0.303536, 0.382062] with:
```

```
Minimum: 0.303536 Maximum: 1.400412 Average: 0.7393 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1021, 400), (1185, 498)) : 2
The minimum distance from car: 3 is 9.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

```
77%| | 966/1261 [43:03<13:08, 2.67s/it]
```

```
The times for each task are: [0.614503, 0.988849, 1.622882, 0.321469, 0.396667] with:
```

```
Minimum: 0.321469 Maximum: 1.622882 Average: 0.7889 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1021, 400), (1185, 507)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

```
77%| | 967/1261 [43:06<13:06, 2.67s/it]
```

```
The times for each task are: [0.631071, 0.965421, 1.671335, 0.389271, 0.513131] with:
```

Minimum: 0.389271 Maximum: 1.671335 Average: 0.834 seconds

Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1021, 400), (1187, 498)) : 2
The minimum distance from car: 3 is 4.123105625617661
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1181, 564))]
Length of task list: 5
Number of processes used: 3

77%| | 968/1261 [43:09<13:03, 2.67s/it]

The times for each task are: [0.571542, 0.93884, 1.471382, 0.407754, 0.333636] with:

Minimum: 0.333636 Maximum: 1.471382 Average: 0.7446 seconds

Number of cars 2
bbox:car_number ((1021, 400), (1187, 498)) : 1
The minimum distance from car: 3 is 0.0
bbox:car_number ((834, 415), (934, 507)) : 2
The minimum distance from car: 1 is 8.94427190999916
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1181, 564))]
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

77%| | 969/1261 [43:11<13:00, 2.67s/it]

The times for each task are: [0.972018, 0.642663, 1.722736, 0.453604, 0.284493] with:

Minimum: 0.284493 Maximum: 1.722736 Average: 0.8151 seconds

Number of cars 2
bbox:car_number ((835, 400), (932, 498)) : 1
The minimum distance from car: 1 is 12.041594578792296
bbox:car_number ((1027, 400), (1200, 512)) : 2
The minimum distance from car: 3 is 11.40175425099138
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

```
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

77%| 970/1261 [43:14<12:58, 2.67s/it]

The times for each task are: [0.66824, 1.390465, 0.858358, 0.383261, 0.425864] with:

Minimum: 0.383261 Maximum: 1.390465 Average: 0.7452 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1027, 400), (1208, 498)) : 2
The minimum distance from car: 3 is 8.06225774829855
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

77%| 971/1261 [43:17<12:55, 2.67s/it]

The times for each task are: [1.391861, 0.991206, 0.619068, 0.311632, 0.413085] with:

Minimum: 0.311632 Maximum: 1.391861 Average: 0.7454 seconds

```
Number of cars 2
bbox:car_number ((1027, 400), (1208, 498)) : 1
The minimum distance from car: 3 is 0.0
bbox:car_number ((835, 415), (932, 495)) : 2
The minimum distance from car: 1 is 6.0
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

77%| 972/1261 [43:19<12:52, 2.67s/it]

The times for each task are: [0.924211, 0.674303, 1.416463, 0.42439, 0.355519] with:

Minimum: 0.355519 Maximum: 1.416463 Average: 0.759 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 6.082762530298219

bbox:car_number ((1027, 400), (1211, 507)) : 2

The minimum distance from car: 3 is 4.47213595499958

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

77% | 973/1261 [43:22<12:50, 2.67s/it]

The times for each task are: [1.418979, 0.628198, 1.034566, 0.356137, 0.395339] with:

Minimum: 0.356137 Maximum: 1.418979 Average: 0.7666 seconds

Number of cars 3

bbox:car_number ((834, 400), (934, 507)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((1021, 400), (1208, 512)) : 2

The minimum distance from car: 3 is 5.830951894845301

bbox:car_number ((1210, 430), (1211, 477)) : 3

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

77% | 974/1261 [43:25<12:47, 2.67s/it]

The times for each task are: [0.955887, 0.695817, 1.475356, 0.542044, 0.367873] with:

Minimum: 0.367873 Maximum: 1.475356 Average: 0.8074 seconds

Number of cars 2

bbox:car_number ((835, 400), (932, 497)) : 1

The minimum distance from car: 1 is 5.0990195135927845

bbox:car_number ((1027, 400), (1208, 498)) : 2

```
The minimum distance from car: 3 is 7.615773105863909
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

77%| | 975/1261 [43:27<12:44, 2.67s/it]

```
The times for each task are: [0.928961, 1.456654, 0.664956, 0.481785, 0.329192] with:
Minimum: 0.329192 Maximum: 1.456654 Average: 0.7723 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((1021, 400), (1211, 512)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

77%| | 976/1261 [43:30<12:42, 2.67s/it]

```
The times for each task are: [0.995975, 0.670268, 1.610377, 0.395458, 0.304323] with:
Minimum: 0.304323 Maximum: 1.610377 Average: 0.7953 seconds
```

```
Number of cars 3
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1027, 400), (1208, 512)) : 2
The minimum distance from car: 3 is 1.0
bbox:car_number ((1210, 415), (1214, 462)) : 3
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

77%| | 977/1261 [43:33<12:39, 2.67s/it]

The times for each task are: [0.654763, 0.979824, 1.421829, 0.309779, 0.427884] with:

Minimum: 0.309779 Maximum: 1.421829 Average: 0.7588 seconds

Number of cars 2

bbox:car_number ((1044, 400), (1214, 512)) : 1

The minimum distance from car: 3 is 12.0

bbox:car_number ((834, 415), (932, 507)) : 2

The minimum distance from car: 1 is 8.06225774829855

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (

Length of task list: 5

Number of processes used: 3

78%| | 978/1261 [43:35<12:36, 2.67s/it]

The times for each task are: [0.911188, 0.647588, 1.510641, 0.348757, 0.53736] with:

Minimum: 0.348757 Maximum: 1.510641 Average: 0.7911 seconds

Number of cars 2

bbox:car_number ((835, 400), (932, 498)) : 1

The minimum distance from car: 1 is 12.0

bbox:car_number ((1045, 400), (1212, 512)) : 2

The minimum distance from car: 3 is 1.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

Length of task list: 5

Number of processes used: 3

78%| | 979/1261 [43:38<12:34, 2.67s/it]

The times for each task are: [0.918477, 0.634975, 1.574873, 0.326789, 0.44113] with:

Minimum: 0.326789 Maximum: 1.574873 Average: 0.7792 seconds

```
Number of cars 2
bbox:car_number ((837, 400), (934, 497)) : 1
The minimum distance from car: 1 is 2.23606797749979
bbox:car_number ((1045, 400), (1214, 512)) : 2
The minimum distance from car: 3 is 1.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]
Length of task list: 5
Number of processes used: 3
```

78%| 980/1261 [43:41<12:31, 2.67s/it]

The times for each task are: [0.701253, 0.958952, 0.343879, 1.738413, 0.426995] with:

Minimum: 0.343879 Maximum: 1.738413 Average: 0.8339 seconds

```
Number of cars 2
bbox:car_number ((837, 400), (934, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1048, 400), (1214, 518)) : 2
The minimum distance from car: 3 is 3.605551275463989
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]
Length of task list: 5
Number of processes used: 3
```

78%| 981/1261 [43:43<12:28, 2.67s/it]

The times for each task are: [0.572051, 1.38194, 0.860411, 0.404165, 0.287922] with:

Minimum: 0.287922 Maximum: 1.38194 Average: 0.7013 seconds

```
Number of cars 3
bbox:car_number ((837, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1046, 400), (1214, 512)) : 2
The minimum distance from car: 3 is 3.1622776601683795
bbox:car_number ((571, 475), (612, 507)) : 3
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
```

```
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

78%| 982/1261 [43:46<12:26, 2.67s/it]

The times for each task are: [0.90255, 1.406447, 0.634923, 0.509247, 0.301456] with:

Minimum: 0.301456 Maximum: 1.406447 Average: 0.7509 seconds

```
Number of cars 5
bbox:car_number ((837, 400), (934, 507)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((1060, 400), (1230, 512)) : 2
The minimum distance from car: 3 is 15.0
bbox:car_number ((550, 475), (611, 537)) : 3
bbox:car_number ((520, 520), (537, 552)) : 4
bbox:car_number ((505, 550), (507, 567)) : 5
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
totalCars: 8
Car Number: 3 Car Positions: [((550, 475), (611, 537))]
Length of task list: 5
Number of processes used: 3
```

78%| 983/1261 [43:48<12:23, 2.67s/it]

The times for each task are: [0.86497, 0.731692, 1.411055, 0.390383, 0.333398] with:

Minimum: 0.333398 Maximum: 1.411055 Average: 0.7463 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1048, 400), (1230, 512)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

78%| 984/1261 [43:51<12:20, 2.67s/it]

The times for each task are: [0.623129, 0.981371, 1.374871, 0.400169, 0.306579] with:

Minimum: 0.306579 Maximum: 1.374871 Average: 0.7372 seconds

Number of cars 3
bbox:car_number ((836, 400), (934, 497)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1048, 400), (1241, 512)) : 2
The minimum distance from car: 3 is 5.0
bbox:car_number ((533, 469), (612, 552)) : 3
The minimum distance from car: 3 is 8.94427190999916
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 490), (1178, 552))]
totalCars: 8
Car Number: 3 Car Positions: [((550, 475), (611, 537)), ((533, 469), (612, 552))]
Length of task list: 5
Number of processes used: 3

78%| 985/1261 [43:54<12:18, 2.67s/it]

The times for each task are: [1.397331, 0.700563, 0.843806, 0.407946, 0.45436] with:

Minimum: 0.407946 Maximum: 1.397331 Average: 0.7608 seconds

Number of cars 3
bbox:car_number ((837, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1046, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 7.0710678118654755
bbox:car_number ((535, 490), (592, 554)) : 3
The minimum distance from car: 3 is 15.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 490), (1178, 552))]
totalCars: 8
Car Number: 3 Car Positions: [((550, 475), (611, 537)), ((533, 469), (612, 552)), ((535, 490), (611, 552))]
Length of task list: 5
Number of processes used: 3

78%| 986/1261 [43:57<12:15, 2.67s/it]

The times for each task are: [0.875541, 1.593565, 0.623997, 0.403832, 0.382394] with:

Minimum: 0.382394 Maximum: 1.593565 Average: 0.7759 seconds

Number of cars 4
bbox:car_number ((837, 400), (932, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1048, 400), (1230, 498)) : 2
The minimum distance from car: 3 is 4.0
bbox:car_number ((514, 492), (586, 564)) : 3
The minimum distance from car: 3 is 14.317821063276353
bbox:car_number ((493, 550), (507, 582)) : 4
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 490), (1181, 564))]
totalCars: 8
Car Number: 3 Car Positions: [((550, 475), (611, 537)), ((533, 469), (612, 552)), ((535, 490), (611, 537))]
Length of task list: 5
Number of processes used: 3

78%| 987/1261 [43:59<12:12, 2.67s/it]

The times for each task are: [0.637678, 1.541309, 0.949828, 0.322097, 0.414062] with:

Minimum: 0.322097 Maximum: 1.541309 Average: 0.773 seconds

Number of cars 2
bbox:car_number ((837, 400), (934, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1048, 400), (1241, 518)) : 2
The minimum distance from car: 3 is 11.180339887498949
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 490), (1181, 564))]
Length of task list: 5
Number of processes used: 3

78%| 988/1261 [44:02<12:10, 2.67s/it]

The times for each task are: [0.895456, 0.632603, 1.487994, 0.41943, 0.313124] with:

Minimum: 0.313124 Maximum: 1.487994 Average: 0.7497 seconds

Number of cars 2

bbox:car_number ((837, 400), (931, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1051, 400), (1238, 512)) : 2

The minimum distance from car: 3 is 3.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

78%| | 989/1261 [44:05<12:07, 2.67s/it]

The times for each task are: [0.713809, 1.397821, 1.005711, 0.482285, 0.298342] with:

Minimum: 0.298342 Maximum: 1.397821 Average: 0.7796 seconds

Number of cars 2

bbox:car_number ((856, 400), (934, 498)) : 1

The minimum distance from car: 1 is 11.045361017187261

bbox:car_number ((1046, 400), (1241, 518)) : 2

The minimum distance from car: 3 is 3.1622776601683795

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

79%| | 990/1261 [44:08<12:04, 2.67s/it]

The times for each task are: [0.617712, 1.042627, 1.681348, 0.423946, 0.396889] with:

Minimum: 0.396889 Maximum: 1.681348 Average: 0.8325 seconds

Number of cars 2

bbox:car_number ((1048, 400), (1241, 518)) : 1

The minimum distance from car: 3 is 1.0

bbox:car_number ((836, 415), (934, 498)) : 2

The minimum distance from car: 1 is 12.206555615733702

```
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

79%| 991/1261 [44:10<12:02, 2.67s/it]

The times for each task are: [0.624205, 0.888847, 1.496778, 0.425025, 0.315318] with:
Minimum: 0.315318 Maximum: 1.496778 Average: 0.75 seconds

```
Number of cars 2
bbox:car_number ((859, 400), (934, 497)) : 1
The minimum distance from car: 1 is 13.601470508735444
bbox:car_number ((1048, 400), (1241, 518)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

79%| 992/1261 [44:13<11:59, 2.67s/it]

The times for each task are: [0.889961, 0.702088, 1.731176, 0.338834, 0.503202] with:
Minimum: 0.338834 Maximum: 1.731176 Average: 0.8331 seconds

```
Number of cars 2
bbox:car_number ((1048, 400), (1242, 522)) : 1
The minimum distance from car: 3 is 2.23606797749979
bbox:car_number ((856, 415), (942, 498)) : 2
The minimum distance from car: 1 is 8.54400374531753
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

79%| | 993/1261 [44:15<11:56, 2.67s/it]

The times for each task are: [0.845164, 1.596004, 0.710834, 0.501504, 0.351181] with:

Minimum: 0.351181 Maximum: 1.596004 Average: 0.8009 seconds

Number of cars 2

bbox:car_number ((856, 400), (934, 497)) : 1

The minimum distance from car: 1 is 8.94427190999916

bbox:car_number ((1051, 400), (1242, 518)) : 2

The minimum distance from car: 3 is 2.23606797749979

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

79%| | 994/1261 [44:18<11:54, 2.67s/it]

The times for each task are: [0.69022, 0.904563, 1.411192, 0.469413, 0.292573] with:

Minimum: 0.292573 Maximum: 1.411192 Average: 0.7536 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 11.045361017187261

bbox:car_number ((1065, 400), (1257, 518)) : 2

The minimum distance from car: 3 is 15.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

79%| | 995/1261 [44:21<11:51, 2.67s/it]

The times for each task are: [1.405896, 0.874676, 0.667139, 0.39571, 0.297467] with:

Minimum: 0.297467 Maximum: 1.405896 Average: 0.7282 seconds

Number of cars 2

bbox:car_number ((1051, 400), (1253, 518)) : 1

The minimum distance from car: 3 is 9.0

```
bbox:car_number ((859, 415), (932, 478)) : 2
The minimum distance from car: 1 is 11.40175425099138
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

79%| 996/1261 [44:24<11:48, 2.67s/it]

The times for each task are: [0.834718, 0.693258, 1.446434, 0.431403, 0.391222] with:

Minimum: 0.391222 Maximum: 1.446434 Average: 0.7594 seconds

```
Number of cars 2
bbox:car_number ((859, 400), (932, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1051, 400), (1253, 522)) : 2
The minimum distance from car: 3 is 2.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

79%| 997/1261 [44:26<11:46, 2.67s/it]

The times for each task are: [0.610542, 1.448996, 0.956353, 0.311605, 0.535194] with:

Minimum: 0.311605 Maximum: 1.448996 Average: 0.7725 seconds

```
Number of cars 2
bbox:car_number ((1051, 400), (1257, 516)) : 1
The minimum distance from car: 3 is 3.605551275463989
bbox:car_number ((859, 415), (927, 485)) : 2
The minimum distance from car: 1 is 3.605551275463989
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

79%| | 998/1261 [44:29<11:43, 2.67s/it]

The times for each task are: [0.903365, 1.445383, 0.745499, 0.408492, 0.354971] with:

Minimum: 0.354971 Maximum: 1.445383 Average: 0.7715 seconds

Number of cars 2

bbox:car_number ((1065, 400), (1253, 512)) : 1

The minimum distance from car: 3 is 5.385164807134504

bbox:car_number ((835, 415), (932, 498)) : 2

The minimum distance from car: 1 is 11.661903789690601

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (

Length of task list: 5

Number of processes used: 3

79%| | 999/1261 [44:32<11:40, 2.68s/it]

The times for each task are: [0.860437, 0.749569, 1.4925, 0.507096, 0.346532] with:

Minimum: 0.346532 Maximum: 1.4925 Average: 0.7912 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 7.0710678118654755

bbox:car_number ((1065, 400), (1242, 512)) : 2

The minimum distance from car: 3 is 6.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

Length of task list: 5

Number of processes used: 3

79%| | 1000/1261 [44:35<11:38, 2.68s/it]

The times for each task are: [0.927117, 0.646638, 0.419255, 1.451554, 0.339086] with:

Minimum: 0.339086 Maximum: 1.451554 Average: 0.7567 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1065, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

79%| 1001/1261 [44:38<11:35, 2.68s/it]

The times for each task are: [0.912315, 0.63437, 1.455428, 0.525234, 0.426108] with:

Minimum: 0.426108 Maximum: 1.455428 Average: 0.7907 seconds

```
Number of cars 2
bbox:car_number ((1129, 423), (1139, 459)) : 1
bbox:car_number ((1165, 430), (1212, 477)) : 2
Length of task list: 5
Number of processes used: 3
```

79%| 1002/1261 [44:40<11:32, 2.68s/it]

The times for each task are: [0.926571, 0.636416, 1.452835, 0.444685, 0.383333] with:

Minimum: 0.383333 Maximum: 1.452835 Average: 0.7688 seconds

```
Number of cars 2
bbox:car_number ((837, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1090, 400), (1214, 485)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

80%| 1003/1261 [44:43<11:30, 2.68s/it]

The times for each task are: [0.874368, 0.636479, 0.409098, 1.727248, 0.334504] with:

Minimum: 0.334504 Maximum: 1.727248 Average: 0.7963 seconds

Number of cars 2

bbox:car_number ((859, 400), (932, 485)) : 1

The minimum distance from car: 1 is 12.529964086141668

bbox:car_number ((1084, 400), (1238, 485)) : 2

The minimum distance from car: 3 is 9.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

80%| 1004/1261 [44:45<11:27, 2.68s/it]

The times for each task are: [1.392279, 0.863091, 0.617316, 0.30216, 0.390169] with:

Minimum: 0.30216 Maximum: 1.392279 Average: 0.713 seconds

Number of cars 2

bbox:car_number ((835, 400), (908, 485)) : 1

The minimum distance from car: 1 is 24.0

bbox:car_number ((1084, 400), (1238, 485)) : 2

The minimum distance from car: 3 is 0.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

80%| 1005/1261 [44:48<11:24, 2.68s/it]

The times for each task are: [0.579833, 0.923417, 1.502294, 0.511344, 0.318754] with:

Minimum: 0.318754 Maximum: 1.502294 Average: 0.7671 seconds

Number of cars 3

bbox:car_number ((859, 400), (932, 495)) : 1

The minimum distance from car: 1 is 24.515301344262525

bbox:car_number ((1156, 400), (1214, 472)) : 2

The minimum distance from car: 3 is 24.73863375370596

```
bbox:car_number ((1129, 427), (1139, 485)) : 3
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

80%| 1006/1261 [44:51<11:22, 2.68s/it]

The times for each task are: [0.880638, 1.356562, 0.72118, 0.497949, 0.40465] with:

Minimum: 0.40465 Maximum: 1.356562 Average: 0.7722 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (944, 498)) : 1
The minimum distance from car: 1 is 6.324555320336759
bbox:car_number ((1105, 400), (1219, 485)) : 2
The minimum distance from car: 3 is 23.769728648009426
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

80%| 1007/1261 [44:53<11:19, 2.67s/it]

The times for each task are: [0.98731, 0.661359, 1.385367, 0.351856, 0.37622] with:

Minimum: 0.351856 Maximum: 1.385367 Average: 0.7524 seconds

```
Number of cars 2
bbox:car_number ((1075, 400), (1219, 498)) : 1
The minimum distance from car: 3 is 16.55294535724685
bbox:car_number ((859, 423), (932, 495)) : 2
The minimum distance from car: 1 is 11.661903789690601
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

80%| 1008/1261 [44:56<11:16, 2.67s/it]

The times for each task are: [0.624937, 1.421736, 0.912295, 0.296116, 0.392715] with:

Minimum: 0.296116 Maximum: 1.421736 Average: 0.7296 seconds

Number of cars 2

bbox:car_number ((1075, 400), (1227, 497)) : 1

The minimum distance from car: 3 is 4.123105625617661

bbox:car_number ((836, 415), (932, 498)) : 2

The minimum distance from car: 1 is 11.40175425099138

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

80%| 1009/1261 [44:59<11:14, 2.68s/it]

The times for each task are: [0.883024, 0.640642, 1.397928, 0.290572, 0.530395] with:

Minimum: 0.290572 Maximum: 1.397928 Average: 0.7485 seconds

Number of cars 2

bbox:car_number ((835, 400), (934, 498)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1090, 400), (1219, 485)) : 2

The minimum distance from car: 3 is 6.708203932499369

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

Length of task list: 5

Number of processes used: 3

80%| 1010/1261 [45:02<11:11, 2.68s/it]

The times for each task are: [1.021854, 0.62611, 1.390654, 0.400691, 0.350568] with:

Minimum: 0.350568 Maximum: 1.390654 Average: 0.758 seconds

Number of cars 2

bbox:car_number ((835, 400), (934, 498)) : 1

The minimum distance from car: 1 is 0.0

```
bbox:car_number ((1090, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 13.038404810405298
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

80%| 1011/1261 [45:05<11:08, 2.68s/it]

The times for each task are: [1.000676, 0.626131, 1.641556, 0.355829, 0.440978] with:

Minimum: 0.355829 Maximum: 1.641556 Average: 0.813 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1082, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

80%| 1012/1261 [45:07<11:06, 2.68s/it]

The times for each task are: [0.929617, 0.706285, 1.617784, 0.414925, 0.342529] with:

Minimum: 0.342529 Maximum: 1.617784 Average: 0.8022 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((1090, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

80%| 1013/1261 [45:10<11:03, 2.68s/it]

The times for each task are: [1.438679, 0.650393, 0.951721, 0.319154, 0.432956] with:

Minimum: 0.319154 Maximum: 1.438679 Average: 0.7586 seconds

Number of cars 2

bbox:car_number ((835, 400), (942, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1102, 400), (1253, 498)) : 2

The minimum distance from car: 3 is 12.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

80%| 1014/1261 [45:13<11:00, 2.68s/it]

The times for each task are: [0.875984, 0.726819, 0.411242, 1.415366, 0.375381] with:

Minimum: 0.375381 Maximum: 1.415366 Average: 0.761 seconds

Number of cars 2

bbox:car_number ((835, 400), (942, 497)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1090, 400), (1253, 498)) : 2

The minimum distance from car: 3 is 6.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

80%| 1015/1261 [45:15<10:58, 2.68s/it]

The times for each task are: [0.898208, 0.675696, 1.424032, 0.30281, 0.480939] with:

Minimum: 0.30281 Maximum: 1.424032 Average: 0.7563 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1103, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

81%| 1016/1261 [45:18<10:55, 2.68s/it]

The times for each task are: [0.643949, 0.908254, 0.471357, 0.396154, 1.416997] with:

Minimum: 0.396154 Maximum: 1.416997 Average: 0.7673 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1102, 400), (1257, 512)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

81%| 1017/1261 [45:21<10:52, 2.68s/it]

The times for each task are: [0.622282, 0.85008, 1.647435, 0.49865, 0.385759] with:

Minimum: 0.385759 Maximum: 1.647435 Average: 0.8008 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1075, 400), (1242, 512)) : 2
The minimum distance from car: 3 is 21.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
81%| 1018/1261 [45:23<10:50, 2.68s/it]
```

```
The times for each task are: [0.926096, 1.376967, 0.698827, 0.308648, 0.496606] with:
```

```
Minimum: 0.308648 Maximum: 1.376967 Average: 0.7614 seconds
```

```
Number of cars 4
bbox:car_number ((834, 400), (942, 498)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1113, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 25.96150997149434
bbox:car_number ((1075, 427), (1077, 459)) : 3
bbox:car_number ((1082, 427), (1086, 459)) : 4
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

```
81%| 1019/1261 [45:26<10:47, 2.68s/it]
```

```
The times for each task are: [0.831858, 1.365793, 0.686761, 0.322548, 0.40633] with:
```

```
Minimum: 0.322548 Maximum: 1.365793 Average: 0.7227 seconds
```

```
Number of cars 2
bbox:car_number ((835, 400), (942, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1046, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 40.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

```
81%| 1020/1261 [45:29<10:44, 2.68s/it]
```

The times for each task are: [0.891539, 0.677227, 1.355484, 0.406848, 0.419719] with:

Minimum: 0.406848 Maximum: 1.355484 Average: 0.7502 seconds

Number of cars 3

bbox:car_number ((834, 400), (944, 498)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((1090, 400), (1253, 498)) : 2

The minimum distance from car: 3 is 28.0

bbox:car_number ((1082, 427), (1086, 459)) : 3

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

81%| 1021/1261 [45:32<10:42, 2.68s/it]

The times for each task are: [0.625466, 0.99509, 1.429135, 0.406143, 0.308223] with:

Minimum: 0.308223 Maximum: 1.429135 Average: 0.7528 seconds

Number of cars 2

bbox:car_number ((835, 400), (942, 498)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1082, 400), (1257, 498)) : 2

The minimum distance from car: 3 is 2.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

81%| 1022/1261 [45:34<10:39, 2.68s/it]

The times for each task are: [0.910132, 0.7042, 1.495912, 0.427355, 0.359437] with:

Minimum: 0.359437 Maximum: 1.495912 Average: 0.7794 seconds

Number of cars 3

bbox:car_number ((835, 400), (942, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1102, 400), (1187, 485)) : 2

```
The minimum distance from car: 3 is 25.96150997149434
bbox:car_number ((1195, 400), (1268, 477)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1175, 400))]
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

81% | 1023/1261 [45:37<10:36, 2.68s/it]

```
The times for each task are: [0.88517, 0.703119, 1.622039, 0.453937, 0.356373] with:
Minimum: 0.356373 Maximum: 1.622039 Average: 0.8041 seconds
```

```
Number of cars 2
bbox:car_number ((835, 400), (934, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((1102, 400), (1268, 498)) : 2
The minimum distance from car: 3 is 41.593268686170845
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1175, 400))]
Length of task list: 5
Number of processes used: 3
```

81% | 1024/1261 [45:40<10:34, 2.68s/it]

```
The times for each task are: [1.013374, 0.620011, 1.385682, 0.472858, 0.341116] with:
Minimum: 0.341116 Maximum: 1.385682 Average: 0.7666 seconds
```

```
Number of cars 3
bbox:car_number ((835, 400), (942, 498)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1175, 400), (1268, 497)) : 2
The minimum distance from car: 3 is 14.142135623730951
bbox:car_number ((1122, 427), (1160, 459)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
81%| | 1025/1261 [45:42<10:31, 2.68s/it]
```

```
The times for each task are: [0.660267, 0.866926, 1.673749, 0.300808, 0.402429] with:
```

```
Minimum: 0.300808 Maximum: 1.673749 Average: 0.7808 seconds
```

```
Number of cars 2
bbox:car_number ((836, 400), (942, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1122, 400), (1273, 485)) : 2
The minimum distance from car: 3 is 13.892443989449804
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

```
81%| | 1026/1261 [45:46<10:28, 2.68s/it]
```

```
The times for each task are: [1.416731, 0.683489, 0.889417, 0.342273, 0.530844] with:
```

```
Minimum: 0.342273 Maximum: 1.416731 Average: 0.7726 seconds
```

```
Number of cars 2
bbox:car_number ((835, 400), (931, 492)) : 1
The minimum distance from car: 1 is 6.708203932499369
bbox:car_number ((1102, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 12.206555615733702
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

```
81%| | 1027/1261 [45:48<10:26, 2.68s/it]
```

```
The times for each task are: [0.89245, 0.615127, 0.410857, 1.729469, 0.419233] with:
```

Minimum: 0.410857 Maximum: 1.729469 Average: 0.8134 seconds

Number of cars 2
bbox:car_number ((835, 400), (932, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1102, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3

82%| 1028/1261 [45:51<10:23, 2.68s/it]

The times for each task are: [0.614165, 1.054729, 0.534203, 0.345344, 1.867115] with:

Minimum: 0.345344 Maximum: 1.867115 Average: 0.8831 seconds

Number of cars 2
bbox:car_number ((835, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1102, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3

82%| 1029/1261 [45:54<10:21, 2.68s/it]

The times for each task are: [0.63079, 1.052743, 1.424294, 0.382393, 0.46175] with:

Minimum: 0.382393 Maximum: 1.424294 Average: 0.7904 seconds

Number of cars 2
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1273, 512)) : 2
The minimum distance from car: 3 is 9.219544457292887
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

```
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

82%| 1030/1261 [45:57<10:18, 2.68s/it]

```
The times for each task are: [1.405329, 0.976788, 0.754328, 0.311659, 0.504686] with:
Minimum: 0.311659 Maximum: 1.405329 Average: 0.7906 seconds
```

```
Number of cars 2
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1113, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

82%| 1031/1261 [46:00<10:15, 2.68s/it]

```
The times for each task are: [0.641551, 0.922958, 1.532446, 0.475677, 0.325731] with:
Minimum: 0.325731 Maximum: 1.532446 Average: 0.7797 seconds
```

```
Number of cars 3
bbox:car_number ((834, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 0.0
bbox:car_number ((535, 475), (567, 507)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

82%| 1032/1261 [46:03<10:13, 2.68s/it]

The times for each task are: [0.706868, 0.935417, 1.464775, 0.484979, 0.350072] with:

Minimum: 0.350072 Maximum: 1.464775 Average: 0.7884 seconds

Number of cars 3

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((1113, 400), (1187, 498)) : 2

The minimum distance from car: 3 is 43.0

bbox:car_number ((1195, 400), (1273, 495)) : 3

The minimum distance from car: 3 is 13.038404810405298

totalCars: 9

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 9

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1175, 400))]

totalCars: 9

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

82% | 1033/1261 [46:05<10:10, 2.68s/it]

The times for each task are: [0.65778, 0.932666, 1.463192, 0.428793, 0.366011] with:

Minimum: 0.366011 Maximum: 1.463192 Average: 0.7697 seconds

Number of cars 2

bbox:car_number ((834, 400), (932, 497)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((1135, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 30.066592756745816

totalCars: 9

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 9

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

82% | 1034/1261 [46:08<10:07, 2.68s/it]

The times for each task are: [0.740974, 1.455243, 0.892139, 0.522706, 0.317334] with:

Minimum: 0.317334 Maximum: 1.455243 Average: 0.7857 seconds

Number of cars 3

```
bbox:car_number ((832, 400), (934, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1135, 400), (1272, 498)) : 2
The minimum distance from car: 3 is 1.0
bbox:car_number ((514, 475), (567, 522)) : 3
The minimum distance from car: 3 is 31.622776601683793
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
totalCars: 9
Car Number: 3 Car Positions: [((550, 475), (611, 537)), ((533, 469), (612, 552)), ((535, 490), (611, 537))]
Length of task list: 5
Number of processes used: 3
```

82%| | 1035/1261 [46:11<10:05, 2.68s/it]

The times for each task are: [0.578041, 1.435782, 0.923949, 0.482694, 0.385929] with:

Minimum: 0.385929 Maximum: 1.435782 Average: 0.7613 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (934, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1268, 498)) : 2
The minimum distance from car: 3 is 13.0
bbox:car_number ((493, 505), (522, 507)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

82%| | 1036/1261 [46:13<10:02, 2.68s/it]

The times for each task are: [0.906199, 0.598757, 0.297699, 0.415934, 1.665732] with:

Minimum: 0.297699 Maximum: 1.665732 Average: 0.7769 seconds

```
Number of cars 3
bbox:car_number ((832, 400), (942, 498)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1113, 400), (1272, 498)) : 2
The minimum distance from car: 3 is 2.0
```

```
bbox:car_number ((505, 505), (522, 537)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

82%| 1037/1261 [46:16<09:59, 2.68s/it]

The times for each task are: [0.925364, 0.617252, 1.632639, 0.401126, 0.326014] with:

Minimum: 0.326014 Maximum: 1.632639 Average: 0.7805 seconds

```
Number of cars 5
bbox:car_number ((834, 400), (942, 512)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((1136, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 12.0
bbox:car_number ((1027, 415), (1029, 447)) : 3
bbox:car_number ((490, 505), (492, 522)) : 4
bbox:car_number ((495, 505), (497, 522)) : 5
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

82%| 1038/1261 [46:19<09:57, 2.68s/it]

The times for each task are: [1.509342, 0.844263, 0.626971, 0.323528, 0.439307] with:

Minimum: 0.323528 Maximum: 1.509342 Average: 0.7487 seconds

```
Number of cars 4
bbox:car_number ((835, 400), (942, 498)) : 1
The minimum distance from car: 1 is 7.0
bbox:car_number ((1113, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 11.0
bbox:car_number ((1102, 427), (1106, 462)) : 3
bbox:car_number ((490, 520), (507, 522)) : 4
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
```

```
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

82% | 1039/1261 [46:22<09:54, 2.68s/it]

The times for each task are: [1.044472, 0.594589, 1.513529, 0.32441, 0.444736] with:

Minimum: 0.32441 Maximum: 1.513529 Average: 0.7843 seconds

```
Number of cars 4  
bbox:car_number ((834, 400), (932, 498)) : 1  
The minimum distance from car: 1 is 5.0  
bbox:car_number ((1113, 400), (1276, 495)) : 2  
The minimum distance from car: 3 is 2.23606797749979  
bbox:car_number ((1030, 415), (1047, 459)) : 3  
bbox:car_number ((460, 514), (516, 560)) : 4  
totalCars: 10  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
totalCars: 10  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
totalCars: 10  
Car Number: 4 Car Positions: [((460, 514), (516, 560))]  
Length of task list: 5  
Number of processes used: 3
```

82% | 1040/1261 [46:24<09:51, 2.68s/it]

The times for each task are: [0.882931, 1.404179, 0.6831, 0.405676, 0.285582] with:

Minimum: 0.285582 Maximum: 1.404179 Average: 0.7323 seconds

```
Number of cars 2  
bbox:car_number ((834, 400), (932, 498)) : 1  
The minimum distance from car: 1 is 0.0  
bbox:car_number ((1113, 400), (1273, 498)) : 2  
The minimum distance from car: 3 is 2.23606797749979  
totalCars: 10  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
totalCars: 10  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

83%| | 1041/1261 [46:27<09:49, 2.68s/it]

The times for each task are: [0.885723, 0.608079, 0.402974, 1.673033, 0.276407] with:

Minimum: 0.276407 Maximum: 1.673033 Average: 0.7692 seconds

Number of cars 4

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1129, 400), (1214, 485)) : 2

The minimum distance from car: 3 is 22.135943621178654

bbox:car_number ((1217, 415), (1272, 472)) : 3

bbox:car_number ((1075, 535), (1107, 582)) : 4

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1175, 535))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

83%| | 1042/1261 [46:30<09:46, 2.68s/it]

The times for each task are: [0.745219, 0.938462, 0.4329, 1.394799, 0.353214] with:

Minimum: 0.353214 Maximum: 1.394799 Average: 0.7729 seconds

Number of cars 2

bbox:car_number ((1113, 400), (1273, 498)) : 1

The minimum distance from car: 3 is 0.0

bbox:car_number ((834, 415), (934, 512)) : 2

The minimum distance from car: 1 is 14.0

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 497))]

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

83%| | 1043/1261 [46:33<09:43, 2.68s/it]

The times for each task are: [0.611678, 1.361687, 0.985162, 0.294819, 0.532941] with:

Minimum: 0.294819 Maximum: 1.361687 Average: 0.7573 seconds

```
Number of cars 4
bbox:car_number ((834, 400), (942, 497)) : 1
The minimum distance from car: 1 is 15.524174696260024
bbox:car_number ((1129, 400), (1200, 485)) : 2
The minimum distance from car: 3 is 7.0
bbox:car_number ((1204, 400), (1214, 472)) : 3
bbox:car_number ((1217, 400), (1276, 477)) : 4
The minimum distance from car: 3 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (934, 497))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477))]
Length of task list: 5
Number of processes used: 3
```

83% | 1044/1261 [46:36<09:41, 2.68s/it]

The times for each task are: [0.868499, 1.359384, 0.702062, 0.371737, 0.283576] with:

Minimum: 0.283576 Maximum: 1.359384 Average: 0.7171 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (934, 497)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1136, 400), (1214, 485)) : 2
The minimum distance from car: 3 is 11.0
bbox:car_number ((1217, 419), (1272, 472)) : 3
The minimum distance from car: 3 is 7.280109889280518
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (934, 497))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 400), (1276, 477))]
Length of task list: 5
Number of processes used: 3
```

83% | 1045/1261 [46:38<09:38, 2.68s/it]

The times for each task are: [0.857632, 0.668632, 1.355453, 0.392284, 0.33764] with:

Minimum: 0.33764 Maximum: 1.355453 Average: 0.7223 seconds

```
Number of cars 4
bbox:car_number ((835, 400), (934, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1200, 485)) : 2
The minimum distance from car: 3 is 19.0
bbox:car_number ((1204, 400), (1214, 485)) : 3
bbox:car_number ((1217, 415), (1276, 477)) : 4
The minimum distance from car: 3 is 2.23606797749979
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1217, 415))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 400), (1276, 477))]
Length of task list: 5
Number of processes used: 3
```

83% | 1046/1261 [46:41<09:35, 2.68s/it]

The times for each task are: [0.609007, 0.884005, 1.421344, 0.397478, 0.340616] with:

Minimum: 0.340616 Maximum: 1.421344 Average: 0.7305 seconds

```
Number of cars 4
bbox:car_number ((834, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1129, 400), (1212, 485)) : 2
The minimum distance from car: 3 is 14.0
bbox:car_number ((1090, 427), (1092, 447)) : 3
bbox:car_number ((1225, 430), (1257, 447)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1217, 415))]
Length of task list: 5
Number of processes used: 3
```

83% | 1047/1261 [46:44<09:33, 2.68s/it]

The times for each task are: [0.602947, 1.044705, 1.419939, 0.364021, 0.497116] with:

Minimum: 0.364021 Maximum: 1.419939 Average: 0.7857 seconds

Number of cars 3

```
bbox:car_number ((834, 400), (934, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1136, 400), (1211, 485)) : 2
The minimum distance from car: 3 is 3.0
bbox:car_number ((1217, 419), (1276, 459)) : 3
The minimum distance from car: 3 is 7.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1217, 415))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 400), (1276, 477))]
Length of task list: 5
Number of processes used: 3
```

83%| | 1048/1261 [46:47<09:30, 2.68s/it]

The times for each task are: [0.649435, 0.921894, 1.511403, 0.444815, 0.336655] with:

Minimum: 0.336655 Maximum: 1.511403 Average: 0.7728 seconds

```
Number of cars 3
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1136, 400), (1211, 472)) : 2
The minimum distance from car: 3 is 6.0
bbox:car_number ((1225, 430), (1272, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1217, 415))]
Length of task list: 5
Number of processes used: 3
```

83%| | 1049/1261 [46:50<09:27, 2.68s/it]

The times for each task are: [0.861221, 1.335825, 0.403138, 0.754453, 0.325512] with:

Minimum: 0.325512 Maximum: 1.335825 Average: 0.736 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (934, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1136, 400), (1211, 485)) : 2
The minimum distance from car: 3 is 6.0
```

```
bbox:car_number ((1225, 430), (1242, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

83%| | 1050/1261 [46:52<09:25, 2.68s/it]

The times for each task are: [0.884977, 0.732667, 1.38925, 0.346646, 0.459896] with:

Minimum: 0.346646 Maximum: 1.38925 Average: 0.7627 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (932, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1136, 400), (1212, 485)) : 2
The minimum distance from car: 3 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

83%| | 1051/1261 [46:55<09:22, 2.68s/it]

The times for each task are: [0.628216, 1.618836, 1.061882, 0.402596, 0.4533] with:

Minimum: 0.402596 Maximum: 1.618836 Average: 0.833 seconds

```
Number of cars 4
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1129, 400), (1211, 472)) : 2
The minimum distance from car: 3 is 7.211102550927978
bbox:car_number ((1113, 431), (1124, 459)) : 3
bbox:car_number ((1210, 460), (1211, 485)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

83%| | 1052/1261 [46:58<09:19, 2.68s/it]

The times for each task are: [1.343773, 0.919476, 0.729004, 0.391062, 0.281247] with:

Minimum: 0.281247 Maximum: 1.343773 Average: 0.7329 seconds

Number of cars 2

bbox:car_number ((834, 400), (932, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1113, 400), (1214, 485)) : 2

The minimum distance from car: 3 is 9.219544457292887

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

84%| | 1053/1261 [47:00<09:17, 2.68s/it]

The times for each task are: [0.931706, 0.752813, 1.418302, 0.491965, 0.356656] with:

Minimum: 0.356656 Maximum: 1.418302 Average: 0.7903 seconds

Number of cars 2

bbox:car_number ((832, 400), (932, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1136, 400), (1214, 498)) : 2

The minimum distance from car: 3 is 13.892443989449804

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

84%| | 1054/1261 [47:03<09:14, 2.68s/it]

The times for each task are: [0.884682, 0.679022, 1.460635, 0.435575, 0.365606] with:

Minimum: 0.365606 Maximum: 1.460635 Average: 0.7651 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (934, 497)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1136, 400), (1214, 498)) : 2
The minimum distance from car: 3 is 0.0
bbox:car_number ((1113, 431), (1122, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

84%| | 1055/1261 [47:06<09:11, 2.68s/it]

The times for each task are: [0.976694, 0.702209, 1.459106, 0.451553, 0.296088] with:
Minimum: 0.296088 Maximum: 1.459106 Average: 0.7771 seconds

```
Number of cars 4
bbox:car_number ((834, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1129, 400), (1219, 498)) : 2
The minimum distance from car: 3 is 1.0
bbox:car_number ((1225, 400), (1242, 447)) : 3
bbox:car_number ((1113, 431), (1122, 447)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

84%| | 1056/1261 [47:09<09:09, 2.68s/it]

The times for each task are: [0.906283, 1.376003, 0.673815, 0.403457, 0.270889] with:
Minimum: 0.270889 Maximum: 1.376003 Average: 0.7261 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (932, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1129, 400), (1219, 498)) : 2
The minimum distance from car: 3 is 0.0
bbox:car_number ((1225, 400), (1257, 447)) : 3
```

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

84%| 1057/1261 [47:11<09:06, 2.68s/it]

The times for each task are: [0.649779, 1.398445, 0.919065, 0.39166, 0.279518] with:

Minimum: 0.279518 Maximum: 1.398445 Average: 0.7277 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1060, 400), (1212, 498)) : 2
The minimum distance from car: 3 is 38.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

84%| 1058/1261 [47:14<09:03, 2.68s/it]

The times for each task are: [0.608932, 0.93042, 1.677839, 0.31389, 0.405982] with:

Minimum: 0.31389 Maximum: 1.677839 Average: 0.7874 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1120, 400), (1214, 498)) : 2
The minimum distance from car: 3 is 26.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 40
Length of task list: 5
Number of processes used: 3
```

84%| | 1059/1261 [47:17<09:01, 2.68s/it]

The times for each task are: [0.878299, 0.597438, 0.451595, 1.590203, 0.288828] with:

Minimum: 0.288828 Maximum: 1.590203 Average: 0.7613 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1113, 400), (1212, 497)) : 2

The minimum distance from car: 3 is 5.0990195135927845

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

84%| | 1060/1261 [47:19<08:58, 2.68s/it]

The times for each task are: [0.930479, 1.530146, 0.639873, 0.412984, 0.359139] with:

Minimum: 0.359139 Maximum: 1.530146 Average: 0.7745 seconds

Number of cars 3

bbox:car_number ((832, 400), (932, 498)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((1129, 400), (1214, 498)) : 2

The minimum distance from car: 3 is 9.055385138137417

bbox:car_number ((1113, 415), (1124, 462)) : 3

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

84%| | 1061/1261 [47:22<08:55, 2.68s/it]

The times for each task are: [0.872667, 0.759344, 1.417759, 0.341522, 0.38724] with:

Minimum: 0.341522 Maximum: 1.417759 Average: 0.7557 seconds

Number of cars 4

bbox:car_number ((834, 400), (932, 497)) : 1

```
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1150, 415), (1214, 478)) : 2
The minimum distance from car: 3 is 11.40175425099138
bbox:car_number ((1217, 419), (1219, 459)) : 3
bbox:car_number ((1129, 438), (1139, 472)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

84%| | 1062/1261 [47:25<08:53, 2.68s/it]

The times for each task are: [1.069344, 1.364028, 0.637655, 0.285146, 0.439527] with:

Minimum: 0.285146 Maximum: 1.364028 Average: 0.7591 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1144, 400), (1219, 485)) : 2
The minimum distance from car: 3 is 4.123105625617661
bbox:car_number ((1129, 431), (1137, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

84%| | 1063/1261 [47:28<08:50, 2.68s/it]

The times for each task are: [1.371849, 1.054098, 0.724004, 0.315363, 0.376516] with:

Minimum: 0.315363 Maximum: 1.371849 Average: 0.7684 seconds

```
Number of cars 3
bbox:car_number ((820, 400), (932, 497)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((1129, 400), (1227, 498)) : 2
The minimum distance from car: 3 is 7.615773105863909
bbox:car_number ((1225, 460), (1230, 472)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

84%| | 1064/1261 [47:30<08:47, 2.68s/it]

The times for each task are: [0.646801, 1.076685, 1.485134, 0.405753, 0.352616] with:
Minimum: 0.352616 Maximum: 1.485134 Average: 0.7934 seconds

Number of cars 3
bbox:car_number ((820, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1214, 498)) : 2
The minimum distance from car: 3 is 15.0
bbox:car_number ((1102, 427), (1105, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

84%| | 1065/1261 [47:33<08:45, 2.68s/it]

The times for each task are: [0.63908, 0.968483, 1.463843, 0.426903, 0.391599] with:
Minimum: 0.391599 Maximum: 1.463843 Average: 0.778 seconds

Number of cars 2
bbox:car_number ((832, 400), (927, 495)) : 1
The minimum distance from car: 1 is 3.605551275463989
bbox:car_number ((1141, 400), (1227, 485)) : 2
The minimum distance from car: 3 is 22.135943621178654
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

85%| | 1066/1261 [47:36<08:42, 2.68s/it]

The times for each task are: [1.39704, 0.897228, 0.637955, 0.407037, 0.340619] with:

Minimum: 0.340619 Maximum: 1.39704 Average: 0.736 seconds

Number of cars 2

bbox:car_number ((813, 400), (931, 495)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1129, 400), (1219, 498)) : 2

The minimum distance from car: 3 is 12.206555615733702

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1067/1261 [47:39<08:39, 2.68s/it]

The times for each task are: [0.875134, 0.613232, 0.4098, 1.42832, 0.42079] with:

Minimum: 0.4098 Maximum: 1.42832 Average: 0.7495 seconds

Number of cars 3

bbox:car_number ((832, 400), (927, 495)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1136, 400), (1230, 495)) : 2

The minimum distance from car: 3 is 9.219544457292887

bbox:car_number ((1225, 460), (1230, 472)) : 3

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1068/1261 [47:41<08:37, 2.68s/it]

The times for each task are: [0.635017, 1.010703, 0.304673, 1.608261, 0.500065] with:

Minimum: 0.304673 Maximum: 1.608261 Average: 0.8117 seconds

Number of cars 2

bbox:car_number ((820, 400), (931, 495)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((1129, 400), (1227, 498)) : 2

```
The minimum distance from car: 3 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

85%| 1069/1261 [47:44<08:34, 2.68s/it]

```
The times for each task are: [1.343531, 0.652503, 0.919734, 0.448662, 0.296088] with:
Minimum: 0.296088 Maximum: 1.343531 Average: 0.7321 seconds
```

```
Number of cars 2
bbox:car_number ((820, 400), (927, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1129, 400), (1227, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

85%| 1070/1261 [47:47<08:31, 2.68s/it]

```
The times for each task are: [0.926708, 0.669792, 1.680118, 0.410669, 0.33427] with:
Minimum: 0.33427 Maximum: 1.680118 Average: 0.8043 seconds
```

```
Number of cars 2
bbox:car_number ((820, 400), (931, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1129, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

85%| | 1071/1261 [47:50<08:29, 2.68s/it]

The times for each task are: [1.064409, 1.507168, 0.703533, 0.413116, 0.482057] with:

Minimum: 0.413116 Maximum: 1.507168 Average: 0.8341 seconds

Number of cars 2

bbox:car_number ((820, 400), (927, 495)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((1156, 400), (1230, 485)) : 2

The minimum distance from car: 3 is 10.63014581273465

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1072/1261 [47:52<08:26, 2.68s/it]

The times for each task are: [0.613839, 0.8677, 1.467901, 0.390707, 0.299185] with:

Minimum: 0.299185 Maximum: 1.467901 Average: 0.7279 seconds

Number of cars 3

bbox:car_number ((820, 400), (927, 495)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1156, 400), (1238, 485)) : 2

The minimum distance from car: 3 is 4.0

bbox:car_number ((1144, 427), (1152, 447)) : 3

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1073/1261 [47:55<08:23, 2.68s/it]

The times for each task are: [0.642128, 0.894888, 1.476134, 0.416067, 0.320891] with:

Minimum: 0.320891 Maximum: 1.476134 Average: 0.75 seconds

Number of cars 2

bbox:car_number ((834, 400), (927, 495)) : 1

```
The minimum distance from car: 1 is 7.0
bbox:car_number ((1158, 400), (1230, 472)) : 2
The minimum distance from car: 3 is 6.708203932499369
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 40
Length of task list: 5
Number of processes used: 3
```

85%| 1074/1261 [47:57<08:21, 2.68s/it]

The times for each task are: [0.955581, 0.689622, 1.626352, 0.440657, 0.361697] with:

Minimum: 0.361697 Maximum: 1.626352 Average: 0.8148 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (917, 495)) : 1
The minimum distance from car: 1 is 6.0
bbox:car_number ((1144, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 13.152946437965905
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 40
Length of task list: 5
Number of processes used: 3
```

85%| 1075/1261 [48:00<08:18, 2.68s/it]

The times for each task are: [1.018191, 1.525539, 0.688105, 0.516497, 0.354294] with:

Minimum: 0.354294 Maximum: 1.525539 Average: 0.8205 seconds

```
Number of cars 3
bbox:car_number ((832, 400), (927, 495)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1158, 400), (1238, 472)) : 2
The minimum distance from car: 3 is 14.317821063276353
bbox:car_number ((1144, 431), (1152, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 40
Length of task list: 5
```

Number of processes used: 3

85%| | 1076/1261 [48:03<08:15, 2.68s/it]

The times for each task are: [0.642859, 0.932326, 0.431996, 1.696092, 0.350664] with:

Minimum: 0.350664 Maximum: 1.696092 Average: 0.8108 seconds

Number of cars 2

bbox:car_number ((832, 400), (927, 495)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1144, 400), (1238, 498)) : 2

The minimum distance from car: 3 is 14.7648230602334

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 497))]

Length of task list: 5

Number of processes used: 3

85%| | 1077/1261 [48:06<08:13, 2.68s/it]

The times for each task are: [0.976921, 0.655534, 1.520108, 0.443575, 0.402721] with:

Minimum: 0.402721 Maximum: 1.520108 Average: 0.7998 seconds

Number of cars 2

bbox:car_number ((832, 400), (927, 495)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1156, 400), (1241, 485)) : 2

The minimum distance from car: 3 is 9.899494936611665

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 497))]

Length of task list: 5

Number of processes used: 3

85%| | 1078/1261 [48:09<08:10, 2.68s/it]

The times for each task are: [0.606921, 0.913782, 0.408671, 1.656287, 0.281725] with:

Minimum: 0.281725 Maximum: 1.656287 Average: 0.7735 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (927, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1144, 400), (1241, 512)) : 2
The minimum distance from car: 3 is 15.231546211727817
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1079/1261 [48:11<08:07, 2.68s/it]

The times for each task are: [0.878573, 0.691228, 1.587967, 0.456484, 0.350793] with:
Minimum: 0.350793 Maximum: 1.587967 Average: 0.793 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (917, 495)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1156, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 9.899494936611665
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1080/1261 [48:13<08:04, 2.68s/it]

The times for each task are: [0.906424, 0.732499, 1.416583, 0.344821, 0.410245] with:
Minimum: 0.344821 Maximum: 1.416583 Average: 0.7621 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (917, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1144, 400), (1241, 507)) : 2
The minimum distance from car: 3 is 8.06225774829855
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
```

```
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

86%| | 1081/1261 [48:16<08:02, 2.68s/it]

The times for each task are: [0.953665, 0.688167, 1.399913, 0.43733, 0.335237] with:

Minimum: 0.335237 Maximum: 1.399913 Average: 0.7629 seconds

```
Number of cars 2  
bbox:car_number ((818, 400), (927, 495)) : 1  
The minimum distance from car: 1 is 2.0  
bbox:car_number ((1156, 400), (1241, 477)) : 2  
The minimum distance from car: 3 is 16.15549442140351  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

86%| | 1082/1261 [48:19<07:59, 2.68s/it]

The times for each task are: [0.572108, 0.892912, 0.491089, 1.581683, 0.339666] with:

Minimum: 0.339666 Maximum: 1.581683 Average: 0.7755 seconds

```
Number of cars 2  
bbox:car_number ((832, 400), (931, 495)) : 1  
The minimum distance from car: 1 is 9.0  
bbox:car_number ((1156, 400), (1241, 485)) : 2  
The minimum distance from car: 3 is 4.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

86%| | 1083/1261 [48:22<07:57, 2.68s/it]

The times for each task are: [0.629634, 0.945381, 1.473192, 0.381987, 0.313646] with:

Minimum: 0.313646 Maximum: 1.473192 Average: 0.7488 seconds

Number of cars 2
bbox:car_number ((834, 400), (931, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1144, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 8.602325267042627
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

86%| | 1084/1261 [48:25<07:54, 2.68s/it]

The times for each task are: [0.585793, 0.915316, 1.617107, 0.464381, 0.332457] with:

Minimum: 0.332457 Maximum: 1.617107 Average: 0.783 seconds

Number of cars 2
bbox:car_number ((834, 400), (917, 495)) : 1
The minimum distance from car: 1 is 7.0
bbox:car_number ((1144, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

86%| | 1085/1261 [48:27<07:51, 2.68s/it]

The times for each task are: [1.004541, 0.619372, 1.398467, 0.348697, 0.437839] with:

Minimum: 0.348697 Maximum: 1.398467 Average: 0.7618 seconds

Number of cars 2
bbox:car_number ((834, 400), (917, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1156, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1086/1261 [48:30<07:48, 2.68s/it]

The times for each task are: [0.930842, 0.755329, 1.545787, 0.347077, 0.453796] with:

Minimum: 0.347077 Maximum: 1.545787 Average: 0.8066 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (927, 485)) : 1
The minimum distance from car: 1 is 6.4031242374328485
bbox:car_number ((1144, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1087/1261 [48:33<07:46, 2.68s/it]

The times for each task are: [0.734742, 0.932916, 1.389067, 0.458726, 0.304599] with:

Minimum: 0.304599 Maximum: 1.389067 Average: 0.764 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (915, 495)) : 1
The minimum distance from car: 1 is 6.4031242374328485
bbox:car_number ((1156, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1088/1261 [48:35<07:43, 2.68s/it]

The times for each task are: [0.858581, 0.619503, 1.567237, 0.304618, 0.406579] with:

Minimum: 0.304618 Maximum: 1.567237 Average: 0.7513 seconds

Number of cars 2

bbox:car_number ((832, 400), (927, 497)) : 1

The minimum distance from car: 1 is 4.123105625617661

bbox:car_number ((1156, 400), (1242, 498)) : 2

The minimum distance from car: 3 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

86%| | 1089/1261 [48:38<07:40, 2.68s/it]

The times for each task are: [0.681458, 1.398812, 1.030435, 0.514407, 0.312505] with:

Minimum: 0.312505 Maximum: 1.398812 Average: 0.7875 seconds

Number of cars 2

bbox:car_number ((820, 400), (915, 495)) : 1

The minimum distance from car: 1 is 12.041594578792296

bbox:car_number ((1144, 400), (1242, 497)) : 2

The minimum distance from car: 3 is 6.082762530298219

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

86%| | 1090/1261 [48:41<07:38, 2.68s/it]

The times for each task are: [1.449879, 0.965811, 0.736691, 0.380736, 0.31369] with:

Minimum: 0.31369 Maximum: 1.449879 Average: 0.7694 seconds

Number of cars 2

bbox:car_number ((820, 400), (917, 497)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((1150, 400), (1242, 497)) : 2

The minimum distance from car: 3 is 3.0

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1091/1261 [48:44<07:35, 2.68s/it]

The times for each task are: [0.86762, 1.469685, 0.65999, 0.385265, 0.392364] with:

Minimum: 0.385265 Maximum: 1.469685 Average: 0.755 seconds

```
Number of cars 2
bbox:car_number ((820, 400), (915, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1156, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 8.06225774829855
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1092/1261 [48:47<07:33, 2.68s/it]

The times for each task are: [0.886584, 0.748875, 1.43434, 0.4482, 0.357984] with:

Minimum: 0.357984 Maximum: 1.43434 Average: 0.7752 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 485)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1144, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1093/1261 [48:50<07:30, 2.68s/it]

The times for each task are: [0.5913, 0.91126, 0.408188, 1.621686, 0.335722] with:

Minimum: 0.335722 Maximum: 1.621686 Average: 0.7736 seconds

Number of cars 2

bbox:car_number ((818, 400), (915, 495)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1156, 400), (1242, 498)) : 2

The minimum distance from car: 3 is 1.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

87%| | 1094/1261 [48:52<07:27, 2.68s/it]

The times for each task are: [0.63689, 0.92282, 1.425612, 0.296258, 0.503735] with:

Minimum: 0.296258 Maximum: 1.425612 Average: 0.7571 seconds

Number of cars 2

bbox:car_number ((813, 400), (915, 495)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((1156, 400), (1253, 498)) : 2

The minimum distance from car: 3 is 5.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

87%| | 1095/1261 [48:55<07:25, 2.68s/it]

The times for each task are: [0.585063, 0.911108, 1.407082, 0.39987, 0.288889] with:

Minimum: 0.288889 Maximum: 1.407082 Average: 0.7184 seconds

Number of cars 2

bbox:car_number ((813, 400), (915, 495)) : 1

The minimum distance from car: 1 is 0.0

```
bbox:car_number ((1156, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

87%| | 1096/1261 [48:58<07:22, 2.68s/it]

The times for each task are: [1.084842, 0.765518, 1.654224, 0.516196, 0.396255] with:

Minimum: 0.396255 Maximum: 1.654224 Average: 0.8834 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (915, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1144, 400), (1257, 498)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

87%| | 1097/1261 [49:00<07:19, 2.68s/it]

The times for each task are: [0.9073, 0.598782, 1.493417, 0.402623, 0.300429] with:

Minimum: 0.300429 Maximum: 1.493417 Average: 0.7405 seconds

```
Number of cars 2
bbox:car_number ((820, 400), (915, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1156, 400), (1253, 497)) : 2
The minimum distance from car: 3 is 4.123105625617661
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

87%| | 1098/1261 [49:03<07:16, 2.68s/it]

The times for each task are: [0.565208, 0.889644, 1.511427, 0.424, 0.4255] with:

Minimum: 0.424 Maximum: 1.511427 Average: 0.7632 seconds

Number of cars 2

bbox:car_number ((818, 400), (915, 495)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1156, 400), (1257, 498)) : 2

The minimum distance from car: 3 is 2.23606797749979

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

87%| | 1099/1261 [49:06<07:14, 2.68s/it]

The times for each task are: [0.913133, 0.606831, 1.612824, 0.287757, 0.492468] with:

Minimum: 0.287757 Maximum: 1.612824 Average: 0.7826 seconds

Number of cars 2

bbox:car_number ((818, 400), (915, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1156, 400), (1257, 498)) : 2

The minimum distance from car: 3 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

87%| | 1100/1261 [49:09<07:11, 2.68s/it]

The times for each task are: [0.728019, 0.823744, 1.426294, 0.316521, 0.476913] with:

Minimum: 0.316521 Maximum: 1.426294 Average: 0.7543 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1156, 400), (1257, 485)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1101/1261 [49:12<07:09, 2.68s/it]

The times for each task are: [0.896015, 0.705853, 1.388604, 0.416425, 0.363082] with:

Minimum: 0.363082 Maximum: 1.388604 Average: 0.754 seconds

```
Number of cars 2
bbox:car_number ((820, 400), (915, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1160, 400), (1257, 498)) : 2
The minimum distance from car: 3 is 7.280109889280518
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1102/1261 [49:14<07:06, 2.68s/it]

The times for each task are: [0.629209, 1.506808, 0.888627, 0.376419, 0.404491] with:

Minimum: 0.376419 Maximum: 1.506808 Average: 0.7611 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (912, 495)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1156, 400), (1257, 498)) : 2
The minimum distance from car: 3 is 2.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
```

```
Length of task list: 5
Number of processes used: 3
```

```
87%| | 1103/1261 [49:17<07:03, 2.68s/it]
```

```
The times for each task are: [1.011824, 1.486883, 0.758928, 0.31052, 0.427342] with:
```

```
Minimum: 0.31052 Maximum: 1.486883 Average: 0.7991 seconds
```

```
Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1158, 400), (1257, 485)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

```
88%| | 1104/1261 [49:20<07:00, 2.68s/it]
```

```
The times for each task are: [0.908469, 1.371039, 0.293647, 0.392001, 0.713377] with:
```

```
Minimum: 0.293647 Maximum: 1.371039 Average: 0.7357 seconds
```

```
Number of cars 2
bbox:car_number ((818, 400), (912, 485)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1144, 400), (1257, 498)) : 2
The minimum distance from car: 3 is 9.899494936611665
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

```
88%| | 1105/1261 [49:22<06:58, 2.68s/it]
```

```
The times for each task are: [0.603804, 0.890079, 1.567056, 0.367905, 0.505338] with:
```

Minimum: 0.367905 Maximum: 1.567056 Average: 0.7868 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1165, 423), (1242, 495)) : 2
The minimum distance from car: 3 is 10.44030650891055
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88%| | 1106/1261 [49:25<06:55, 2.68s/it]

The times for each task are: [0.88237, 0.609792, 0.430579, 1.645565, 0.398417] with:

Minimum: 0.398417 Maximum: 1.645565 Average: 0.7933 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1156, 400), (1257, 507)) : 2
The minimum distance from car: 3 is 6.708203932499369
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88%| | 1107/1261 [49:28<06:52, 2.68s/it]

The times for each task are: [0.925097, 0.685083, 1.439471, 0.34961, 0.466554] with:

Minimum: 0.34961 Maximum: 1.439471 Average: 0.7732 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1175, 400), (1257, 485)) : 2
The minimum distance from car: 3 is 14.866068747318506
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88% | 1108/1261 [49:30<06:50, 2.68s/it]

The times for each task are: [0.695401, 1.522316, 0.982392, 0.332247, 0.422823] with:

Minimum: 0.332247 Maximum: 1.522316 Average: 0.791 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (908, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1175, 400), (1268, 498)) : 2
The minimum distance from car: 3 is 8.602325267042627
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88% | 1109/1261 [49:33<06:47, 2.68s/it]

The times for each task are: [0.908599, 0.627148, 1.435524, 0.406688, 0.409422] with:

Minimum: 0.406688 Maximum: 1.435524 Average: 0.7575 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1183, 400), (1257, 485)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88% | 1110/1261 [49:36<06:44, 2.68s/it]

The times for each task are: [0.591985, 1.373566, 1.029658, 0.425582, 0.393191] with:

Minimum: 0.393191 Maximum: 1.373566 Average: 0.7628 seconds

Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1175, 400), (1272, 485)) : 2
The minimum distance from car: 3 is 3.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

88%| | 1111/1261 [49:38<06:42, 2.68s/it]

The times for each task are: [1.431415, 0.649884, 0.940463, 0.40755, 0.307363] with:

Minimum: 0.307363 Maximum: 1.431415 Average: 0.7473 seconds

Number of cars 2
bbox:car_number ((813, 400), (915, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1198, 400), (1268, 472)) : 2
The minimum distance from car: 3 is 11.661903789690601
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

88%| | 1112/1261 [49:41<06:39, 2.68s/it]

The times for each task are: [0.697878, 0.937484, 0.317159, 0.48759, 1.391236] with:

Minimum: 0.317159 Maximum: 1.391236 Average: 0.7663 seconds

Number of cars 2
bbox:car_number ((820, 400), (915, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1144, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 28.178005607210743

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

88%| 1113/1261 [49:44<06:36, 2.68s/it]

The times for each task are: [0.87493, 1.481395, 0.689455, 0.300105, 0.463222] with:

Minimum: 0.300105 Maximum: 1.481395 Average: 0.7618 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (915, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1175, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 16.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

88%| 1114/1261 [49:47<06:34, 2.68s/it]

The times for each task are: [0.65252, 1.337935, 1.073444, 0.398241, 0.286742] with:

Minimum: 0.286742 Maximum: 1.337935 Average: 0.7498 seconds

```
Number of cars 4
bbox:car_number ((813, 400), (912, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1103, 400), (1105, 459)) : 2
bbox:car_number ((1175, 400), (1273, 485)) : 3
The minimum distance from car: 3 is 7.0
bbox:car_number ((1156, 427), (1162, 459)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

88%| | 1115/1261 [49:49<06:31, 2.68s/it]

The times for each task are: [0.892514, 1.353342, 0.711241, 0.357628, 0.412708] with:

Minimum: 0.357628 Maximum: 1.353342 Average: 0.7455 seconds

Number of cars 2

bbox:car_number ((813, 400), (912, 495)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1156, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 12.206555615733702

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

89%| | 1116/1261 [49:52<06:28, 2.68s/it]

The times for each task are: [0.947394, 0.64411, 1.460767, 0.542364, 0.361537] with:

Minimum: 0.361537 Maximum: 1.460767 Average: 0.7912 seconds

Number of cars 3

bbox:car_number ((832, 400), (908, 495)) : 1

The minimum distance from car: 1 is 8.0

bbox:car_number ((1103, 400), (1107, 459)) : 2

bbox:car_number ((1120, 400), (1273, 512)) : 3

The minimum distance from car: 3 is 19.313207915827967

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

89%| | 1117/1261 [49:55<06:26, 2.68s/it]

The times for each task are: [0.665184, 0.961164, 1.550093, 0.425469, 0.359527] with:

Minimum: 0.359527 Maximum: 1.550093 Average: 0.7923 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (908, 497)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((1156, 400), (1273, 512)) : 2
The minimum distance from car: 3 is 18.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| 1118/1261 [49:57<06:23, 2.68s/it]

The times for each task are: [0.970828, 1.501631, 0.715932, 0.413218, 0.329625] with:

Minimum: 0.329625 Maximum: 1.501631 Average: 0.7862 seconds

```
Number of cars 3
bbox:car_number ((820, 400), (908, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1175, 400), (1273, 497)) : 2
The minimum distance from car: 3 is 12.806248474865697
bbox:car_number ((1160, 438), (1167, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| 1119/1261 [50:01<06:20, 2.68s/it]

The times for each task are: [0.626181, 1.530342, 1.016358, 0.477497, 0.397917] with:

Minimum: 0.397917 Maximum: 1.530342 Average: 0.8097 seconds

```
Number of cars 3
bbox:car_number ((820, 400), (908, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1175, 400), (1272, 478)) : 2
The minimum distance from car: 3 is 9.055385138137417
bbox:car_number ((1122, 415), (1122, 447)) : 3
totalCars: 11
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))  
Length of task list: 5  
Number of processes used: 3
```

89%| | 1120/1261 [50:03<06:18, 2.68s/it]

The times for each task are: [0.623348, 1.022156, 1.563204, 0.519207, 0.321853] with:

Minimum: 0.321853 Maximum: 1.563204 Average: 0.81 seconds

```
Number of cars 2  
bbox:car_number ((820, 400), (912, 495)) : 1  
The minimum distance from car: 1 is 2.0  
bbox:car_number ((1175, 400), (1273, 478)) : 2  
The minimum distance from car: 3 is 1.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))  
Length of task list: 5  
Number of processes used: 3
```

89%| | 1121/1261 [50:06<06:15, 2.68s/it]

The times for each task are: [0.954237, 1.574679, 0.720138, 0.469219, 0.364668] with:

Minimum: 0.364668 Maximum: 1.574679 Average: 0.8166 seconds

```
Number of cars 2  
bbox:car_number ((813, 400), (908, 495)) : 1  
The minimum distance from car: 1 is 6.0  
bbox:car_number ((1183, 400), (1273, 485)) : 2  
The minimum distance from car: 3 is 5.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))  
Length of task list: 5  
Number of processes used: 3
```

89%| | 1122/1261 [50:08<06:12, 2.68s/it]

The times for each task are: [1.052366, 0.617215, 1.5502, 0.300953, 0.51288] with:

Minimum: 0.300953 Maximum: 1.5502 Average: 0.8067 seconds

Number of cars 2

bbox:car_number ((813, 400), (908, 485)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1160, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 13.892443989449804

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

89% | 1123/1261 [50:11<06:10, 2.68s/it]

The times for each task are: [0.930722, 0.665729, 1.667247, 0.316948, 0.453208] with:

Minimum: 0.316948 Maximum: 1.667247 Average: 0.8068 seconds

Number of cars 2

bbox:car_number ((832, 400), (901, 485)) : 1

The minimum distance from car: 1 is 6.0

bbox:car_number ((1180, 400), (1273, 477)) : 2

The minimum distance from car: 3 is 14.866068747318506

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

89% | 1124/1261 [50:14<06:07, 2.68s/it]

The times for each task are: [0.67501, 0.924132, 1.596298, 0.410538, 0.305325] with:

Minimum: 0.305325 Maximum: 1.596298 Average: 0.7823 seconds

Number of cars 3

bbox:car_number ((813, 400), (901, 485)) : 1

The minimum distance from car: 1 is 9.0

bbox:car_number ((1156, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 16.278820596099706

```
bbox:car_number ((1144, 431), (1152, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| 1125/1261 [50:17<06:04, 2.68s/it]

The times for each task are: [0.892546, 0.753175, 1.36986, 0.400901, 0.30652] with:

Minimum: 0.30652 Maximum: 1.36986 Average: 0.7446 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (901, 485)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1180, 400), (1272, 477)) : 2
The minimum distance from car: 3 is 16.278820596099706
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| 1126/1261 [50:19<06:02, 2.68s/it]

The times for each task are: [0.915462, 0.625378, 0.417048, 1.662486, 0.306072] with:

Minimum: 0.306072 Maximum: 1.662486 Average: 0.7853 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (901, 485)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1144, 400), (1273, 507)) : 2
The minimum distance from car: 3 is 23.430749027719962
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| | 1127/1261 [50:22<05:59, 2.68s/it]

The times for each task are: [0.929592, 0.707183, 1.438095, 0.418485, 0.350453] with:

Minimum: 0.350453 Maximum: 1.438095 Average: 0.7688 seconds

Number of cars 2

bbox:car_number ((813, 400), (901, 485)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1179, 400), (1273, 485)) : 2

The minimum distance from car: 3 is 20.223748416156685

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

89%| | 1128/1261 [50:25<05:56, 2.68s/it]

The times for each task are: [0.663255, 1.024572, 1.525734, 0.441134, 0.305401] with:

Minimum: 0.305401 Maximum: 1.525734 Average: 0.792 seconds

Number of cars 2

bbox:car_number ((820, 400), (901, 485)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((1183, 400), (1273, 472)) : 2

The minimum distance from car: 3 is 6.324555320336759

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

90%| | 1129/1261 [50:28<05:54, 2.68s/it]

The times for each task are: [1.023533, 0.695369, 1.449007, 0.442376, 0.36438] with:

Minimum: 0.36438 Maximum: 1.449007 Average: 0.7949 seconds

Number of cars 2

bbox:car_number ((813, 400), (901, 485)) : 1

The minimum distance from car: 1 is 3.0

```
bbox:car_number ((1175, 400), (1273, 478)) : 2
The minimum distance from car: 3 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1130/1261 [50:30<05:51, 2.68s/it]

The times for each task are: [0.914309, 1.467371, 0.730077, 0.320746, 0.45838] with:

Minimum: 0.320746 Maximum: 1.467371 Average: 0.7782 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (901, 485)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1179, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 10.198039027185569
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1131/1261 [50:33<05:48, 2.68s/it]

The times for each task are: [0.674323, 0.907297, 1.38291, 0.440543, 0.299382] with:

Minimum: 0.299382 Maximum: 1.38291 Average: 0.7409 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (908, 495)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1183, 400), (1273, 497)) : 2
The minimum distance from car: 3 is 2.23606797749979
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1132/1261 [50:36<05:46, 2.68s/it]

The times for each task are: [1.024513, 0.647622, 1.441945, 0.478324, 0.317488] with:

Minimum: 0.317488 Maximum: 1.441945 Average: 0.782 seconds

Number of cars 2

bbox:car_number ((805, 400), (901, 495)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1175, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 4.123105625617661

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

90%| | 1133/1261 [50:39<05:43, 2.68s/it]

The times for each task are: [0.632883, 1.404105, 0.926725, 0.324644, 0.44968] with:

Minimum: 0.324644 Maximum: 1.404105 Average: 0.7476 seconds

Number of cars 2

bbox:car_number ((820, 400), (901, 485)) : 1

The minimum distance from car: 1 is 8.602325267042627

bbox:car_number ((1217, 430), (1268, 459)) : 2

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

90%| | 1134/1261 [50:42<05:40, 2.68s/it]

The times for each task are: [0.922404, 1.42887, 0.676428, 0.454495, 0.346879] with:

Minimum: 0.346879 Maximum: 1.42887 Average: 0.7658 seconds

Number of cars 2

bbox:car_number ((805, 400), (897, 495)) : 1

The minimum distance from car: 1 is 10.295630140987

```
bbox:car_number ((1179, 430), (1241, 497)) : 2
The minimum distance from car: 3 is 10.198039027185569
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1135/1261 [50:44<05:38, 2.68s/it]

The times for each task are: [0.583851, 1.376988, 0.829059, 0.313624, 0.435479] with:

Minimum: 0.313624 Maximum: 1.376988 Average: 0.7078 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (901, 485)) : 1
The minimum distance from car: 1 is 7.810249675906654
bbox:car_number ((1204, 419), (1273, 478)) : 2
The minimum distance from car: 3 is 14.035668847618199
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1136/1261 [50:47<05:35, 2.68s/it]

The times for each task are: [1.316985, 0.864663, 0.678266, 0.299375, 0.372615] with:

Minimum: 0.299375 Maximum: 1.316985 Average: 0.7064 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (897, 485)) : 1
The minimum distance from car: 1 is 6.0
bbox:car_number ((1183, 427), (1268, 485)) : 2
The minimum distance from car: 3 is 15.264337522473747
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1137/1261 [50:50<05:32, 2.68s/it]

The times for each task are: [0.603167, 0.813225, 1.654233, 0.544051, 0.291559] with:

Minimum: 0.291559 Maximum: 1.654233 Average: 0.7812 seconds

Number of cars 2

bbox:car_number ((805, 400), (901, 495)) : 1

The minimum distance from car: 1 is 5.385164807134504

bbox:car_number ((1204, 431), (1268, 472)) : 2

The minimum distance from car: 3 is 12.083045973594572

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

90%| | 1138/1261 [50:52<05:29, 2.68s/it]

The times for each task are: [1.041133, 0.64824, 1.69149, 0.359509, 0.45048] with:

Minimum: 0.359509 Maximum: 1.69149 Average: 0.8382 seconds

Number of cars 2

bbox:car_number ((805, 400), (897, 478)) : 1

The minimum distance from car: 1 is 8.246211251235321

bbox:car_number ((1175, 431), (1268, 497)) : 2

The minimum distance from car: 3 is 11.045361017187261

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

90%| | 1139/1261 [50:55<05:27, 2.68s/it]

The times for each task are: [0.748321, 0.892138, 1.36838, 0.417169, 0.291895] with:

Minimum: 0.291895 Maximum: 1.36838 Average: 0.7436 seconds

```
Number of cars 3
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1204, 430), (1273, 498)) : 2
The minimum distance from car: 3 is 13.152946437965905
bbox:car_number ((1179, 438), (1185, 472)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1140/1261 [50:58<05:24, 2.68s/it]

The times for each task are: [0.575269, 0.882216, 1.422439, 0.403615, 0.395494] with:

Minimum: 0.395494 Maximum: 1.422439 Average: 0.7358 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1175, 427), (1273, 516)) : 2
The minimum distance from car: 3 is 7.615773105863909
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1141/1261 [51:01<05:21, 2.68s/it]

The times for each task are: [0.56465, 0.897192, 1.369906, 0.411074, 0.297768] with:

Minimum: 0.297768 Maximum: 1.369906 Average: 0.7081 seconds

```
Number of cars 3
bbox:car_number ((803, 400), (897, 485)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1175, 423), (1268, 498)) : 2
The minimum distance from car: 3 is 11.40175425099138
bbox:car_number ((1156, 431), (1167, 462)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91% | 1142/1261 [51:03<05:19, 2.68s/it]

The times for each task are: [1.336064, 0.661924, 0.975848, 0.495468, 0.405752] with:

Minimum: 0.405752 Maximum: 1.336064 Average: 0.775 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (897, 485)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1175, 427), (1272, 512)) : 2
The minimum distance from car: 3 is 9.219544457292887
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91% | 1143/1261 [51:06<05:16, 2.68s/it]

The times for each task are: [0.613071, 1.420613, 0.907501, 0.408004, 0.276912] with:

Minimum: 0.276912 Maximum: 1.420613 Average: 0.7252 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (897, 485)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1183, 430), (1272, 498)) : 2
The minimum distance from car: 3 is 6.4031242374328485
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91% | 1144/1261 [51:09<05:13, 2.68s/it]

The times for each task are: [0.612421, 0.890314, 1.635937, 0.411348, 0.306767] with:

Minimum: 0.306767 Maximum: 1.635937 Average: 0.7714 seconds

Number of cars 2

bbox:car_number ((799, 400), (897, 495)) : 1

The minimum distance from car: 1 is 5.830951894845301

bbox:car_number ((1183, 460), (1227, 507)) : 2

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

91% | 1145/1261 [51:11<05:11, 2.68s/it]

The times for each task are: [0.899451, 0.7104, 1.449049, 0.414128, 0.333894] with:

Minimum: 0.333894 Maximum: 1.449049 Average: 0.7614 seconds

Number of cars 2

bbox:car_number ((805, 400), (896, 478)) : 1

The minimum distance from car: 1 is 8.246211251235321

bbox:car_number ((1204, 460), (1227, 472)) : 2

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

91% | 1146/1261 [51:14<05:08, 2.68s/it]

The times for each task are: [0.847835, 1.376862, 0.675788, 0.293288, 0.40079] with:

Minimum: 0.293288 Maximum: 1.376862 Average: 0.7189 seconds

Number of cars 2

bbox:car_number ((805, 400), (897, 485)) : 1

The minimum distance from car: 1 is 3.1622776601683795

bbox:car_number ((1204, 430), (1268, 477)) : 2

The minimum distance from car: 3 is 11.180339887498949

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

91%| 1147/1261 [51:16<05:05, 2.68s/it]

The times for each task are: [1.35425, 1.050977, 0.758662, 0.300502, 0.388245] with:

Minimum: 0.300502 Maximum: 1.35425 Average: 0.7705 seconds

Number of cars 2

bbox:car_number ((803, 400), (897, 495)) : 1

The minimum distance from car: 1 is 5.0990195135927845

bbox:car_number ((1183, 427), (1268, 485)) : 2

The minimum distance from car: 3 is 8.246211251235321

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

91%| 1148/1261 [51:19<05:03, 2.68s/it]

The times for each task are: [0.687902, 0.896432, 1.355219, 0.523454, 0.33935] with:

Minimum: 0.33935 Maximum: 1.355219 Average: 0.7605 seconds

Number of cars 4

bbox:car_number ((805, 400), (885, 485)) : 1

The minimum distance from car: 1 is 7.0710678118654755

bbox:car_number ((1195, 427), (1268, 478)) : 2

The minimum distance from car: 3 is 5.0990195135927845

bbox:car_number ((595, 460), (627, 492)) : 3

bbox:car_number ((640, 460), (647, 497)) : 4

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

91%| 1149/1261 [51:22<05:00, 2.68s/it]

The times for each task are: [1.365772, 0.691848, 0.911139, 0.345125, 0.391847] with:

Minimum: 0.345125 Maximum: 1.365772 Average: 0.7411 seconds

```
Number of cars 2
bbox:car_number ((803, 400), (885, 478)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((1195, 427), (1268, 478)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

91%| 1150/1261 [51:24<04:57, 2.68s/it]

The times for each task are: [0.874134, 1.626371, 0.651907, 0.316154, 0.446804] with:

Minimum: 0.316154 Maximum: 1.626371 Average: 0.7831 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (890, 478)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1183, 427), (1272, 497)) : 2
The minimum distance from car: 3 is 6.324555320336759
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91%|| 1151/1261 [51:27<04:55, 2.68s/it]

The times for each task are: [0.927533, 0.721937, 1.494551, 0.529916, 0.343153] with:

Minimum: 0.343153 Maximum: 1.494551 Average: 0.8034 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1179, 427), (1268, 497)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91%|| 1152/1261 [51:30<04:52, 2.68s/it]

The times for each task are: [1.420006, 0.923534, 0.639086, 0.319959, 0.404215] with:

Minimum: 0.319959 Maximum: 1.420006 Average: 0.7414 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (896, 485)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1181, 427), (1268, 507)) : 2
The minimum distance from car: 3 is 5.0990195135927845
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91%|| 1153/1261 [51:33<04:49, 2.68s/it]

The times for each task are: [0.887633, 0.668151, 1.411217, 0.390781, 0.310189] with:

Minimum: 0.310189 Maximum: 1.411217 Average: 0.7336 seconds

```
Number of cars 1
bbox:car_number ((803, 400), (896, 495)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1154/1261 [51:36<04:47, 2.68s/it]

The times for each task are: [0.881982, 1.345169, 0.612221, 0.501062, 0.338163] with:

Minimum: 0.338163 Maximum: 1.345169 Average: 0.7357 seconds

```
Number of cars 1
bbox:car_number ((803, 400), (885, 478)) : 1
The minimum distance from car: 1 is 9.433981132056603
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1155/1261 [51:38<04:44, 2.68s/it]

The times for each task are: [0.658787, 1.464549, 1.080629, 0.30441, 0.468874] with:
Minimum: 0.30441 Maximum: 1.464549 Average: 0.7954 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1156/1261 [51:41<04:41, 2.68s/it]

The times for each task are: [0.959269, 0.657849, 1.608155, 0.401354, 0.299093] with:
Minimum: 0.299093 Maximum: 1.608155 Average: 0.7851 seconds

```
Number of cars 1
bbox:car_number ((803, 400), (885, 485)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1157/1261 [51:44<04:39, 2.68s/it]

The times for each task are: [1.362386, 0.661017, 0.94236, 0.423364, 0.344031] with:
Minimum: 0.344031 Maximum: 1.362386 Average: 0.7466 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 478)) : 1

The minimum distance from car: 1 is 4.242640687119285

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

92%|| 1158/1261 [51:46<04:36, 2.68s/it]

The times for each task are: [1.409586, 0.682418, 0.980272, 0.276021, 0.431997] with:

Minimum: 0.276021 Maximum: 1.409586 Average: 0.7561 seconds

Number of cars 1

bbox:car_number ((799, 400), (890, 485)) : 1

The minimum distance from car: 1 is 4.242640687119285

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

92%|| 1159/1261 [51:49<04:33, 2.68s/it]

The times for each task are: [0.890516, 1.384851, 0.696486, 0.393175, 0.29366] with:

Minimum: 0.29366 Maximum: 1.384851 Average: 0.7317 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 477)) : 1

The minimum distance from car: 1 is 6.4031242374328485

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

92%|| 1160/1261 [51:52<04:30, 2.68s/it]

The times for each task are: [0.850934, 1.321809, 0.747798, 0.335995, 0.46484] with:

Minimum: 0.335995 Maximum: 1.321809 Average: 0.7443 seconds

Number of cars 1

bbox:car_number ((820, 415), (882, 477)) : 1

```
The minimum distance from car: 1 is 8.246211251235321
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1161/1261 [51:54<04:28, 2.68s/it]

The times for each task are: [0.807572, 1.386878, 0.59317, 0.396675, 0.286712] with:

Minimum: 0.286712 Maximum: 1.386878 Average: 0.6942 seconds

```
Number of cars 3
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 8.246211251235321
bbox:car_number ((1183, 454), (1185, 495)) : 2
bbox:car_number ((1210, 454), (1257, 477)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1162/1261 [51:57<04:25, 2.68s/it]

The times for each task are: [0.625022, 0.936203, 1.433247, 0.374376, 0.513533] with:

Minimum: 0.374376 Maximum: 1.433247 Average: 0.7765 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1210, 430), (1272, 507)) : 2
The minimum distance from car: 3 is 17.029386365926403
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1163/1261 [52:00<04:22, 2.68s/it]

The times for each task are: [0.663414, 1.044445, 0.359807, 0.415685, 1.456828] with:

Minimum: 0.359807 Maximum: 1.456828 Average: 0.788 seconds

Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

92%|| 1164/1261 [52:03<04:20, 2.68s/it]

The times for each task are: [0.903256, 0.69737, 1.443594, 0.398001, 0.42073] with:

Minimum: 0.398001 Maximum: 1.443594 Average: 0.7726 seconds

Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

92%|| 1165/1261 [52:06<04:17, 2.68s/it]

The times for each task are: [0.911063, 1.438896, 0.716478, 0.421633, 0.349729] with:

Minimum: 0.349729 Maximum: 1.438896 Average: 0.7676 seconds

Number of cars 1
bbox:car_number ((805, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

92%|| 1166/1261 [52:08<04:14, 2.68s/it]

The times for each task are: [0.877108, 0.663715, 1.506131, 0.529801, 0.352953] with:

Minimum: 0.352953 Maximum: 1.506131 Average: 0.7859 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1167/1261 [52:11<04:12, 2.68s/it]

The times for each task are: [1.378278, 0.83631, 0.765089, 0.419702, 0.376083] with:

Minimum: 0.376083 Maximum: 1.378278 Average: 0.7551 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1168/1261 [52:14<04:09, 2.68s/it]

The times for each task are: [0.859481, 0.665746, 1.397795, 0.474481, 0.331534] with:

Minimum: 0.331534 Maximum: 1.397795 Average: 0.7458 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 472)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1169/1261 [52:17<04:06, 2.68s/it]

The times for each task are: [1.543787, 0.911646, 0.691496, 0.343031, 0.454709] with:

Minimum: 0.343031 Maximum: 1.543787 Average: 0.7889 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 11.180339887498949
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1170/1261 [52:19<04:04, 2.68s/it]

```
The times for each task are: [0.907001, 0.718455, 1.461976, 0.391555, 0.348499] with:
Minimum: 0.348499 Maximum: 1.461976 Average: 0.7655 seconds
```

```
Number of cars 2
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 5.385164807134504
bbox:car_number ((1225, 460), (1257, 477)) : 2
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1171/1261 [52:22<04:01, 2.68s/it]

```
The times for each task are: [1.547721, 0.84841, 0.702104, 0.344303, 0.455974] with:
Minimum: 0.344303 Maximum: 1.547721 Average: 0.7797 seconds
```

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1172/1261 [52:25<03:58, 2.68s/it]

```
The times for each task are: [0.848601, 0.719663, 1.602879, 0.483668, 0.31832] with:
Minimum: 0.31832 Maximum: 1.602879 Average: 0.7946 seconds
```

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1173/1261 [52:28<03:56, 2.68s/it]

The times for each task are: [0.893318, 0.62519, 1.65149, 0.419428, 0.319058] with:

Minimum: 0.319058 Maximum: 1.65149 Average: 0.7817 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 485)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1174/1261 [52:30<03:53, 2.68s/it]

The times for each task are: [1.031925, 0.689898, 1.465316, 0.449119, 0.394337] with:

Minimum: 0.394337 Maximum: 1.465316 Average: 0.8061 seconds

```
Number of cars 1
bbox:car_number ((803, 400), (890, 497)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1175/1261 [52:33<03:50, 2.68s/it]

The times for each task are: [0.644108, 1.460283, 0.907615, 0.303685, 0.516864] with:

Minimum: 0.303685 Maximum: 1.460283 Average: 0.7665 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 495)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

93%|| 1176/1261 [52:36<03:48, 2.68s/it]

The times for each task are: [0.923024, 0.616371, 1.483411, 0.408189, 0.349582] with:

Minimum: 0.349582 Maximum: 1.483411 Average: 0.7561 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 478)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

93%|| 1177/1261 [52:38<03:45, 2.68s/it]

The times for each task are: [1.037777, 0.597519, 1.411747, 0.34549, 0.447333] with:

Minimum: 0.34549 Maximum: 1.411747 Average: 0.768 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 478)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

93%|| 1178/1261 [52:41<03:42, 2.68s/it]

The times for each task are: [0.681893, 0.90325, 1.612835, 0.289065, 0.446206] with:

Minimum: 0.289065 Maximum: 1.612835 Average: 0.7866 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 477)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

93%|| 1179/1261 [52:44<03:40, 2.68s/it]

The times for each task are: [0.908748, 0.708334, 1.388671, 0.388986, 0.325228] with:

Minimum: 0.325228 Maximum: 1.388671 Average: 0.744 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 478)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1180/1261 [52:46<03:37, 2.68s/it]

The times for each task are: [1.369947, 0.959189, 0.653415, 0.299835, 0.383038] with:

Minimum: 0.299835 Maximum: 1.369947 Average: 0.7331 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 477)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1181/1261 [52:49<03:34, 2.68s/it]

The times for each task are: [0.699691, 0.990778, 1.390466, 0.34773, 0.473183] with:

Minimum: 0.34773 Maximum: 1.390466 Average: 0.7804 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 477)) : 1

The minimum distance from car: 1 is 0.0

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

94%|| 1182/1261 [52:52<03:32, 2.68s/it]

The times for each task are: [0.887752, 0.601211, 0.308964, 1.645232, 0.444016] with:

Minimum: 0.308964 Maximum: 1.645232 Average: 0.7774 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

94%|| 1183/1261 [52:54<03:29, 2.68s/it]

The times for each task are: [0.674712, 1.497717, 0.980058, 0.362994, 0.408675] with:

Minimum: 0.362994 Maximum: 1.497717 Average: 0.7848 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

94%|| 1184/1261 [52:57<03:26, 2.68s/it]

The times for each task are: [0.665552, 0.944401, 1.423489, 0.36254, 0.522282] with:

Minimum: 0.36254 Maximum: 1.423489 Average: 0.7837 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (890, 492)) : 1
The minimum distance from car: 1 is 8.246211251235321
totalCars: 11
```

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

94%|| 1185/1261 [53:00<03:23, 2.68s/it]

The times for each task are: [0.999898, 0.700082, 1.699938, 0.373683, 0.551516] with:

Minimum: 0.373683 Maximum: 1.699938 Average: 0.865 seconds

Number of cars 2
bbox:car_number ((805, 415), (890, 485)) : 1
The minimum distance from car: 1 is 5.656854249492381
bbox:car_number ((1165, 415), (1227, 462)) : 2
The minimum distance from car: 3 is 37.69615364994153
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3

94%|| 1186/1261 [53:03<03:21, 2.68s/it]

The times for each task are: [0.630712, 0.933641, 1.737084, 0.292083, 0.402328] with:

Minimum: 0.292083 Maximum: 1.737084 Average: 0.7992 seconds

Number of cars 1
bbox:car_number ((805, 400), (890, 492)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

94%|| 1187/1261 [53:06<03:18, 2.68s/it]

The times for each task are: [0.953198, 1.417057, 0.723992, 0.431109, 0.383294] with:

Minimum: 0.383294 Maximum: 1.417057 Average: 0.7817 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 492)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1188/1261 [53:08<03:15, 2.68s/it]

The times for each task are: [0.610864, 0.977646, 0.413749, 1.624668, 0.323712] with:

Minimum: 0.323712 Maximum: 1.624668 Average: 0.7901 seconds

Number of cars 1

bbox:car_number ((813, 415), (885, 477)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1189/1261 [53:11<03:13, 2.68s/it]

The times for each task are: [0.617234, 0.938822, 1.458769, 0.448952, 0.339146] with:

Minimum: 0.339146 Maximum: 1.458769 Average: 0.7606 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 2.23606797749979

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1190/1261 [53:14<03:10, 2.68s/it]

The times for each task are: [0.569648, 0.454797, 0.970721, 1.41796, 0.394247] with:

Minimum: 0.394247 Maximum: 1.41796 Average: 0.7615 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 485)) : 1

```
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

94%|| 1191/1261 [53:17<03:07, 2.68s/it]

The times for each task are: [0.898956, 1.444783, 0.760113, 0.30353, 0.437234] with:

Minimum: 0.30353 Maximum: 1.444783 Average: 0.7689 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1192/1261 [53:19<03:05, 2.68s/it]

The times for each task are: [1.538858, 0.962017, 0.659041, 0.40431, 0.346258] with:

Minimum: 0.346258 Maximum: 1.538858 Average: 0.7821 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (890, 485)) : 1
The minimum distance from car: 1 is 6.4031242374328485
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1193/1261 [53:22<03:02, 2.68s/it]

The times for each task are: [0.628732, 0.91217, 1.552298, 0.313228, 0.413486] with:

Minimum: 0.313228 Maximum: 1.552298 Average: 0.764 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (890, 485)) : 1
The minimum distance from car: 1 is 0.0
```

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1194/1261 [53:24<02:59, 2.68s/it]

The times for each task are: [0.882593, 1.481992, 0.754427, 0.300505, 0.399523] with:

Minimum: 0.300505 Maximum: 1.481992 Average: 0.7638 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1195/1261 [53:27<02:57, 2.68s/it]

The times for each task are: [0.920092, 1.414224, 0.687545, 0.413872, 0.386506] with:

Minimum: 0.386506 Maximum: 1.414224 Average: 0.7644 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1196/1261 [53:30<02:54, 2.68s/it]

The times for each task are: [0.611529, 1.059907, 1.528123, 0.465571, 0.430727] with:

Minimum: 0.430727 Maximum: 1.528123 Average: 0.8192 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 498)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 11
```

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

95%|| 1197/1261 [53:32<02:51, 2.68s/it]

The times for each task are: [0.578365, 1.414687, 1.082594, 0.411027, 0.39403] with:

Minimum: 0.39403 Maximum: 1.414687 Average: 0.7761 seconds

Number of cars 1
bbox:car_number ((813, 400), (885, 485)) : 1
The minimum distance from car: 1 is 7.280109889280518
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

95%|| 1198/1261 [53:35<02:49, 2.68s/it]

The times for each task are: [0.905209, 1.417025, 0.722206, 0.408382, 0.509152] with:

Minimum: 0.408382 Maximum: 1.417025 Average: 0.7924 seconds

Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

95%|| 1199/1261 [53:38<02:46, 2.68s/it]

The times for each task are: [1.01114, 0.608263, 1.620499, 0.456768, 0.440287] with:

Minimum: 0.440287 Maximum: 1.620499 Average: 0.8274 seconds

Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

```
Length of task list: 5
Number of processes used: 3
```

```
95%|| 1200/1261 [53:40<02:43, 2.68s/it]
```

```
The times for each task are: [0.908129, 1.429548, 0.677183, 0.305058, 0.393163] with:
```

```
Minimum: 0.305058 Maximum: 1.429548 Average: 0.7426 seconds
```

```
Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
95%|| 1201/1261 [53:43<02:41, 2.68s/it]
```

```
The times for each task are: [0.970613, 0.742777, 0.46225, 1.739192, 0.306415] with:
```

```
Minimum: 0.306415 Maximum: 1.739192 Average: 0.8442 seconds
```

```
Number of cars 1
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
95%|| 1202/1261 [53:46<02:38, 2.68s/it]
```

```
The times for each task are: [0.970099, 0.722737, 1.41431, 0.3076, 0.482122] with:
```

```
Minimum: 0.3076 Maximum: 1.41431 Average: 0.7794 seconds
```

```
Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
```

Number of processes used: 3

95%|| 1203/1261 [53:48<02:35, 2.68s/it]

The times for each task are: [0.63023, 0.871255, 1.401729, 0.408701, 0.345692] with:

Minimum: 0.345692 Maximum: 1.401729 Average: 0.7315 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

95%|| 1204/1261 [53:51<02:32, 2.68s/it]

The times for each task are: [0.644919, 1.443533, 0.926318, 0.5137, 0.394356] with:

Minimum: 0.394356 Maximum: 1.443533 Average: 0.7846 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 485)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1205/1261 [53:53<02:30, 2.68s/it]

The times for each task are: [0.91075, 0.650619, 1.685345, 0.342361, 0.435069] with:

Minimum: 0.342361 Maximum: 1.685345 Average: 0.8048 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 5.385164807134504

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1206/1261 [53:56<02:27, 2.68s/it]

The times for each task are: [1.524233, 1.089218, 0.657218, 0.354543, 0.485641] with:

Minimum: 0.354543 Maximum: 1.524233 Average: 0.8222 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1207/1261 [53:59<02:24, 2.68s/it]

The times for each task are: [0.629317, 0.988313, 1.575942, 0.344482, 0.499607] with:

Minimum: 0.344482 Maximum: 1.575942 Average: 0.8075 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 485)) : 1

The minimum distance from car: 1 is 5.385164807134504

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1208/1261 [54:01<02:22, 2.68s/it]

The times for each task are: [0.980603, 0.719463, 1.536723, 0.365273, 0.509192] with:

Minimum: 0.365273 Maximum: 1.536723 Average: 0.8223 seconds

Number of cars 1

bbox:car_number ((805, 400), (896, 495)) : 1

The minimum distance from car: 1 is 5.0990195135927845

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1209/1261 [54:04<02:19, 2.68s/it]

The times for each task are: [1.007264, 1.484729, 0.766492, 0.474889, 0.309965] with:

Minimum: 0.309965 Maximum: 1.484729 Average: 0.8087 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 485)) : 1

The minimum distance from car: 1 is 7.0710678118654755

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1210/1261 [54:07<02:16, 2.68s/it]

The times for each task are: [0.889087, 0.601856, 1.457558, 0.29319, 0.411448] with:

Minimum: 0.29319 Maximum: 1.457558 Average: 0.7306 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 5.385164807134504

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1211/1261 [54:09<02:14, 2.68s/it]

The times for each task are: [0.584985, 1.021451, 1.512327, 0.415972, 0.308518] with:

Minimum: 0.308518 Maximum: 1.512327 Average: 0.7687 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1212/1261 [54:12<02:11, 2.68s/it]

The times for each task are: [0.650159, 0.906862, 0.418616, 0.317398, 1.669358] with:

Minimum: 0.317398 Maximum: 1.669358 Average: 0.7925 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 485)) : 1

The minimum distance from car: 1 is 5.385164807134504

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1213/1261 [54:14<02:08, 2.68s/it]

The times for each task are: [0.698791, 0.953666, 1.512065, 0.329421, 0.456773] with:

Minimum: 0.329421 Maximum: 1.512065 Average: 0.7901 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 485)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1214/1261 [54:16<02:06, 2.68s/it]

The times for each task are: [0.926239, 0.584372, 0.446302, 1.44131, 0.315514] with:

Minimum: 0.315514 Maximum: 1.44131 Average: 0.7427 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 485)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1215/1261 [54:19<02:03, 2.68s/it]

The times for each task are: [0.636806, 1.001457, 1.414245, 0.418001, 0.342465] with:

Minimum: 0.342465 Maximum: 1.414245 Average: 0.7626 seconds

Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

96%|| 1216/1261 [54:21<02:00, 2.68s/it]

The times for each task are: [1.433436, 0.931922, 0.715134, 0.450884, 0.385159] with:

Minimum: 0.385159 Maximum: 1.433436 Average: 0.7833 seconds

Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

97%|| 1217/1261 [54:24<01:58, 2.68s/it]

The times for each task are: [0.625623, 0.91155, 1.393086, 0.305475, 0.408136] with:

Minimum: 0.305475 Maximum: 1.393086 Average: 0.7288 seconds

Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

97%|| 1218/1261 [54:27<01:55, 2.68s/it]

The times for each task are: [1.380217, 1.022621, 0.620603, 0.390119, 0.330583] with:

Minimum: 0.330583 Maximum: 1.380217 Average: 0.7488 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

97%|| 1219/1261 [54:29<01:52, 2.68s/it]

The times for each task are: [0.936577, 0.682281, 0.313576, 1.681525, 0.443989] with:
Minimum: 0.313576 Maximum: 1.681525 Average: 0.8116 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

97%|| 1220/1261 [54:32<01:49, 2.68s/it]

The times for each task are: [0.895958, 1.58751, 0.654867, 0.345327, 0.417682] with:
Minimum: 0.345327 Maximum: 1.58751 Average: 0.7803 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

97%|| 1221/1261 [54:34<01:47, 2.68s/it]

The times for each task are: [0.683014, 0.988401, 0.326912, 1.800445, 0.403911] with:
Minimum: 0.326912 Maximum: 1.800445 Average: 0.8405 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1222/1261 [54:37<01:44, 2.68s/it]

The times for each task are: [1.431898, 0.611057, 1.053829, 0.302509, 0.427126] with:
Minimum: 0.302509 Maximum: 1.431898 Average: 0.7653 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1223/1261 [54:40<01:41, 2.68s/it]

The times for each task are: [1.428722, 0.708105, 0.966419, 0.401926, 0.361599] with:
Minimum: 0.361599 Maximum: 1.428722 Average: 0.7734 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1224/1261 [54:42<01:39, 2.68s/it]

The times for each task are: [0.940666, 1.477664, 0.637287, 0.368045, 0.440397] with:
Minimum: 0.368045 Maximum: 1.477664 Average: 0.7728 seconds

Number of cars 1

```
bbox:car_number ((813, 419), (885, 485)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1225/1261 [54:45<01:36, 2.68s/it]

The times for each task are: [0.88529, 0.701297, 1.480081, 0.423382, 0.463091] with:

Minimum: 0.423382 Maximum: 1.480081 Average: 0.7906 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 10.770329614269007
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1226/1261 [54:48<01:33, 2.68s/it]

The times for each task are: [0.612004, 0.980856, 1.537514, 0.417784, 0.327036] with:

Minimum: 0.327036 Maximum: 1.537514 Average: 0.775 seconds

```
Number of cars 1
bbox:car_number ((799, 400), (885, 485)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1227/1261 [54:51<01:31, 2.68s/it]

The times for each task are: [1.396939, 0.735117, 0.992456, 0.458733, 0.303889] with:

Minimum: 0.303889 Maximum: 1.396939 Average: 0.7774 seconds

```
Number of cars 1
bbox:car_number ((799, 400), (885, 485)) : 1
```

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

97%|| 1228/1261 [54:53<01:28, 2.68s/it]

The times for each task are: [1.053508, 0.614218, 1.480665, 0.288132, 0.466363] with:

Minimum: 0.288132 Maximum: 1.480665 Average: 0.7806 seconds

Number of cars 1

bbox:car_number ((799, 400), (885, 485)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

97%|| 1229/1261 [54:56<01:25, 2.68s/it]

The times for each task are: [0.676499, 1.490271, 0.939069, 0.316861, 0.417615] with:

Minimum: 0.316861 Maximum: 1.490271 Average: 0.7681 seconds

Number of cars 1

bbox:car_number ((803, 400), (885, 478)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

98%|| 1230/1261 [54:59<01:23, 2.68s/it]

The times for each task are: [0.962063, 0.684438, 1.58717, 0.32418, 0.47821] with:

Minimum: 0.32418 Maximum: 1.58717 Average: 0.8072 seconds

Number of cars 1

bbox:car_number ((805, 400), (877, 477)) : 1

The minimum distance from car: 1 is 3.1622776601683795

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

98%|| 1231/1261 [55:02<01:20, 2.68s/it]

The times for each task are: [0.883701, 0.631466, 1.479142, 0.419621, 0.296053] with:

Minimum: 0.296053 Maximum: 1.479142 Average: 0.742 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (877, 478)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

98%|| 1232/1261 [55:04<01:17, 2.68s/it]

The times for each task are: [0.680781, 0.893707, 1.578797, 0.32849, 0.500639] with:

Minimum: 0.32849 Maximum: 1.578797 Average: 0.7965 seconds

```
Number of cars 1
bbox:car_number ((799, 400), (867, 478)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

98%|| 1233/1261 [55:07<01:15, 2.68s/it]

The times for each task are: [0.894605, 0.648027, 1.476096, 0.383843, 0.336182] with:

Minimum: 0.336182 Maximum: 1.476096 Average: 0.7478 seconds

```
Number of cars 1
bbox:car_number ((790, 400), (867, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
Length of task list: 5  
Number of processes used: 3
```

98%|| 1234/1261 [55:10<01:12, 2.68s/it]

The times for each task are: [1.420657, 0.722959, 0.993731, 0.363347, 0.408114] with:

Minimum: 0.363347 Maximum: 1.420657 Average: 0.7818 seconds

```
Number of cars 1  
bbox:car_number ((790, 400), (867, 485)) : 1  
The minimum distance from car: 1 is 3.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
Length of task list: 5  
Number of processes used: 3
```

98%|| 1235/1261 [55:12<01:09, 2.68s/it]

The times for each task are: [0.87239, 0.62273, 0.511309, 1.655493, 0.325797] with:

Minimum: 0.325797 Maximum: 1.655493 Average: 0.7975 seconds

```
Number of cars 1  
bbox:car_number ((790, 400), (867, 485)) : 1  
The minimum distance from car: 1 is 0.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
Length of task list: 5  
Number of processes used: 3
```

98%|| 1236/1261 [55:15<01:07, 2.68s/it]

The times for each task are: [0.867342, 1.377372, 0.616031, 0.413275, 0.360414] with:

Minimum: 0.360414 Maximum: 1.377372 Average: 0.7269 seconds

```
Number of cars 2  
bbox:car_number ((790, 400), (867, 485)) : 1  
The minimum distance from car: 1 is 0.0  
bbox:car_number ((580, 490), (597, 507)) : 2  
totalCars: 11
```

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

98%|| 1237/1261 [55:18<01:04, 2.68s/it]

The times for each task are: [0.583767, 1.046464, 1.438697, 0.392545, 0.292666] with:

Minimum: 0.292666 Maximum: 1.438697 Average: 0.7508 seconds

Number of cars 1
bbox:car_number ((790, 400), (862, 478)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

98%|| 1238/1261 [55:20<01:01, 2.68s/it]

The times for each task are: [1.37406, 0.647983, 0.885524, 0.445705, 0.353238] with:

Minimum: 0.353238 Maximum: 1.37406 Average: 0.7413 seconds

Number of cars 1
bbox:car_number ((790, 400), (867, 485)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

98%|| 1239/1261 [55:23<00:59, 2.68s/it]

The times for each task are: [1.06216, 0.637468, 1.477255, 0.326634, 0.459767] with:

Minimum: 0.326634 Maximum: 1.477255 Average: 0.7927 seconds

Number of cars 1
bbox:car_number ((790, 400), (863, 478)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

```
Length of task list: 5
Number of processes used: 3
```

```
98%|| 1240/1261 [55:26<00:56, 2.68s/it]
```

```
The times for each task are: [0.944954, 0.745227, 1.452179, 0.39503, 0.339668] with:
```

```
Minimum: 0.339668 Maximum: 1.452179 Average: 0.7754 seconds
```

```
Number of cars 1
bbox:car_number ((790, 400), (863, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
98%|| 1241/1261 [55:28<00:53, 2.68s/it]
```

```
The times for each task are: [0.609283, 0.970051, 1.398491, 0.428139, 0.391583] with:
```

```
Minimum: 0.391583 Maximum: 1.398491 Average: 0.7595 seconds
```

```
Number of cars 1
bbox:car_number ((790, 400), (862, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
98%|| 1242/1261 [55:31<00:50, 2.68s/it]
```

```
The times for each task are: [0.901587, 1.390878, 0.737773, 0.332515, 0.402904] with:
```

```
Minimum: 0.332515 Maximum: 1.390878 Average: 0.7531 seconds
```

```
Number of cars 1
bbox:car_number ((790, 400), (863, 485)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
```

Number of processes used: 3

99%|| 1243/1261 [55:33<00:48, 2.68s/it]

The times for each task are: [0.933422, 0.659288, 1.688544, 0.315242, 0.401092] with:

Minimum: 0.315242 Maximum: 1.688544 Average: 0.7995 seconds

Number of cars 1

bbox:car_number ((780, 400), (863, 478)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

99%|| 1244/1261 [55:36<00:45, 2.68s/it]

The times for each task are: [0.656792, 1.114186, 1.496053, 0.41857, 0.341847] with:

Minimum: 0.341847 Maximum: 1.496053 Average: 0.8055 seconds

Number of cars 1

bbox:car_number ((780, 400), (863, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

99%|| 1245/1261 [55:38<00:42, 2.68s/it]

The times for each task are: [0.904308, 1.463639, 0.635067, 0.397072, 0.498926] with:

Minimum: 0.397072 Maximum: 1.463639 Average: 0.7798 seconds

Number of cars 1

bbox:car_number ((780, 400), (862, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

99%|| 1246/1261 [55:41<00:40, 2.68s/it]

The times for each task are: [0.595713, 1.427252, 1.069884, 0.302208, 0.388048] with:

Minimum: 0.302208 Maximum: 1.427252 Average: 0.7566 seconds

Number of cars 1

bbox:car_number ((780, 400), (862, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

99%|| 1247/1261 [55:44<00:37, 2.68s/it]

The times for each task are: [0.654997, 0.886765, 0.413881, 1.692887, 0.293038] with:

Minimum: 0.293038 Maximum: 1.692887 Average: 0.7883 seconds

Number of cars 2

bbox:car_number ((780, 400), (862, 485)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((571, 475), (630, 522)) : 2

totalCars: 12

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 12

Car Number: 2 Car Positions: [((571, 475), (630, 522))]

Length of task list: 5

Number of processes used: 3

99%|| 1248/1261 [55:47<00:34, 2.68s/it]

The times for each task are: [0.589712, 0.933103, 1.705098, 0.422195, 0.309674] with:

Minimum: 0.309674 Maximum: 1.705098 Average: 0.792 seconds

Number of cars 1

bbox:car_number ((778, 400), (862, 485)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 12

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

```
Length of task list: 5
Number of processes used: 3
```

```
99%|| 1249/1261 [55:50<00:32, 2.68s/it]
```

```
The times for each task are: [0.893733, 0.702722, 0.389018, 1.615978, 0.341101] with:
```

```
Minimum: 0.341101 Maximum: 1.615978 Average: 0.7885 seconds
```

```
Number of cars 3
```

```
bbox:car_number ((780, 400), (862, 478)) : 1
```

```
The minimum distance from car: 1 is 3.1622776601683795
```

```
bbox:car_number ((1165, 445), (1197, 462)) : 2
```

```
bbox:car_number ((1204, 445), (1212, 462)) : 3
```

```
totalCars: 12
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
99%|| 1250/1261 [55:52<00:29, 2.68s/it]
```

```
The times for each task are: [1.357647, 0.692462, 0.909843, 0.399947, 0.35929] with:
```

```
Minimum: 0.35929 Maximum: 1.357647 Average: 0.7438 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((778, 400), (862, 478)) : 1
```

```
The minimum distance from car: 1 is 1.0
```

```
totalCars: 12
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
99%|| 1251/1261 [55:55<00:26, 2.68s/it]
```

```
The times for each task are: [0.583393, 0.93878, 1.384495, 0.38325, 0.386397] with:
```

```
Minimum: 0.38325 Maximum: 1.384495 Average: 0.7353 seconds
```

```
Number of cars 2
```

```
bbox:car_number ((799, 400), (858, 472)) : 1
```

```
The minimum distance from car: 1 is 8.54400374531753
```

```
bbox:car_number ((1217, 430), (1272, 477)) : 2
```

```
The minimum distance from car: 3 is 15.297058540778355
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 12
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

99%|| 1252/1261 [55:58<00:24, 2.68s/it]

The times for each task are: [0.619332, 0.92076, 0.313575, 1.533795, 0.406995] with:

Minimum: 0.313575 Maximum: 1.533795 Average: 0.7589 seconds

```
Number of cars 1
bbox:car_number ((778, 415), (852, 478)) : 1
The minimum distance from car: 1 is 16.401219466856727
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

99%|| 1253/1261 [56:00<00:21, 2.68s/it]

The times for each task are: [0.885562, 1.497902, 0.617607, 0.416324, 0.311076] with:

Minimum: 0.311076 Maximum: 1.497902 Average: 0.7457 seconds

```
Number of cars 2
bbox:car_number ((775, 415), (852, 477)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1183, 419), (1273, 497)) : 2
The minimum distance from car: 3 is 16.76305461424021
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 12
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

99%|| 1254/1261 [56:03<00:18, 2.68s/it]

The times for each task are: [0.92601, 0.598974, 0.398086, 1.431261, 0.438991] with:

Minimum: 0.398086 Maximum: 1.431261 Average: 0.7587 seconds

Number of cars 2
bbox:car_number ((775, 415), (852, 478)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1181, 423), (1268, 498)) : 2
The minimum distance from car: 3 is 4.47213595499958
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 12
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

100%|| 1255/1261 [56:06<00:16, 2.68s/it]

The times for each task are: [0.938498, 0.618259, 1.572112, 0.369521, 0.452663] with:

Minimum: 0.369521 Maximum: 1.572112 Average: 0.7902 seconds

Number of cars 2
bbox:car_number ((775, 400), (852, 477)) : 1
The minimum distance from car: 1 is 8.0
bbox:car_number ((1180, 419), (1253, 492)) : 2
The minimum distance from car: 3 is 9.433981132056603
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 12
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

100%|| 1256/1261 [56:09<00:13, 2.68s/it]

The times for each task are: [0.884573, 0.667413, 1.603738, 0.302086, 0.390204] with:

Minimum: 0.302086 Maximum: 1.603738 Average: 0.7696 seconds

Number of cars 2
bbox:car_number ((1158, 400), (1238, 495)) : 1
The minimum distance from car: 3 is 9.219544457292887
bbox:car_number ((772, 415), (852, 478)) : 2
The minimum distance from car: 1 is 8.06225774829855
totalCars: 12
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

```
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

100%|| 1257/1261 [56:11<00:10, 2.68s/it]

The times for each task are: [0.673739, 1.396026, 0.915638, 0.294361, 0.452464] with:

Minimum: 0.294361 Maximum: 1.396026 Average: 0.7464 seconds

```
Number of cars 2
bbox:car_number ((772, 400), (840, 478)) : 1
The minimum distance from car: 1 is 9.219544457292887
bbox:car_number ((1158, 419), (1230, 485)) : 2
The minimum distance from car: 3 is 6.4031242374328485
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 12
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

100%|| 1258/1261 [56:14<00:08, 2.68s/it]

The times for each task are: [0.628723, 0.92029, 1.394499, 0.328446, 0.420065] with:

Minimum: 0.328446 Maximum: 1.394499 Average: 0.7384 seconds

```
Number of cars 2
bbox:car_number ((780, 400), (840, 477)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((1158, 415), (1241, 495)) : 2
The minimum distance from car: 3 is 5.830951894845301
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 12
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

100%|| 1259/1261 [56:17<00:05, 2.68s/it]

```
The times for each task are: [0.625054, 0.856797, 1.645487, 0.44303, 0.36733] with:
```

```
Minimum: 0.36733 Maximum: 1.645487 Average: 0.7875 seconds
```

```
Number of cars 2
```

```
bbox:car_number ((780, 400), (839, 472)) : 1
```

```
The minimum distance from car: 1 is 2.23606797749979
```

```
bbox:car_number ((1156, 400), (1230, 485)) : 2
```

```
The minimum distance from car: 3 is 14.317821063276353
```

```
totalCars: 12
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 12
```

```
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
100%|| 1260/1261 [56:20<00:02, 2.68s/it]
```

```
The times for each task are: [1.068214, 0.715227, 1.405648, 0.339424, 0.462745] with:
```

```
Minimum: 0.339424 Maximum: 1.405648 Average: 0.7983 seconds
```

```
Number of cars 3
```

```
bbox:car_number ((778, 400), (839, 478)) : 1
```

```
The minimum distance from car: 1 is 3.1622776601683795
```

```
bbox:car_number ((1136, 400), (1208, 485)) : 2
```

```
The minimum distance from car: 3 is 21.0
```

```
bbox:car_number ((1210, 415), (1230, 472)) : 3
```

```
totalCars: 12
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 12
```

```
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
```

```
[MoviePy] Done.
```

```
[MoviePy] >>> Video ready: project_videos_output/project_video_after_process_without_vicitniy9.mp4
```

```
CPU times: user 1min 38s, sys: 21min 42s, total: 23min 21s
```

```
Wall time: 56min 21s
```

```
In [165]: Car.totalCars
```

```
Out[165]: 12
```

```
In [167]: vicinity_check = 1
vicinity_print = 1

Car.totalCars = 0
allCars= []

project_video_output = 'project_videos_output/project_video_after_process_with_vicinity'
## To speed up the testing process you may want to try your pipeline on a shorter subclip
## To do so add .subclip(start_second,end_second) to the end of the line below
## Where start_second and end_second are integer values representing the start and end
## You may also uncomment the following line for a subclip of the first 5 seconds
##clip1 = VideoFileClip( "test_videos/solidWhiteRight.mp4").subclip(0,5)
clip1 = VideoFileClip("project_video/project_video_with_lane.mp4")
white_clip = clip1.fl_image(process_image) #NOTE: this function expects color images!!!
%time white_clip.write_videofile(project_video_output, audio=False)
```

The times for each task are: [0.879234, 0.708544, 1.445117, 0.399026, 0.394081] with:

Minimum: 0.394081 Maximum: 1.445117 Average: 0.7652 seconds

Number of cars 0

[MoviePy] >>> Building video project_videos_output/project_video_after_process_with_vicitniy9.m

[MoviePy] Writing video project_videos_output/project_video_after_process_with_vicitniy9.mp4

0% | 0/1261 [00:00<?, ?it/s]

Length of task list: 5

Number of processes used: 3

0% | 1/1261 [00:02<55:56, 2.66s/it]

The times for each task are: [0.903263, 1.402521, 0.701435, 0.294611, 0.432821] with:

Minimum: 0.294611 Maximum: 1.402521 Average: 0.7469 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

0% | 2/1261 [00:05<56:35, 2.70s/it]

The times for each task are: [0.905973, 0.618372, 0.488439, 0.346847, 1.689604] with:

Minimum: 0.346847 Maximum: 1.689604 Average: 0.8098 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

0% | 3/1261 [00:07<54:24, 2.60s/it]

The times for each task are: [0.623475, 1.463289, 0.918226, 0.361288, 0.42798] with:

Minimum: 0.361288 Maximum: 1.463289 Average: 0.7589 seconds

Number of cars 1

bbox:car_number ((565, 460), (611, 507)) : 1

Length of task list: 5
Number of processes used: 3

0% | 4/1261 [00:10<53:32, 2.56s/it]

The times for each task are: [0.905049, 0.689321, 1.351968, 0.447955, 0.283476] with:

Minimum: 0.283476 Maximum: 1.351968 Average: 0.7356 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

0% | 5/1261 [00:13<54:48, 2.62s/it]

The times for each task are: [1.464543, 0.714009, 0.905262, 0.400366, 0.304067] with:

Minimum: 0.304067 Maximum: 1.464543 Average: 0.7576 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

0% | 6/1261 [00:15<54:45, 2.62s/it]

The times for each task are: [0.89619, 0.679847, 1.421247, 0.334903, 0.385744] with:

Minimum: 0.334903 Maximum: 1.421247 Average: 0.7436 seconds

Number of cars 1
bbox:car_number ((565, 475), (597, 492)) : 1
Length of task list: 5
Number of processes used: 3

1% | 7/1261 [00:18<54:59, 2.63s/it]

The times for each task are: [0.62917, 0.933603, 1.4545, 0.294677, 0.394882] with:

Minimum: 0.294677 Maximum: 1.4545 Average: 0.7414 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

1% | 8/1261 [00:21<55:19, 2.65s/it]

The times for each task are: [0.920448, 0.62469, 1.684578, 0.390456, 0.293599] with:

Minimum: 0.293599 Maximum: 1.684578 Average: 0.7828 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

1% | 9/1261 [00:24<55:43, 2.67s/it]

The times for each task are: [0.875105, 1.641237, 0.629201, 0.465464, 0.356561] with:

Minimum: 0.356561 Maximum: 1.641237 Average: 0.7935 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

1% | 10/1261 [00:26<55:15, 2.65s/it]

The times for each task are: [0.632168, 1.44381, 0.980293, 0.300447, 0.402762] with:

Minimum: 0.300447 Maximum: 1.44381 Average: 0.7519 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

1% | 11/1261 [00:28<54:22, 2.61s/it]

The times for each task are: [0.874513, 0.584762, 0.307181, 1.71677, 0.430946] with:

Minimum: 0.307181 Maximum: 1.71677 Average: 0.7828 seconds

Number of cars 1
bbox:car_number ((565, 460), (597, 492)) : 1

Length of task list: 5
Number of processes used: 3

1% | 12/1261 [00:31<54:33, 2.62s/it]

The times for each task are: [0.902804, 0.421636, 0.733636, 1.419239, 0.292301] with:

Minimum: 0.292301 Maximum: 1.419239 Average: 0.7539 seconds

Number of cars 1
bbox:car_number ((552, 475), (582, 507)) : 1
Length of task list: 5
Number of processes used: 3

1% | 13/1261 [00:34<54:26, 2.62s/it]

The times for each task are: [0.722213, 0.931366, 1.441201, 0.319695, 0.469578] with:

Minimum: 0.319695 Maximum: 1.441201 Average: 0.7768 seconds

Number of cars 1
bbox:car_number ((552, 460), (597, 507)) : 1
Length of task list: 5
Number of processes used: 3

1% | 14/1261 [00:36<54:41, 2.63s/it]

The times for each task are: [0.648361, 1.559182, 1.00524, 0.504881, 0.429445] with:

Minimum: 0.429445 Maximum: 1.559182 Average: 0.8294 seconds

Number of cars 1
bbox:car_number ((550, 475), (582, 516)) : 1
Length of task list: 5
Number of processes used: 3

1% | 15/1261 [00:39<54:57, 2.65s/it]

The times for each task are: [1.44701, 1.1327, 0.629858, 0.408779, 0.437166] with:

Minimum: 0.408779 Maximum: 1.44701 Average: 0.8111 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

1%| 16/1261 [00:42<54:48, 2.64s/it]

The times for each task are: [0.99931, 0.628699, 1.43029, 0.46039, 0.427739] with:
Minimum: 0.427739 Maximum: 1.43029 Average: 0.7893 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

1%| 17/1261 [00:45<55:15, 2.66s/it]

The times for each task are: [1.398513, 0.62664, 1.039838, 0.302753, 0.47023] with:
Minimum: 0.302753 Maximum: 1.398513 Average: 0.7676 seconds

Number of cars 1
bbox:car_number ((552, 505), (552, 516)) : 1
Length of task list: 5
Number of processes used: 3

1%| 18/1261 [00:47<55:09, 2.66s/it]

The times for each task are: [0.627609, 1.374752, 1.025289, 0.35968, 0.388043] with:
Minimum: 0.35968 Maximum: 1.374752 Average: 0.7551 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 19/1261 [00:50<54:48, 2.65s/it]

The times for each task are: [0.880488, 0.624355, 0.334329, 1.674287, 0.41582] with:
Minimum: 0.334329 Maximum: 1.674287 Average: 0.7859 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 20/1261 [00:53<54:55, 2.66s/it]

The times for each task are: [0.913887, 1.484651, 0.746464, 0.411998, 0.31342] with:
Minimum: 0.31342 Maximum: 1.484651 Average: 0.7741 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 21/1261 [00:55<55:02, 2.66s/it]

The times for each task are: [0.641205, 0.918617, 1.666839, 0.396386, 0.355239] with:
Minimum: 0.355239 Maximum: 1.666839 Average: 0.7957 seconds

Number of cars 1
bbox:car_number ((550, 460), (597, 518)) : 1
Length of task list: 5
Number of processes used: 3

2%| 22/1261 [00:58<54:59, 2.66s/it]

The times for each task are: [0.601981, 0.932121, 1.640219, 0.440336, 0.316895] with:
Minimum: 0.316895 Maximum: 1.640219 Average: 0.7863 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 23/1261 [01:01<55:08, 2.67s/it]

The times for each task are: [0.859551, 0.650124, 0.456547, 1.415186, 0.325654] with:
Minimum: 0.325654 Maximum: 1.415186 Average: 0.7414 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 24/1261 [01:04<55:15, 2.68s/it]

The times for each task are: [0.653037, 0.876232, 1.460014, 0.406205, 0.572475] with:

Minimum: 0.406205 Maximum: 1.460014 Average: 0.7936 seconds

Number of cars 1
bbox:car_number ((1075, 415), (1086, 447)) : 1
Length of task list: 5
Number of processes used: 3

2%| 25/1261 [01:06<55:02, 2.67s/it]

The times for each task are: [1.460741, 0.913543, 0.770617, 0.415852, 0.359436] with:

Minimum: 0.359436 Maximum: 1.460741 Average: 0.784 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 26/1261 [01:09<55:18, 2.69s/it]

The times for each task are: [0.636811, 1.009369, 1.50048, 0.435701, 0.359194] with:

Minimum: 0.359194 Maximum: 1.50048 Average: 0.7883 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 27/1261 [01:12<55:16, 2.69s/it]

The times for each task are: [0.943242, 1.402749, 0.65512, 0.349399, 0.510245] with:

Minimum: 0.349399 Maximum: 1.402749 Average: 0.7722 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 28/1261 [01:15<55:24, 2.70s/it]

The times for each task are: [1.467731, 0.657537, 1.026071, 0.435872, 0.395414] with:
Minimum: 0.395414 Maximum: 1.467731 Average: 0.7965 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 29/1261 [01:18<55:28, 2.70s/it]

The times for each task are: [0.661944, 0.923747, 1.458422, 0.331329, 0.471069] with:
Minimum: 0.331329 Maximum: 1.458422 Average: 0.7693 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

2%| 30/1261 [01:21<55:31, 2.71s/it]

The times for each task are: [0.638664, 1.456791, 0.936972, 0.387503, 0.413537] with:
Minimum: 0.387503 Maximum: 1.456791 Average: 0.7667 seconds

Number of cars 1
bbox:car_number ((1045, 415), (1077, 447)) : 1
Length of task list: 5
Number of processes used: 3

2%| 31/1261 [01:23<55:20, 2.70s/it]

The times for each task are: [0.620265, 0.932151, 1.656351, 0.394109, 0.304143] with:
Minimum: 0.304143 Maximum: 1.656351 Average: 0.7814 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

3%| 32/1261 [01:26<55:29, 2.71s/it]

The times for each task are: [0.934933, 0.643443, 1.519002, 0.533282, 0.327533] with:
Minimum: 0.327533 Maximum: 1.519002 Average: 0.7916 seconds

Number of cars 2
bbox:car_number ((1120, 400), (1152, 447)) : 1
bbox:car_number ((1105, 415), (1105, 447)) : 2
Length of task list: 5
Number of processes used: 3

3%| 33/1261 [01:29<55:20, 2.70s/it]

The times for each task are: [0.879108, 0.633443, 1.482315, 0.421442, 0.313083] with:
Minimum: 0.313083 Maximum: 1.482315 Average: 0.7459 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

3%| 34/1261 [01:32<55:31, 2.72s/it]

The times for each task are: [1.471835, 0.687886, 1.13219, 0.401044, 0.411595] with:
Minimum: 0.401044 Maximum: 1.471835 Average: 0.8209 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

3%| 35/1261 [01:34<55:26, 2.71s/it]

The times for each task are: [0.961249, 1.420824, 0.747156, 0.435696, 0.388697] with:

Minimum: 0.388697 Maximum: 1.420824 Average: 0.7907 seconds

Number of cars 1
bbox:car_number ((552, 475), (597, 492)) : 1
Length of task list: 5
Number of processes used: 3

3% | 36/1261 [01:37<55:19, 2.71s/it]

The times for each task are: [1.014642, 1.539355, 0.750897, 0.472971, 0.331238] with:

Minimum: 0.331238 Maximum: 1.539355 Average: 0.8218 seconds

Number of cars 1
bbox:car_number ((552, 490), (567, 492)) : 1
Length of task list: 5
Number of processes used: 3

3% | 37/1261 [01:40<55:21, 2.71s/it]

The times for each task are: [0.929019, 0.706427, 1.459618, 0.468414, 0.407946] with:

Minimum: 0.407946 Maximum: 1.459618 Average: 0.7943 seconds

Number of cars 1
bbox:car_number ((1135, 415), (1152, 447)) : 1
Length of task list: 5
Number of processes used: 3

3% | 38/1261 [01:42<55:07, 2.70s/it]

The times for each task are: [0.64862, 0.945958, 1.470792, 0.401209, 0.421009] with:

Minimum: 0.401209 Maximum: 1.470792 Average: 0.7775 seconds

Number of cars 3
bbox:car_number ((1150, 400), (1167, 447)) : 1
bbox:car_number ((1179, 400), (1182, 447)) : 2
bbox:car_number ((1120, 415), (1122, 447)) : 3
Length of task list: 5
Number of processes used: 3

3%| | 39/1261 [01:45<55:00, 2.70s/it]

The times for each task are: [0.643013, 1.51988, 1.11577, 0.530988, 0.414072] with:

Minimum: 0.414072 Maximum: 1.51988 Average: 0.8447 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

3%| | 40/1261 [01:47<54:55, 2.70s/it]

The times for each task are: [0.928751, 1.465775, 0.591787, 0.357818, 0.442101] with:

Minimum: 0.357818 Maximum: 1.465775 Average: 0.7572 seconds

Number of cars 1

bbox:car_number ((552, 505), (567, 507)) : 1

Length of task list: 5

Number of processes used: 3

3%| | 41/1261 [01:50<54:50, 2.70s/it]

The times for each task are: [0.606039, 0.951297, 1.430537, 0.426788, 0.313279] with:

Minimum: 0.313279 Maximum: 1.430537 Average: 0.7456 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

3%| | 42/1261 [01:53<54:39, 2.69s/it]

The times for each task are: [0.884318, 1.420005, 0.352804, 0.644087, 0.489781] with:

Minimum: 0.352804 Maximum: 1.420005 Average: 0.7582 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

3%| | 43/1261 [01:55<54:37, 2.69s/it]

The times for each task are: [1.399393, 0.881409, 0.758622, 0.402006, 0.31884] with:

Minimum: 0.31884 Maximum: 1.399393 Average: 0.7521 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

3% | 44/1261 [01:58<54:29, 2.69s/it]

The times for each task are: [0.614537, 1.443351, 0.92126, 0.409523, 0.454111] with:

Minimum: 0.409523 Maximum: 1.443351 Average: 0.7686 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 45/1261 [02:00<54:23, 2.68s/it]

The times for each task are: [0.599711, 1.4589, 1.067956, 0.40088, 0.35467] with:

Minimum: 0.35467 Maximum: 1.4589 Average: 0.7764 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 46/1261 [02:03<54:15, 2.68s/it]

The times for each task are: [0.909066, 0.628518, 0.470106, 0.394631, 1.637882] with:

Minimum: 0.394631 Maximum: 1.637882 Average: 0.808 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4% | 47/1261 [02:05<54:09, 2.68s/it]

The times for each task are: [0.955917, 1.468227, 0.650094, 0.421017, 0.30036] with:

Minimum: 0.30036 Maximum: 1.468227 Average: 0.7591 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4%| 48/1261 [02:08<54:06, 2.68s/it]

The times for each task are: [0.651818, 0.913036, 1.451425, 0.391369, 0.285274] with:

Minimum: 0.285274 Maximum: 1.451425 Average: 0.7386 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4%| 49/1261 [02:11<54:14, 2.69s/it]

The times for each task are: [0.607564, 0.929985, 0.301305, 0.48569, 1.604305] with:

Minimum: 0.301305 Maximum: 1.604305 Average: 0.7858 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4%| 50/1261 [02:14<54:15, 2.69s/it]

The times for each task are: [0.895561, 0.631035, 1.460687, 0.414525, 0.380545] with:

Minimum: 0.380545 Maximum: 1.460687 Average: 0.7565 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

4%| 51/1261 [02:17<54:14, 2.69s/it]

The times for each task are: [0.93964, 0.656347, 1.508559, 0.408921, 0.30915] with:

Minimum: 0.30915 Maximum: 1.508559 Average: 0.7645 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4%| 52/1261 [02:19<54:03, 2.68s/it]

The times for each task are: [0.865245, 1.413344, 0.654835, 0.292281, 0.399724] with:
Minimum: 0.292281 Maximum: 1.413344 Average: 0.7251 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4%| 53/1261 [02:21<53:55, 2.68s/it]

The times for each task are: [0.619716, 1.089025, 1.400508, 0.349494, 0.438899] with:
Minimum: 0.349494 Maximum: 1.400508 Average: 0.7795 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4%| 54/1261 [02:24<53:51, 2.68s/it]

The times for each task are: [0.678482, 0.897357, 1.470786, 0.410786, 0.29951] with:
Minimum: 0.29951 Maximum: 1.470786 Average: 0.7514 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4%| 55/1261 [02:27<53:45, 2.67s/it]

The times for each task are: [0.643795, 0.893324, 1.459083, 0.418487, 0.297463] with:
Minimum: 0.297463 Maximum: 1.459083 Average: 0.7424 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

4% | 56/1261 [02:29<53:41, 2.67s/it]

The times for each task are: [0.665254, 0.874717, 1.44041, 0.359285, 0.43988] with:

Minimum: 0.359285 Maximum: 1.44041 Average: 0.7559 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 57/1261 [02:32<53:31, 2.67s/it]

The times for each task are: [0.893695, 0.634381, 0.403743, 1.42382, 0.393701] with:

Minimum: 0.393701 Maximum: 1.42382 Average: 0.7499 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 58/1261 [02:34<53:32, 2.67s/it]

The times for each task are: [1.003905, 1.521054, 0.661367, 0.351318, 0.390039] with:

Minimum: 0.351318 Maximum: 1.521054 Average: 0.7855 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 59/1261 [02:37<53:31, 2.67s/it]

The times for each task are: [0.887746, 0.627524, 0.402935, 1.688675, 0.2852] with:

Minimum: 0.2852 Maximum: 1.688675 Average: 0.7784 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

5% | 60/1261 [02:40<53:25, 2.67s/it]

The times for each task are: [0.943844, 0.664823, 1.418025, 0.398238, 0.282281] with:

Minimum: 0.282281 Maximum: 1.418025 Average: 0.7414 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 61/1261 [02:42<53:16, 2.66s/it]

The times for each task are: [0.590631, 0.907858, 1.43751, 0.411424, 0.304893] with:

Minimum: 0.304893 Maximum: 1.43751 Average: 0.7305 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 62/1261 [02:45<53:13, 2.66s/it]

The times for each task are: [0.64247, 0.891735, 1.490037, 0.355221, 0.405344] with:

Minimum: 0.355221 Maximum: 1.490037 Average: 0.757 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

5% | 63/1261 [02:47<53:14, 2.67s/it]

The times for each task are: [0.645957, 0.927948, 1.659077, 0.304869, 0.386699] with:

Minimum: 0.304869 Maximum: 1.659077 Average: 0.7849 seconds

Number of cars 0
Length of task list: 5

Number of processes used: 3

5%| | 64/1261 [02:50<53:14, 2.67s/it]

The times for each task are: [0.911054, 0.677015, 1.683708, 0.49443, 0.345971] with:

Minimum: 0.345971 Maximum: 1.683708 Average: 0.8224 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 65/1261 [02:53<53:11, 2.67s/it]

The times for each task are: [0.907713, 0.592479, 1.453677, 0.482134, 0.318421] with:

Minimum: 0.318421 Maximum: 1.453677 Average: 0.7509 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 66/1261 [02:56<53:10, 2.67s/it]

The times for each task are: [1.391179, 1.026364, 0.747719, 0.348516, 0.403806] with:

Minimum: 0.348516 Maximum: 1.391179 Average: 0.7835 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 67/1261 [02:58<53:03, 2.67s/it]

The times for each task are: [1.01632, 1.496049, 0.651943, 0.452433, 0.404952] with:

Minimum: 0.404952 Maximum: 1.496049 Average: 0.8043 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 68/1261 [03:01<53:03, 2.67s/it]

The times for each task are: [0.927712, 0.734733, 0.418206, 1.435786, 0.315898] with:

Minimum: 0.315898 Maximum: 1.435786 Average: 0.7665 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

5%| | 69/1261 [03:04<53:00, 2.67s/it]

The times for each task are: [0.580261, 1.399647, 1.05497, 0.427364, 0.37454] with:

Minimum: 0.37454 Maximum: 1.399647 Average: 0.7674 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6%| | 70/1261 [03:06<52:53, 2.66s/it]

The times for each task are: [0.897593, 0.688972, 1.659626, 0.38731, 0.288876] with:

Minimum: 0.288876 Maximum: 1.659626 Average: 0.7845 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6%| | 71/1261 [03:09<52:51, 2.67s/it]

The times for each task are: [0.858391, 0.742839, 1.43747, 0.457345, 0.351213] with:

Minimum: 0.351213 Maximum: 1.43747 Average: 0.7695 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6%| | 72/1261 [03:11<52:49, 2.67s/it]

The times for each task are: [0.589717, 1.394252, 0.867787, 0.411349, 0.312837] with:

Minimum: 0.312837 Maximum: 1.394252 Average: 0.7152 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 73/1261 [03:14<52:45, 2.66s/it]

The times for each task are: [0.586532, 1.375115, 0.921736, 0.384082, 0.283167] with:

Minimum: 0.283167 Maximum: 1.375115 Average: 0.7101 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 74/1261 [03:17<52:46, 2.67s/it]

The times for each task are: [1.002667, 0.694582, 1.633112, 0.463146, 0.387895] with:

Minimum: 0.387895 Maximum: 1.633112 Average: 0.8363 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 75/1261 [03:20<52:43, 2.67s/it]

The times for each task are: [1.445899, 0.974329, 0.774587, 0.43832, 0.322145] with:

Minimum: 0.322145 Maximum: 1.445899 Average: 0.7911 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 76/1261 [03:22<52:38, 2.67s/it]

The times for each task are: [0.96927, 0.637381, 1.474657, 0.314606, 0.422479] with:

Minimum: 0.314606 Maximum: 1.474657 Average: 0.7637 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 77/1261 [03:25<52:38, 2.67s/it]

The times for each task are: [1.523316, 0.650851, 1.089663, 0.417748, 0.376196] with:

Minimum: 0.376196 Maximum: 1.523316 Average: 0.8116 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 78/1261 [03:28<52:37, 2.67s/it]

The times for each task are: [0.595194, 1.06711, 1.428206, 0.417304, 0.357816] with:

Minimum: 0.357816 Maximum: 1.428206 Average: 0.7731 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 79/1261 [03:30<52:31, 2.67s/it]

The times for each task are: [0.684488, 1.403737, 0.872731, 0.425468, 0.381692] with:

Minimum: 0.381692 Maximum: 1.403737 Average: 0.7536 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

6% | 80/1261 [03:33<52:35, 2.67s/it]

The times for each task are: [0.705776, 0.955593, 1.463586, 0.318605, 0.527191] with:

Minimum: 0.318605 Maximum: 1.463586 Average: 0.7942 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

6%| 81/1261 [03:36<52:32, 2.67s/it]

The times for each task are: [0.700281, 1.001411, 1.53686, 0.440604, 0.332157] with:
Minimum: 0.332157 Maximum: 1.53686 Average: 0.8023 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 82/1261 [03:38<52:28, 2.67s/it]

The times for each task are: [0.932063, 1.498811, 0.754904, 0.304898, 0.493995] with:
Minimum: 0.304898 Maximum: 1.498811 Average: 0.7969 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 83/1261 [03:41<52:28, 2.67s/it]

The times for each task are: [0.959682, 0.738477, 1.70926, 0.35654, 0.42364] with:
Minimum: 0.35654 Maximum: 1.70926 Average: 0.8375 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 84/1261 [03:44<52:26, 2.67s/it]

The times for each task are: [1.372835, 0.93632, 0.650778, 0.346813, 0.38757] with:
Minimum: 0.346813 Maximum: 1.372835 Average: 0.7389 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 85/1261 [03:47<52:26, 2.68s/it]

The times for each task are: [0.864717, 0.638115, 1.702362, 0.425763, 0.343013] with:
Minimum: 0.343013 Maximum: 1.702362 Average: 0.7948 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 86/1261 [03:50<52:23, 2.67s/it]

The times for each task are: [0.941626, 1.449756, 0.636824, 0.316462, 0.452352] with:
Minimum: 0.316462 Maximum: 1.449756 Average: 0.7594 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 87/1261 [03:52<52:17, 2.67s/it]

The times for each task are: [0.892196, 0.622513, 1.4643, 0.443252, 0.383577] with:
Minimum: 0.383577 Maximum: 1.4643 Average: 0.7612 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7%| 88/1261 [03:55<52:19, 2.68s/it]

The times for each task are: [0.738047, 0.918133, 0.396966, 0.436943, 1.510895] with:
Minimum: 0.396966 Maximum: 1.510895 Average: 0.8002 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

7% | 89/1261 [03:58<52:18, 2.68s/it]

The times for each task are: [0.86605, 0.637119, 1.652566, 0.40514, 0.344803] with:

Minimum: 0.344803 Maximum: 1.652566 Average: 0.7811 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7% | 90/1261 [04:00<52:13, 2.68s/it]

The times for each task are: [0.878614, 0.623802, 1.413823, 0.426822, 0.301574] with:

Minimum: 0.301574 Maximum: 1.413823 Average: 0.7289 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7% | 91/1261 [04:03<52:09, 2.67s/it]

The times for each task are: [0.640447, 0.934166, 0.319047, 0.444097, 1.829296] with:

Minimum: 0.319047 Maximum: 1.829296 Average: 0.8334 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

7% | 92/1261 [04:05<52:05, 2.67s/it]

The times for each task are: [0.934849, 0.625181, 0.444906, 1.786658, 0.427027] with:

Minimum: 0.427027 Maximum: 1.786658 Average: 0.8437 seconds

Number of cars 0
Length of task list: 5

Number of processes used: 3

7%| | 93/1261 [04:08<52:01, 2.67s/it]

The times for each task are: [0.90573, 0.735671, 1.371717, 0.411959, 0.383066] with:

Minimum: 0.383066 Maximum: 1.371717 Average: 0.7616 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

7%| | 94/1261 [04:11<52:00, 2.67s/it]

The times for each task are: [0.921907, 0.626309, 1.424662, 0.408996, 0.350116] with:

Minimum: 0.350116 Maximum: 1.424662 Average: 0.7464 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 95/1261 [04:14<51:57, 2.67s/it]

The times for each task are: [1.409965, 0.625554, 0.86899, 0.386431, 0.343574] with:

Minimum: 0.343574 Maximum: 1.409965 Average: 0.7269 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 96/1261 [04:16<51:56, 2.68s/it]

The times for each task are: [1.430701, 0.858905, 0.62768, 0.436263, 0.357505] with:

Minimum: 0.357505 Maximum: 1.430701 Average: 0.7422 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 97/1261 [04:19<51:56, 2.68s/it]

The times for each task are: [0.630246, 0.954795, 1.472972, 0.321485, 0.533483] with:

Minimum: 0.321485 Maximum: 1.472972 Average: 0.7826 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 98/1261 [04:22<51:49, 2.67s/it]

The times for each task are: [0.939566, 0.600282, 1.478404, 0.438702, 0.39091] with:

Minimum: 0.39091 Maximum: 1.478404 Average: 0.7696 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 99/1261 [04:24<51:43, 2.67s/it]

The times for each task are: [0.974727, 1.387427, 0.664549, 0.333061, 0.474579] with:

Minimum: 0.333061 Maximum: 1.387427 Average: 0.7669 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 100/1261 [04:27<51:41, 2.67s/it]

The times for each task are: [0.688797, 0.882754, 1.643749, 0.39409, 0.286856] with:

Minimum: 0.286856 Maximum: 1.643749 Average: 0.7792 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8%| | 101/1261 [04:29<51:40, 2.67s/it]

The times for each task are: [0.928755, 0.656966, 1.612993, 0.411275, 0.351223] with:

Minimum: 0.351223 Maximum: 1.612993 Average: 0.7922 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 102/1261 [04:32<51:39, 2.67s/it]

The times for each task are: [0.633881, 0.877911, 1.640664, 0.311988, 0.475464] with:

Minimum: 0.311988 Maximum: 1.640664 Average: 0.788 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 103/1261 [04:35<51:36, 2.67s/it]

The times for each task are: [1.399842, 0.935134, 0.636947, 0.307596, 0.453145] with:

Minimum: 0.307596 Maximum: 1.399842 Average: 0.7465 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 104/1261 [04:38<51:33, 2.67s/it]

The times for each task are: [0.588041, 0.92383, 1.404948, 0.352114, 0.521038] with:

Minimum: 0.352114 Maximum: 1.404948 Average: 0.758 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 105/1261 [04:40<51:27, 2.67s/it]

The times for each task are: [0.920559, 0.588563, 1.446749, 0.416049, 0.287124] with:

Minimum: 0.287124 Maximum: 1.446749 Average: 0.7318 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 106/1261 [04:43<51:25, 2.67s/it]

The times for each task are: [0.894789, 0.623827, 1.435137, 0.409353, 0.315039] with:

Minimum: 0.315039 Maximum: 1.435137 Average: 0.7356 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

8% | 107/1261 [04:45<51:23, 2.67s/it]

The times for each task are: [0.640068, 0.938023, 1.452679, 0.398818, 0.352286] with:

Minimum: 0.352286 Maximum: 1.452679 Average: 0.7564 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

9% | 108/1261 [04:48<51:21, 2.67s/it]

The times for each task are: [0.867373, 0.699518, 1.655913, 0.350817, 0.452865] with:

Minimum: 0.350817 Maximum: 1.655913 Average: 0.8053 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

9% | 109/1261 [04:51<51:17, 2.67s/it]

The times for each task are: [0.662771, 0.961215, 1.465184, 0.389497, 0.287837] with:

Minimum: 0.287837 Maximum: 1.465184 Average: 0.7533 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9%| 110/1261 [04:54<51:16, 2.67s/it]

The times for each task are: [1.374398, 0.64732, 0.933367, 0.322443, 0.503108] with:
Minimum: 0.322443 Maximum: 1.374398 Average: 0.7561 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9%| 111/1261 [04:56<51:12, 2.67s/it]

The times for each task are: [0.591337, 0.910857, 1.703674, 0.394275, 0.401828] with:
Minimum: 0.394275 Maximum: 1.703674 Average: 0.8004 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9%| 112/1261 [04:59<51:13, 2.67s/it]

The times for each task are: [1.001603, 1.33326, 0.710664, 0.336354, 0.493835] with:
Minimum: 0.336354 Maximum: 1.33326 Average: 0.7751 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9%| 113/1261 [05:02<51:13, 2.68s/it]

The times for each task are: [0.921541, 1.420664, 0.454498, 0.710926, 0.327714] with:
Minimum: 0.327714 Maximum: 1.420664 Average: 0.7671 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 114/1261 [05:05<51:12, 2.68s/it]

The times for each task are: [0.661688, 1.493132, 1.054888, 0.444314, 0.363646] with:
Minimum: 0.363646 Maximum: 1.493132 Average: 0.8035 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 115/1261 [05:08<51:09, 2.68s/it]

The times for each task are: [0.637344, 1.073884, 1.436338, 0.381582, 0.329967] with:
Minimum: 0.329967 Maximum: 1.436338 Average: 0.7718 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 116/1261 [05:10<51:05, 2.68s/it]

The times for each task are: [0.918808, 0.587854, 1.356473, 0.437473, 0.292795] with:
Minimum: 0.292795 Maximum: 1.356473 Average: 0.7187 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 117/1261 [05:13<51:00, 2.68s/it]

The times for each task are: [1.095957, 0.639341, 1.453686, 0.297153, 0.503111] with:
Minimum: 0.297153 Maximum: 1.453686 Average: 0.7978 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

9% | 118/1261 [05:15<50:56, 2.67s/it]

The times for each task are: [0.891475, 0.623622, 1.498675, 0.450705, 0.295802] with:

Minimum: 0.295802 Maximum: 1.498675 Average: 0.7521 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

9% | 119/1261 [05:18<50:55, 2.68s/it]

The times for each task are: [0.887894, 0.673111, 1.562471, 0.418972, 0.38499] with:

Minimum: 0.38499 Maximum: 1.562471 Average: 0.7855 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

10% | 120/1261 [05:21<50:55, 2.68s/it]

The times for each task are: [0.913198, 1.450468, 0.76804, 0.488575, 0.35692] with:

Minimum: 0.35692 Maximum: 1.450468 Average: 0.7954 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

10% | 121/1261 [05:23<50:52, 2.68s/it]

The times for each task are: [0.586843, 0.92026, 1.694898, 0.387044, 0.435327] with:

Minimum: 0.387044 Maximum: 1.694898 Average: 0.8049 seconds

Number of cars 0
Length of task list: 5

Number of processes used: 3

10%| 122/1261 [05:26<50:47, 2.68s/it]

The times for each task are: [0.642607, 0.921982, 1.398946, 0.317349, 0.447469] with:

Minimum: 0.317349 Maximum: 1.398946 Average: 0.7457 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10%| 123/1261 [05:29<50:44, 2.68s/it]

The times for each task are: [0.87644, 1.448568, 0.718177, 0.492826, 0.345563] with:

Minimum: 0.345563 Maximum: 1.448568 Average: 0.7763 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10%| 124/1261 [05:31<50:42, 2.68s/it]

The times for each task are: [1.451751, 1.021914, 0.777584, 0.336521, 0.475143] with:

Minimum: 0.336521 Maximum: 1.451751 Average: 0.8126 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10%| 125/1261 [05:34<50:41, 2.68s/it]

The times for each task are: [0.980473, 0.620987, 1.698389, 0.481891, 0.420977] with:

Minimum: 0.420977 Maximum: 1.698389 Average: 0.8405 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 126/1261 [05:37<50:39, 2.68s/it]

The times for each task are: [0.695911, 1.613901, 1.199383, 0.4564, 0.325583] with:

Minimum: 0.325583 Maximum: 1.613901 Average: 0.8582 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 127/1261 [05:40<50:37, 2.68s/it]

The times for each task are: [0.60473, 1.543067, 0.983723, 0.315447, 0.411913] with:

Minimum: 0.315447 Maximum: 1.543067 Average: 0.7718 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 128/1261 [05:42<50:35, 2.68s/it]

The times for each task are: [0.92797, 0.62635, 1.67275, 0.362328, 0.388138] with:

Minimum: 0.362328 Maximum: 1.67275 Average: 0.7955 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 129/1261 [05:45<50:32, 2.68s/it]

The times for each task are: [1.472502, 0.638224, 1.095911, 0.340912, 0.39994] with:

Minimum: 0.340912 Maximum: 1.472502 Average: 0.7895 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 130/1261 [05:48<50:28, 2.68s/it]

The times for each task are: [0.676948, 0.85626, 1.481064, 0.432216, 0.392638] with:

Minimum: 0.392638 Maximum: 1.481064 Average: 0.7678 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 131/1261 [05:50<50:26, 2.68s/it]

The times for each task are: [0.658312, 0.919249, 1.762973, 0.295743, 0.474711] with:

Minimum: 0.295743 Maximum: 1.762973 Average: 0.8222 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

10% | 132/1261 [05:53<50:23, 2.68s/it]

The times for each task are: [0.912273, 0.589067, 1.50305, 0.339741, 0.391806] with:

Minimum: 0.339741 Maximum: 1.50305 Average: 0.7472 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 133/1261 [05:56<50:21, 2.68s/it]

The times for each task are: [1.401741, 0.986811, 0.670223, 0.333896, 0.448608] with:

Minimum: 0.333896 Maximum: 1.401741 Average: 0.7683 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 134/1261 [05:59<50:20, 2.68s/it]

The times for each task are: [1.372336, 0.777461, 0.913905, 0.479646, 0.3371] with:

Minimum: 0.3371 Maximum: 1.372336 Average: 0.7761 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 135/1261 [06:01<50:17, 2.68s/it]

The times for each task are: [1.064824, 0.646891, 1.418029, 0.467777, 0.368114] with:

Minimum: 0.368114 Maximum: 1.418029 Average: 0.7931 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 136/1261 [06:04<50:14, 2.68s/it]

The times for each task are: [1.005614, 1.431251, 0.612485, 0.28888, 0.452014] with:

Minimum: 0.28888 Maximum: 1.431251 Average: 0.758 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 137/1261 [06:07<50:11, 2.68s/it]

The times for each task are: [0.895792, 0.61591, 1.397777, 0.345925, 0.444861] with:

Minimum: 0.345925 Maximum: 1.397777 Average: 0.7401 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

11% | 138/1261 [06:09<50:08, 2.68s/it]

The times for each task are: [0.949311, 1.41224, 0.662746, 0.490137, 0.312008] with:

Minimum: 0.312008 Maximum: 1.41224 Average: 0.7653 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 139/1261 [06:12<50:06, 2.68s/it]

The times for each task are: [0.88852, 0.763336, 1.614694, 0.40822, 0.35435] with:

Minimum: 0.35435 Maximum: 1.614694 Average: 0.8058 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 140/1261 [06:15<50:02, 2.68s/it]

The times for each task are: [0.921415, 1.397324, 0.654775, 0.29142, 0.387874] with:

Minimum: 0.29142 Maximum: 1.397324 Average: 0.7306 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 141/1261 [06:17<50:00, 2.68s/it]

The times for each task are: [1.054229, 1.443215, 0.674206, 0.496318, 0.288673] with:

Minimum: 0.288673 Maximum: 1.443215 Average: 0.7913 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 142/1261 [06:20<49:57, 2.68s/it]

The times for each task are: [0.839613, 1.400012, 0.767975, 0.513731, 0.348497] with:

Minimum: 0.348497 Maximum: 1.400012 Average: 0.774 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 143/1261 [06:23<49:55, 2.68s/it]

The times for each task are: [1.438322, 0.631465, 0.901356, 0.342379, 0.43757] with:
Minimum: 0.342379 Maximum: 1.438322 Average: 0.7502 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 144/1261 [06:25<49:49, 2.68s/it]

The times for each task are: [0.937438, 0.683524, 1.392732, 0.466028, 0.36545] with:
Minimum: 0.36545 Maximum: 1.392732 Average: 0.769 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

11% | 145/1261 [06:28<49:47, 2.68s/it]

The times for each task are: [0.685793, 0.891895, 1.409395, 0.297455, 0.514266] with:
Minimum: 0.297455 Maximum: 1.409395 Average: 0.7598 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

12% | 146/1261 [06:30<49:44, 2.68s/it]

The times for each task are: [0.6498, 0.894721, 0.300742, 1.46089, 0.478998] with:
Minimum: 0.300742 Maximum: 1.46089 Average: 0.757 seconds

Number of cars 0

Length of task list: 5
Number of processes used: 3

12% | 147/1261 [06:33<49:39, 2.67s/it]

The times for each task are: [0.851113, 0.699101, 1.476077, 0.385573, 0.452969] with:

Minimum: 0.385573 Maximum: 1.476077 Average: 0.773 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

12% | 148/1261 [06:36<49:38, 2.68s/it]

The times for each task are: [1.060781, 0.663144, 1.58594, 0.441994, 0.340061] with:

Minimum: 0.340061 Maximum: 1.58594 Average: 0.8184 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

12% | 149/1261 [06:38<49:37, 2.68s/it]

The times for each task are: [0.63083, 1.396615, 1.038589, 0.373991, 0.453028] with:

Minimum: 0.373991 Maximum: 1.396615 Average: 0.7786 seconds

Number of cars 0
Length of task list: 5
Number of processes used: 3

12% | 150/1261 [06:41<49:35, 2.68s/it]

The times for each task are: [0.653247, 0.940674, 1.401812, 0.312, 0.467482] with:

Minimum: 0.312 Maximum: 1.401812 Average: 0.755 seconds

Number of cars 0
Length of task list: 5

Number of processes used: 3

12%| 151/1261 [06:44<49:31, 2.68s/it]

The times for each task are: [0.656863, 0.927305, 1.482618, 0.332898, 0.43358] with:

Minimum: 0.332898 Maximum: 1.482618 Average: 0.7667 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12%| 152/1261 [06:46<49:28, 2.68s/it]

The times for each task are: [0.911062, 1.381373, 0.637559, 0.294414, 0.380476] with:

Minimum: 0.294414 Maximum: 1.381373 Average: 0.721 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12%| 153/1261 [06:49<49:25, 2.68s/it]

The times for each task are: [0.63281, 1.052752, 1.397067, 0.344421, 0.388892] with:

Minimum: 0.344421 Maximum: 1.397067 Average: 0.7632 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12%| 154/1261 [06:52<49:21, 2.68s/it]

The times for each task are: [0.852048, 0.728259, 0.480642, 1.508742, 0.335758] with:

Minimum: 0.335758 Maximum: 1.508742 Average: 0.7811 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12%| 155/1261 [06:54<49:20, 2.68s/it]

The times for each task are: [1.487888, 0.944173, 0.676872, 0.379666, 0.465547] with:

Minimum: 0.379666 Maximum: 1.487888 Average: 0.7908 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12%| 156/1261 [06:57<49:17, 2.68s/it]

The times for each task are: [0.890218, 0.720387, 1.619566, 0.414853, 0.343795] with:

Minimum: 0.343795 Maximum: 1.619566 Average: 0.7978 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

12%| 157/1261 [07:00<49:15, 2.68s/it]

The times for each task are: [0.620625, 0.845325, 0.39395, 1.610366, 0.290754] with:

Minimum: 0.290754 Maximum: 1.610366 Average: 0.7522 seconds

Number of cars 1

bbox:car_number ((1175, 460), (1212, 498)) : 1

Length of task list: 5

Number of processes used: 3

13%| 158/1261 [07:02<49:12, 2.68s/it]

The times for each task are: [0.632625, 0.908962, 1.484664, 0.424065, 0.349597] with:

Minimum: 0.349597 Maximum: 1.484664 Average: 0.76 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

13%| 159/1261 [07:05<49:09, 2.68s/it]

The times for each task are: [0.712847, 1.061859, 1.469735, 0.366015, 0.438281] with:

Minimum: 0.366015 Maximum: 1.469735 Average: 0.8097 seconds

Number of cars 1
bbox:car_number ((1217, 438), (1219, 459)) : 1
Length of task list: 5
Number of processes used: 3

13% | 160/1261 [07:08<49:05, 2.68s/it]

The times for each task are: [0.902109, 0.623809, 1.411691, 0.304649, 0.499552] with:

Minimum: 0.304649 Maximum: 1.411691 Average: 0.7484 seconds

Number of cars 1
bbox:car_number ((1204, 400), (1273, 478)) : 1
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478))]
Length of task list: 5
Number of processes used: 3

13% | 161/1261 [07:10<49:01, 2.67s/it]

The times for each task are: [0.988044, 0.694217, 1.386558, 0.355363, 0.397251] with:

Minimum: 0.355363 Maximum: 1.386558 Average: 0.7643 seconds

Number of cars 1
bbox:car_number ((1175, 400), (1273, 512)) : 1
The minimum distance from car: 1 is 22.02271554554524
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

13% | 162/1261 [07:12<48:56, 2.67s/it]

The times for each task are: [0.82891, 1.60463, 0.711976, 0.412785, 0.350079] with:

Minimum: 0.350079 Maximum: 1.60463 Average: 0.7817 seconds

Number of cars 1

```
bbox:car_number ((1183, 400), (1273, 497)) : 1
The minimum distance from car: 1 is 8.94427190999916
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13% | 163/1261 [07:15<48:53, 2.67s/it]

The times for each task are: [0.969186, 0.640746, 1.625646, 0.391783, 0.292245] with:
Minimum: 0.292245 Maximum: 1.625646 Average: 0.7839 seconds

```
Number of cars 1
bbox:car_number ((1175, 400), (1273, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13% | 164/1261 [07:18<48:51, 2.67s/it]

The times for each task are: [0.967227, 0.712385, 1.429132, 0.372342, 0.385125] with:
Minimum: 0.372342 Maximum: 1.429132 Average: 0.7732 seconds

```
Number of cars 1
bbox:car_number ((1175, 400), (1273, 512)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13% | 165/1261 [07:21<48:50, 2.67s/it]

The times for each task are: [0.863071, 0.624754, 0.404482, 1.367445, 0.294135] with:
Minimum: 0.294135 Maximum: 1.367445 Average: 0.7108 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1273, 512)) : 1
```

The minimum distance from car: 1 is 16.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

13% | 166/1261 [07:23<48:46, 2.67s/it]

The times for each task are: [0.926965, 0.685406, 1.383852, 0.288535, 0.388481] with:

Minimum: 0.288535 Maximum: 1.383852 Average: 0.7346 seconds

Number of cars 1

bbox:car_number ((1144, 400), (1257, 498)) : 1

The minimum distance from car: 1 is 10.63014581273465

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

13% | 167/1261 [07:26<48:43, 2.67s/it]

The times for each task are: [0.952272, 0.657116, 1.436893, 0.295131, 0.546655] with:

Minimum: 0.295131 Maximum: 1.436893 Average: 0.7776 seconds

Number of cars 1

bbox:car_number ((1156, 400), (1257, 497)) : 1

The minimum distance from car: 1 is 6.082762530298219

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

13% | 168/1261 [07:29<48:41, 2.67s/it]

The times for each task are: [0.627009, 1.379601, 0.945713, 0.412406, 0.298706] with:

Minimum: 0.298706 Maximum: 1.379601 Average: 0.7327 seconds

Number of cars 1

bbox:car_number ((1144, 400), (1257, 498)) : 1

The minimum distance from car: 1 is 6.082762530298219

```
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13%| 169/1261 [07:31<48:38, 2.67s/it]

The times for each task are: [1.468139, 0.891916, 0.750105, 0.472428, 0.36958] with:

Minimum: 0.36958 Maximum: 1.468139 Average: 0.7904 seconds

```
Number of cars 1
bbox:car_number ((1141, 400), (1257, 507)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

13%| 170/1261 [07:33<48:33, 2.67s/it]

The times for each task are: [0.594339, 0.962516, 1.540889, 0.432927, 0.327572] with:

Minimum: 0.327572 Maximum: 1.540889 Average: 0.7716 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 7.211102550927978
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14%| 171/1261 [07:36<48:31, 2.67s/it]

The times for each task are: [1.003739, 0.653901, 1.496896, 0.419649, 0.416369] with:

Minimum: 0.416369 Maximum: 1.496896 Average: 0.7981 seconds

```
Number of cars 2
bbox:car_number ((1156, 400), (1253, 498)) : 1
The minimum distance from car: 1 is 11.0
bbox:car_number ((1075, 438), (1122, 459)) : 2
```

```
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14% | 172/1261 [07:39<48:28, 2.67s/it]

The times for each task are: [0.886224, 1.391537, 0.700301, 0.410199, 0.391677] with:

Minimum: 0.391677 Maximum: 1.391537 Average: 0.756 seconds

```
Number of cars 1
bbox:car_number ((1135, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 17.46424919657298
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14% | 173/1261 [07:41<48:24, 2.67s/it]

The times for each task are: [0.977905, 0.62086, 1.443937, 0.401718, 0.29073] with:

Minimum: 0.29073 Maximum: 1.443937 Average: 0.747 seconds

```
Number of cars 2
bbox:car_number ((1129, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1113, 438), (1124, 497)) : 2
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14% | 174/1261 [07:44<48:20, 2.67s/it]

The times for each task are: [0.616774, 1.050123, 0.398157, 1.469999, 0.349167] with:

Minimum: 0.349167 Maximum: 1.469999 Average: 0.7768 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1242, 497)) : 1
The minimum distance from car: 1 is 11.313708498984761
```

```
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14%| 175/1261 [07:47<48:18, 2.67s/it]

The times for each task are: [1.412472, 0.606899, 0.880891, 0.414932, 0.415015] with:

Minimum: 0.414932 Maximum: 1.412472 Average: 0.746 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1238, 497)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14%| 176/1261 [07:49<48:14, 2.67s/it]

The times for each task are: [0.574685, 1.379101, 1.030748, 0.2913, 0.392792] with:

Minimum: 0.2913 Maximum: 1.379101 Average: 0.7337 seconds

```
Number of cars 1
bbox:car_number ((1144, 400), (1238, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

14%| 177/1261 [07:52<48:13, 2.67s/it]

The times for each task are: [0.882917, 0.639834, 1.376739, 0.42189, 0.359288] with:

Minimum: 0.359288 Maximum: 1.376739 Average: 0.7361 seconds

```
Number of cars 1
bbox:car_number ((1120, 400), (1230, 498)) : 1
The minimum distance from car: 1 is 16.0
totalCars: 1
```

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

14% | 178/1261 [07:55<48:10, 2.67s/it]

The times for each task are: [0.5866, 1.421695, 0.893527, 0.388004, 0.303912] with:

Minimum: 0.303912 Maximum: 1.421695 Average: 0.7187 seconds

Number of cars 1
bbox:car_number ((1120, 400), (1230, 507)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

14% | 179/1261 [07:57<48:07, 2.67s/it]

The times for each task are: [0.585196, 0.906719, 1.472619, 0.294333, 0.405542] with:

Minimum: 0.294333 Maximum: 1.472619 Average: 0.7329 seconds

Number of cars 1
bbox:car_number ((1113, 400), (1227, 512)) : 1
The minimum distance from car: 1 is 5.830951894845301
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

14% | 180/1261 [08:00<48:05, 2.67s/it]

The times for each task are: [1.050929, 1.498646, 0.58483, 0.487004, 0.303718] with:

Minimum: 0.303718 Maximum: 1.498646 Average: 0.785 seconds

Number of cars 1
bbox:car_number ((1120, 400), (1230, 512)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

```
Length of task list: 5
Number of processes used: 3
```

```
14%| 181/1261 [08:03<48:02, 2.67s/it]
```

```
The times for each task are: [0.846915, 0.598649, 1.388462, 0.363807, 0.442405] with:
```

```
Minimum: 0.363807 Maximum: 1.388462 Average: 0.728 seconds
```

```
Number of cars 1
bbox:car_number ((1113, 400), (1230, 516)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1219, 512))]
Length of task list: 5
Number of processes used: 3
```

```
14%| 182/1261 [08:05<47:58, 2.67s/it]
```

```
The times for each task are: [0.725571, 0.823246, 0.405589, 1.572452, 0.29887] with:
```

```
Minimum: 0.29887 Maximum: 1.572452 Average: 0.7651 seconds
```

```
Number of cars 1
bbox:car_number ((1113, 400), (1219, 512)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1219, 512))]
Length of task list: 5
Number of processes used: 3
```

```
15%| 183/1261 [08:08<47:55, 2.67s/it]
```

```
The times for each task are: [0.910592, 0.701661, 1.606459, 0.485612, 0.293795] with:
```

```
Minimum: 0.293795 Maximum: 1.606459 Average: 0.7996 seconds
```

```
Number of cars 1
bbox:car_number ((1113, 400), (1219, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1219, 512))]
Length of task list: 5
```

Number of processes used: 3

15%| 184/1261 [08:10<47:53, 2.67s/it]

The times for each task are: [0.865535, 1.400857, 0.409349, 0.659385, 0.415524] with:

Minimum: 0.409349 Maximum: 1.400857 Average: 0.7501 seconds

Number of cars 1

bbox:car_number ((1129, 400), (1214, 498)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

15%| 185/1261 [08:13<47:52, 2.67s/it]

The times for each task are: [0.909136, 0.727982, 1.596864, 0.310303, 0.442504] with:

Minimum: 0.310303 Maximum: 1.596864 Average: 0.7974 seconds

Number of cars 1

bbox:car_number ((1105, 400), (1214, 512)) : 1

The minimum distance from car: 1 is 13.892443989449804

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

15%| 186/1261 [08:16<47:50, 2.67s/it]

The times for each task are: [1.344742, 0.91895, 0.604869, 0.291606, 0.415905] with:

Minimum: 0.291606 Maximum: 1.344742 Average: 0.7152 seconds

Number of cars 1

bbox:car_number ((1113, 400), (1214, 507)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

15%| 187/1261 [08:19<47:47, 2.67s/it]

The times for each task are: [0.838227, 0.623729, 1.577781, 0.309984, 0.469812] with:

Minimum: 0.309984 Maximum: 1.577781 Average: 0.7639 seconds

Number of cars 1
bbox:car_number ((1103, 400), (1212, 507)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

15%| 188/1261 [08:21<47:44, 2.67s/it]

The times for each task are: [0.883108, 0.632293, 1.424211, 0.435855, 0.390134] with:

Minimum: 0.390134 Maximum: 1.424211 Average: 0.7531 seconds

Number of cars 1
bbox:car_number ((1103, 400), (1212, 516)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

15%| 189/1261 [08:24<47:42, 2.67s/it]

The times for each task are: [1.378059, 0.915323, 0.763708, 0.483198, 0.341253] with:

Minimum: 0.341253 Maximum: 1.378059 Average: 0.7763 seconds

Number of cars 1
bbox:car_number ((1102, 400), (1214, 518)) : 1
The minimum distance from car: 1 is 1.4142135623730951
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

15% | 190/1261 [08:27<47:39, 2.67s/it]

The times for each task are: [0.977859, 0.666468, 1.622437, 0.467045, 0.346041] with:

Minimum: 0.346041 Maximum: 1.622437 Average: 0.816 seconds

Number of cars 1
bbox:car_number ((1113, 400), (1212, 512)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1225, 475))]
Length of task list: 5
Number of processes used: 3

15% | 191/1261 [08:29<47:36, 2.67s/it]

The times for each task are: [0.614068, 0.975486, 0.387189, 0.400674, 1.638056] with:

Minimum: 0.387189 Maximum: 1.638056 Average: 0.8031 seconds

Number of cars 1
bbox:car_number ((1103, 400), (1212, 516)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1225, 475))]
Length of task list: 5
Number of processes used: 3

15% | 192/1261 [08:32<47:33, 2.67s/it]

The times for each task are: [0.884358, 0.646647, 1.411332, 0.4892, 0.323389] with:

Minimum: 0.323389 Maximum: 1.411332 Average: 0.751 seconds

Number of cars 2
bbox:car_number ((1105, 400), (1214, 518)) : 1
The minimum distance from car: 1 is 2.23606797749979
bbox:car_number ((1225, 475), (1257, 477)) : 2
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1225, 475))]
Length of task list: 5
Number of processes used: 3

15%| 193/1261 [08:35<47:31, 2.67s/it]

The times for each task are: [0.976809, 0.659518, 1.541857, 0.312052, 0.395647] with:

Minimum: 0.312052 Maximum: 1.541857 Average: 0.7772 seconds

Number of cars 1

bbox:car_number ((1102, 400), (1211, 512)) : 1

The minimum distance from car: 1 is 4.242640687119285

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1211, 498))]

Length of task list: 5

Number of processes used: 3

15%| 194/1261 [08:37<47:28, 2.67s/it]

The times for each task are: [0.584205, 0.914479, 1.67139, 0.395663, 0.297802] with:

Minimum: 0.297802 Maximum: 1.67139 Average: 0.7727 seconds

Number of cars 1

bbox:car_number ((1113, 400), (1211, 498)) : 1

The minimum distance from car: 1 is 9.219544457292887

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1211, 498))]

Length of task list: 5

Number of processes used: 3

15%| 195/1261 [08:40<47:26, 2.67s/it]

The times for each task are: [1.372273, 0.696716, 1.018749, 0.452859, 0.290527] with:

Minimum: 0.290527 Maximum: 1.372273 Average: 0.7662 seconds

Number of cars 1

bbox:car_number ((1113, 400), (1211, 512)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1211, 498))]

Length of task list: 5

Number of processes used: 3

16%| 196/1261 [08:42<47:20, 2.67s/it]

The times for each task are: [0.87236, 0.667928, 1.652731, 0.305632, 0.398756] with:

Minimum: 0.305632 Maximum: 1.652731 Average: 0.7795 seconds

Number of cars 2

bbox:car_number ((1090, 400), (1211, 512)) : 1

The minimum distance from car: 1 is 12.0

bbox:car_number ((1225, 431), (1272, 477)) : 2

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 197/1261 [08:45<47:19, 2.67s/it]

The times for each task are: [1.380034, 0.956854, 0.634541, 0.411707, 0.43787] with:

Minimum: 0.411707 Maximum: 1.380034 Average: 0.7642 seconds

Number of cars 2

bbox:car_number ((1090, 400), (1211, 512)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1225, 430), (1272, 477)) : 2

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 198/1261 [08:48<47:15, 2.67s/it]

The times for each task are: [0.913289, 0.698774, 1.521235, 0.404098, 0.392874] with:

Minimum: 0.392874 Maximum: 1.521235 Average: 0.7861 seconds

Number of cars 1

bbox:car_number ((1113, 400), (1272, 512)) : 1

The minimum distance from car: 1 is 42.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 199/1261 [08:50<47:12, 2.67s/it]

The times for each task are: [0.643658, 0.902806, 0.331571, 1.425437, 0.409413] with:

Minimum: 0.331571 Maximum: 1.425437 Average: 0.7426 seconds

Number of cars 1

bbox:car_number ((1102, 400), (1273, 498)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 200/1261 [08:53<47:09, 2.67s/it]

The times for each task are: [0.940968, 0.694272, 1.439247, 0.408027, 0.293681] with:

Minimum: 0.293681 Maximum: 1.439247 Average: 0.7552 seconds

Number of cars 1

bbox:car_number ((1090, 400), (1273, 512)) : 1

The minimum distance from car: 1 is 9.219544457292887

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 201/1261 [08:56<47:06, 2.67s/it]

The times for each task are: [0.675389, 0.855693, 1.368427, 0.488383, 0.376159] with:

Minimum: 0.376159 Maximum: 1.368427 Average: 0.7528 seconds

Number of cars 1

bbox:car_number ((1102, 400), (1272, 498)) : 1

The minimum distance from car: 1 is 9.219544457292887

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

16% | 202/1261 [08:58<47:05, 2.67s/it]

The times for each task are: [1.101179, 1.449289, 0.649094, 0.391022, 0.287403] with:

Minimum: 0.287403 Maximum: 1.449289 Average: 0.7756 seconds

Number of cars 1
bbox:car_number ((1102, 400), (1273, 512)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

16% | 203/1261 [09:01<47:02, 2.67s/it]

The times for each task are: [0.886712, 0.751105, 1.397789, 0.408887, 0.347484] with:

Minimum: 0.347484 Maximum: 1.397789 Average: 0.7584 seconds

Number of cars 1
bbox:car_number ((1102, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

16% | 204/1261 [09:04<46:59, 2.67s/it]

The times for each task are: [0.906246, 1.437406, 0.748001, 0.407917, 0.315989] with:

Minimum: 0.315989 Maximum: 1.437406 Average: 0.7631 seconds

Number of cars 1
bbox:car_number ((1102, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

16% | 205/1261 [09:06<46:56, 2.67s/it]

The times for each task are: [0.897601, 1.390741, 0.584098, 0.403031, 0.392369] with:

Minimum: 0.392369 Maximum: 1.390741 Average: 0.7336 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1272, 518)) : 1
The minimum distance from car: 1 is 8.54400374531753
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1268, 518))]
Length of task list: 5
Number of processes used: 3
```

16% | 206/1261 [09:09<46:53, 2.67s/it]

The times for each task are: [0.731732, 0.961282, 1.375372, 0.310982, 0.428651] with:

Minimum: 0.310982 Maximum: 1.375372 Average: 0.7616 seconds

```
Number of cars 1
bbox:car_number ((1084, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1268, 518))]
Length of task list: 5
Number of processes used: 3
```

16% | 207/1261 [09:12<46:51, 2.67s/it]

The times for each task are: [1.301767, 0.889543, 0.741627, 0.317997, 0.487523] with:

Minimum: 0.317997 Maximum: 1.301767 Average: 0.7477 seconds

```
Number of cars 1
bbox:car_number ((1084, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1268, 518))]
Length of task list: 5
Number of processes used: 3
```

16% | 208/1261 [09:14<46:49, 2.67s/it]

The times for each task are: [0.858267, 0.624195, 1.671809, 0.411166, 0.400514] with:

Minimum: 0.400514 Maximum: 1.671809 Average: 0.7932 seconds

```
Number of cars 1
bbox:car_number ((1084, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17% | 209/1261 [09:17<46:46, 2.67s/it]

The times for each task are: [0.635545, 0.878361, 1.428261, 0.308781, 0.44555] with:
Minimum: 0.308781 Maximum: 1.428261 Average: 0.7393 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17% | 210/1261 [09:20<46:43, 2.67s/it]

The times for each task are: [0.829699, 1.369587, 0.746552, 0.41684, 0.288698] with:
Minimum: 0.288698 Maximum: 1.369587 Average: 0.7303 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17% | 211/1261 [09:22<46:41, 2.67s/it]

The times for each task are: [0.69959, 0.881967, 0.495203, 1.588619, 0.39758] with:
Minimum: 0.39758 Maximum: 1.588619 Average: 0.8126 seconds

Number of cars 1

```
bbox:car_number ((1090, 400), (1268, 498)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 507))]
Length of task list: 5
Number of processes used: 3
```

17% | 212/1261 [09:25<46:38, 2.67s/it]

The times for each task are: [1.615696, 0.873517, 0.709759, 0.304572, 0.394967] with:
Minimum: 0.304572 Maximum: 1.615696 Average: 0.7797 seconds

```
Number of cars 1
bbox:car_number ((1090, 400), (1268, 498)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 507))]
Length of task list: 5
Number of processes used: 3
```

17% | 213/1261 [09:28<46:37, 2.67s/it]

The times for each task are: [0.634781, 0.917539, 1.753278, 0.417263, 0.379306] with:
Minimum: 0.379306 Maximum: 1.753278 Average: 0.8204 seconds

```
Number of cars 1
bbox:car_number ((1090, 400), (1257, 507)) : 1
The minimum distance from car: 1 is 7.211102550927978
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]
Length of task list: 5
Number of processes used: 3
```

17% | 214/1261 [09:31<46:34, 2.67s/it]

The times for each task are: [0.880261, 0.682851, 1.665154, 0.437531, 0.52244] with:
Minimum: 0.437531 Maximum: 1.665154 Average: 0.8376 seconds

```
Number of cars 1
bbox:car_number ((1090, 400), (1257, 512)) : 1
```

The minimum distance from car: 1 is 3.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

17% | 215/1261 [09:34<46:32, 2.67s/it]

The times for each task are: [0.903395, 0.64768, 1.740895, 0.500538, 0.365119] with:

Minimum: 0.365119 Maximum: 1.740895 Average: 0.8315 seconds

Number of cars 1

bbox:car_number ((1082, 400), (1257, 498)) : 1

The minimum distance from car: 1 is 8.06225774829855

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

17% | 216/1261 [09:36<46:29, 2.67s/it]

The times for each task are: [0.916175, 0.65789, 0.42925, 1.783238, 0.385415] with:

Minimum: 0.385415 Maximum: 1.783238 Average: 0.8344 seconds

Number of cars 1

bbox:car_number ((1082, 400), (1268, 498)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40

Length of task list: 5

Number of processes used: 3

17% | 217/1261 [09:39<46:27, 2.67s/it]

The times for each task are: [0.754969, 0.886317, 1.469031, 0.304631, 0.440187] with:

Minimum: 0.304631 Maximum: 1.469031 Average: 0.771 seconds

Number of cars 1

bbox:car_number ((1082, 400), (1257, 512)) : 1

The minimum distance from car: 1 is 9.219544457292887

```
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17%| 218/1261 [09:42<46:24, 2.67s/it]

The times for each task are: [0.610744, 1.007134, 0.417829, 0.31974, 1.6041] with:

Minimum: 0.31974 Maximum: 1.6041 Average: 0.7919 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1268, 498)) : 1
The minimum distance from car: 1 is 9.219544457292887
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17%| 219/1261 [09:44<46:21, 2.67s/it]

The times for each task are: [0.895057, 0.65818, 1.424799, 0.461614, 0.324356] with:

Minimum: 0.324356 Maximum: 1.424799 Average: 0.7528 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1268, 512)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

17%| 220/1261 [09:47<46:18, 2.67s/it]

The times for each task are: [0.653648, 0.404412, 0.926441, 1.657827, 0.298716] with:

Minimum: 0.298716 Maximum: 1.657827 Average: 0.7882 seconds

```
Number of cars 1
bbox:car_number ((1082, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 13.0
totalCars: 1
```

```
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]  
Length of task list: 5  
Number of processes used: 3
```

18% | 221/1261 [09:50<46:16, 2.67s/it]

The times for each task are: [1.376059, 0.644394, 0.945973, 0.477528, 0.341687] with:

Minimum: 0.341687 Maximum: 1.376059 Average: 0.7571 seconds

```
Number of cars 1  
bbox:car_number ((1082, 400), (1257, 498)) : 1  
The minimum distance from car: 1 is 9.899494936611665  
totalCars: 1  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]  
Length of task list: 5  
Number of processes used: 3
```

18% | 222/1261 [09:52<46:13, 2.67s/it]

The times for each task are: [0.587444, 0.920112, 1.592113, 0.328614, 0.483078] with:

Minimum: 0.328614 Maximum: 1.592113 Average: 0.7823 seconds

```
Number of cars 1  
bbox:car_number ((1082, 400), (1257, 512)) : 1  
The minimum distance from car: 1 is 7.0  
totalCars: 1  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]  
Length of task list: 5  
Number of processes used: 3
```

18% | 223/1261 [09:55<46:11, 2.67s/it]

The times for each task are: [0.61026, 1.003798, 1.407887, 0.403335, 0.326313] with:

Minimum: 0.326313 Maximum: 1.407887 Average: 0.7503 seconds

```
Number of cars 1  
bbox:car_number ((1082, 400), (1253, 512)) : 1  
The minimum distance from car: 1 is 2.0  
totalCars: 1  
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
18%| 224/1261 [09:58<46:09, 2.67s/it]
```

```
The times for each task are: [0.927234, 0.702246, 1.528047, 0.482563, 0.343543] with:
```

```
Minimum: 0.343543 Maximum: 1.528047 Average: 0.7967 seconds
```

```
Number of cars 1
bbox:car_number ((1075, 400), (1257, 507)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 507))]
Length of task list: 5
Number of processes used: 3
```

```
18%| 225/1261 [10:01<46:07, 2.67s/it]
```

```
The times for each task are: [0.894084, 1.507702, 0.654137, 0.391411, 0.300241] with:
```

```
Minimum: 0.300241 Maximum: 1.507702 Average: 0.7495 seconds
```

```
Number of cars 2
bbox:car_number ((1075, 400), (1257, 512)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1051, 431), (1070, 495)) : 2
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]
Length of task list: 5
Number of processes used: 3
```

```
18%| 226/1261 [10:03<46:05, 2.67s/it]
```

```
The times for each task are: [0.876326, 0.686332, 1.364503, 0.384218, 0.4782] with:
```

```
Minimum: 0.384218 Maximum: 1.364503 Average: 0.7579 seconds
```

```
Number of cars 1
bbox:car_number ((1082, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
18%|           | 227/1261 [10:06<46:03,  2.67s/it]
```

```
The times for each task are: [0.656208, 1.350078, 1.023201, 0.300445, 0.400771] with:
```

```
Minimum: 0.300445 Maximum: 1.350078 Average: 0.7461 seconds
```

```
Number of cars 1
bbox:car_number ((1075, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1242, 498))]
Length of task list: 5
Number of processes used: 3
```

```
18%|           | 228/1261 [10:09<46:00,  2.67s/it]
```

```
The times for each task are: [0.935815, 1.387245, 0.757664, 0.366327, 0.462385] with:
```

```
Minimum: 0.366327 Maximum: 1.387245 Average: 0.7819 seconds
```

```
Number of cars 1
bbox:car_number ((1075, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1242, 498))]
Length of task list: 5
Number of processes used: 3
```

```
18%|           | 229/1261 [10:11<45:57,  2.67s/it]
```

```
The times for each task are: [1.00053, 0.665663, 1.496689, 0.473675, 0.307483] with:
```

```
Minimum: 0.307483 Maximum: 1.496689 Average: 0.7888 seconds
```

```
Number of cars 1
bbox:car_number ((1065, 400), (1257, 512)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]
Length of task list: 5
```

Number of processes used: 3

18%| 230/1261 [10:14<45:54, 2.67s/it]

The times for each task are: [1.366182, 0.966392, 0.717217, 0.301084, 0.492755] with:

Minimum: 0.301084 Maximum: 1.366182 Average: 0.7687 seconds

Number of cars 1

bbox:car_number ((1065, 400), (1242, 512)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 498))]

Length of task list: 5

Number of processes used: 3

18%| 231/1261 [10:17<45:51, 2.67s/it]

The times for each task are: [0.619832, 1.412038, 0.904939, 0.402964, 0.282328] with:

Minimum: 0.282328 Maximum: 1.412038 Average: 0.7244 seconds

Number of cars 1

bbox:car_number ((1065, 400), (1241, 498)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 498))]

Length of task list: 5

Number of processes used: 3

18%| 232/1261 [10:19<45:49, 2.67s/it]

The times for each task are: [1.371715, 0.718935, 0.990697, 0.391855, 0.291853] with:

Minimum: 0.291853 Maximum: 1.371715 Average: 0.753 seconds

Number of cars 1

bbox:car_number ((1067, 400), (1242, 498)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1242, 498))]

Length of task list: 5

Number of processes used: 3

18%| 233/1261 [10:22<45:47, 2.67s/it]

The times for each task are: [0.598693, 1.384021, 0.987068, 0.395033, 0.280252] with:

Minimum: 0.280252 Maximum: 1.384021 Average: 0.729 seconds

Number of cars 1
bbox:car_number ((1060, 400), (1253, 498)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 498))]
Length of task list: 5
Number of processes used: 3

19%| 234/1261 [10:25<45:43, 2.67s/it]

The times for each task are: [0.591037, 1.306038, 0.989064, 0.393626, 0.386047] with:

Minimum: 0.386047 Maximum: 1.306038 Average: 0.7332 seconds

Number of cars 1
bbox:car_number ((1051, 400), (1242, 498)) : 1
The minimum distance from car: 1 is 10.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 498))]
Length of task list: 5
Number of processes used: 3

19%| 235/1261 [10:27<45:40, 2.67s/it]

The times for each task are: [0.621715, 0.851569, 1.438524, 0.304724, 0.541862] with:

Minimum: 0.304724 Maximum: 1.438524 Average: 0.7517 seconds

Number of cars 1
bbox:car_number ((1060, 400), (1238, 512)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 498))]
Length of task list: 5
Number of processes used: 3

19%| 236/1261 [10:30<45:39, 2.67s/it]

The times for each task are: [0.866559, 0.679753, 1.401124, 0.407569, 0.341191] with:

Minimum: 0.341191 Maximum: 1.401124 Average: 0.7392 seconds

Number of cars 1

bbox:car_number ((1067, 400), (1242, 512)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 512))]

Length of task list: 5

Number of processes used: 3

19%| 237/1261 [10:33<45:36, 2.67s/it]

The times for each task are: [0.682869, 0.928344, 1.482591, 0.33715, 0.46686] with:

Minimum: 0.33715 Maximum: 1.482591 Average: 0.7796 seconds

Number of cars 1

bbox:car_number ((1067, 400), (1253, 512)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 512))]

Length of task list: 5

Number of processes used: 3

19%| 238/1261 [10:36<45:34, 2.67s/it]

The times for each task are: [0.605667, 0.917279, 1.380513, 0.435118, 0.336968] with:

Minimum: 0.336968 Maximum: 1.380513 Average: 0.7351 seconds

Number of cars 1

bbox:car_number ((1067, 400), (1257, 512)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 512))]

Length of task list: 5

Number of processes used: 3

19%| 239/1261 [10:39<45:32, 2.67s/it]

The times for each task are: [0.610446, 0.38988, 0.992197, 1.445436, 0.288803] with:

Minimum: 0.288803 Maximum: 1.445436 Average: 0.7454 seconds

Number of cars 1

bbox:car_number ((1051, 400), (1268, 518)) : 1

The minimum distance from car: 1 is 4.242640687119285

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 518))]

Length of task list: 5

Number of processes used: 3

19% | 240/1261 [10:41<45:30, 2.67s/it]

The times for each task are: [1.418842, 0.84635, 0.71338, 0.41197, 0.408002] with:

Minimum: 0.408002 Maximum: 1.418842 Average: 0.7597 seconds

Number of cars 1

bbox:car_number ((1051, 400), (1257, 518)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 518))]

Length of task list: 5

Number of processes used: 3

19% | 241/1261 [10:44<45:28, 2.67s/it]

The times for each task are: [0.634391, 0.901339, 1.364763, 0.448585, 0.375753] with:

Minimum: 0.375753 Maximum: 1.364763 Average: 0.745 seconds

Number of cars 1

bbox:car_number ((1051, 400), (1253, 518)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 1

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 518))]

Length of task list: 5

Number of processes used: 3

19% | 242/1261 [10:47<45:26, 2.68s/it]

The times for each task are: [0.882201, 0.677295, 1.632619, 0.37967, 0.334126] with:

Minimum: 0.334126 Maximum: 1.632619 Average: 0.7812 seconds

Number of cars 1
bbox:car_number ((1067, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 15.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

19% | 243/1261 [10:50<45:23, 2.68s/it]

The times for each task are: [0.648421, 1.365708, 1.05991, 0.398976, 0.295037] with:

Minimum: 0.295037 Maximum: 1.365708 Average: 0.7536 seconds

Number of cars 1
bbox:car_number ((1051, 400), (1268, 518)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

19% | 244/1261 [10:52<45:20, 2.68s/it]

The times for each task are: [1.344985, 0.926341, 0.74012, 0.396477, 0.355893] with:

Minimum: 0.355893 Maximum: 1.344985 Average: 0.7528 seconds

Number of cars 1
bbox:car_number ((1051, 400), (1253, 518)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

19% | 245/1261 [10:55<45:17, 2.68s/it]

The times for each task are: [0.980313, 1.42187, 0.567567, 0.454467, 0.332686] with:

Minimum: 0.332686 Maximum: 1.42187 Average: 0.7514 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1242, 518))]
Length of task list: 5
Number of processes used: 3
```

20% | 246/1261 [10:58<45:15, 2.68s/it]

The times for each task are: [0.952986, 1.366692, 0.695767, 0.288774, 0.447275] with:

Minimum: 0.288774 Maximum: 1.366692 Average: 0.7503 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1257, 518)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 518))]
Length of task list: 5
Number of processes used: 3
```

20% | 247/1261 [11:00<45:13, 2.68s/it]

The times for each task are: [1.376493, 0.98828, 0.628458, 0.294347, 0.377104] with:

Minimum: 0.294347 Maximum: 1.376493 Average: 0.7329 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1242, 522)) : 1
The minimum distance from car: 1 is 8.246211251235321
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1242, 522))]
Length of task list: 5
Number of processes used: 3
```

20% | 248/1261 [11:03<45:10, 2.68s/it]

The times for each task are: [0.649632, 1.13036, 1.437771, 0.350339, 0.392801] with:

Minimum: 0.350339 Maximum: 1.437771 Average: 0.7922 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1253, 522)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 522))]
Length of task list: 5
Number of processes used: 3
```

20%| 249/1261 [11:06<45:08, 2.68s/it]

The times for each task are: [0.922582, 1.363051, 0.681848, 0.286399, 0.440951] with:
Minimum: 0.286399 Maximum: 1.363051 Average: 0.739 seconds

```
Number of cars 1
bbox:car_number ((1046, 400), (1253, 529)) : 1
The minimum distance from car: 1 is 4.242640687119285
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 529))]
Length of task list: 5
Number of processes used: 3
```

20%| 250/1261 [11:09<45:06, 2.68s/it]

The times for each task are: [0.636965, 0.934104, 0.402128, 0.39827, 1.685636] with:
Minimum: 0.39827 Maximum: 1.685636 Average: 0.8114 seconds

```
Number of cars 1
bbox:car_number ((1046, 400), (1253, 518)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1253, 518))]
Length of task list: 5
Number of processes used: 3
```

20%| 251/1261 [11:11<45:03, 2.68s/it]

The times for each task are: [0.608864, 0.953839, 1.383909, 0.459001, 0.297377] with:
Minimum: 0.297377 Maximum: 1.383909 Average: 0.7406 seconds

Number of cars 0

```
Length of task list: 5
Number of processes used: 3
```

```
20%|           | 252/1261 [11:14<44:59,  2.68s/it]
```

```
The times for each task are: [0.901038, 1.406222, 0.61481, 0.288496, 0.39334] with:
```

```
Minimum: 0.288496 Maximum: 1.406222 Average: 0.7208 seconds
```

```
Number of cars 2
bbox:car_number ((1136, 419), (1200, 495)) : 1
The minimum distance from car: 1 is 19.1049731745428
bbox:car_number ((1113, 427), (1133, 485)) : 2
totalCars: 1
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1197, 478))]
Length of task list: 5
Number of processes used: 3
```

```
20%|           | 253/1261 [11:17<44:57,  2.68s/it]
```

```
The times for each task are: [1.30747, 0.940505, 0.599787, 0.449185, 0.339641] with:
```

```
Minimum: 0.339641 Maximum: 1.30747 Average: 0.7273 seconds
```

```
Number of cars 2
bbox:car_number ((1075, 427), (1133, 485)) : 1
bbox:car_number ((1144, 431), (1197, 478)) : 2
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485))]
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1197, 478))]
Length of task list: 5
Number of processes used: 3
```

```
20%|           | 254/1261 [11:20<44:56,  2.68s/it]
```

```
The times for each task are: [0.912919, 0.690979, 1.599324, 0.355504, 0.449524] with:
```

```
Minimum: 0.355504 Maximum: 1.599324 Average: 0.8017 seconds
```

```
Number of cars 2
bbox:car_number ((1051, 400), (1133, 495)) : 1
```

```
The minimum distance from car: 1 is 15.0
bbox:car_number ((1141, 415), (1211, 498)) : 2
The minimum distance from car: 1 is 6.324555320336759
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495))]
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

20%| 255/1261 [11:22<44:53, 2.68s/it]

The times for each task are: [1.35411, 0.667127, 0.897647, 0.328907, 0.464361] with:

Minimum: 0.328907 Maximum: 1.35411 Average: 0.7424 seconds

```
Number of cars 2
bbox:car_number ((1051, 400), (1133, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1141, 415), (1200, 478)) : 2
The minimum distance from car: 1 is 11.661903789690601
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

20%| 256/1261 [11:25<44:50, 2.68s/it]

The times for each task are: [0.658311, 1.427612, 1.070154, 0.279141, 0.430154] with:

Minimum: 0.279141 Maximum: 1.427612 Average: 0.7731 seconds

```
Number of cars 1
bbox:car_number ((1051, 400), (1211, 498)) : 1
The minimum distance from car: 1 is 39.05124837953327
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

20%| 257/1261 [11:28<44:48, 2.68s/it]

The times for each task are: [0.865288, 0.631413, 1.389647, 0.29948, 0.476937] with:

Minimum: 0.29948 Maximum: 1.389647 Average: 0.7326 seconds

Number of cars 2

bbox:car_number ((1065, 400), (1212, 498)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1225, 431), (1242, 477)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

20% | 258/1261 [11:30<44:45, 2.68s/it]

The times for each task are: [0.997216, 1.357182, 0.632232, 0.399234, 0.289349] with:

Minimum: 0.289349 Maximum: 1.357182 Average: 0.735 seconds

Number of cars 1

bbox:car_number ((1065, 400), (1242, 516)) : 1

The minimum distance from car: 1 is 17.4928556845359

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

21% | 259/1261 [11:33<44:44, 2.68s/it]

The times for each task are: [0.585706, 1.032113, 0.436505, 1.677513, 0.433579] with:

Minimum: 0.433579 Maximum: 1.677513 Average: 0.8331 seconds

Number of cars 1

bbox:car_number ((1048, 400), (1242, 512)) : 1

The minimum distance from car: 1 is 8.246211251235321

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

21% | 260/1261 [11:36<44:41, 2.68s/it]

The times for each task are: [0.713426, 1.441609, 1.021657, 0.452545, 0.416026] with:

Minimum: 0.416026 Maximum: 1.441609 Average: 0.8091 seconds

Number of cars 1

bbox:car_number ((1048, 400), (1242, 512)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

21% | 261/1261 [11:39<44:38, 2.68s/it]

The times for each task are: [0.992166, 0.661251, 1.801713, 0.487034, 0.356438] with:

Minimum: 0.356438 Maximum: 1.801713 Average: 0.8597 seconds

Number of cars 1

bbox:car_number ((1048, 400), (1219, 498)) : 1

The minimum distance from car: 1 is 13.892443989449804

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

21% | 262/1261 [11:41<44:35, 2.68s/it]

The times for each task are: [0.920535, 0.580371, 1.387977, 0.288429, 0.382087] with:

Minimum: 0.288429 Maximum: 1.387977 Average: 0.7119 seconds

Number of cars 1

bbox:car_number ((1051, 400), (1268, 518)) : 1

The minimum distance from car: 1 is 17.029386365926403

totalCars: 2

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

21% | 263/1261 [11:44<44:33, 2.68s/it]

The times for each task are: [0.714424, 0.983826, 1.382477, 0.334186, 0.445623] with:

Minimum: 0.334186 Maximum: 1.382477 Average: 0.7721 seconds

Number of cars 1
bbox:car_number ((1046, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 13.038404810405298
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21%| 264/1261 [11:47<44:30, 2.68s/it]

The times for each task are: [0.915597, 0.582139, 1.410404, 0.47726, 0.391834] with:

Minimum: 0.391834 Maximum: 1.410404 Average: 0.7554 seconds

Number of cars 2
bbox:car_number ((1044, 400), (1238, 518)) : 1
The minimum distance from car: 1 is 4.242640687119285
bbox:car_number ((1021, 431), (1032, 462)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21%| 265/1261 [11:49<44:27, 2.68s/it]

The times for each task are: [0.621166, 1.397986, 0.870203, 0.429756, 0.32647] with:

Minimum: 0.32647 Maximum: 1.397986 Average: 0.7291 seconds

Number of cars 1
bbox:car_number ((1044, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21%| 266/1261 [11:52<44:25, 2.68s/it]

The times for each task are: [0.619873, 1.463342, 1.023785, 0.327962, 0.474189] with:

Minimum: 0.327962 Maximum: 1.463342 Average: 0.7818 seconds

Number of cars 1
bbox:car_number ((1044, 400), (1219, 512)) : 1
The minimum distance from car: 1 is 11.40175425099138
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21% | 267/1261 [11:55<44:22, 2.68s/it]

The times for each task are: [0.886158, 1.400538, 0.621299, 0.292114, 0.385604] with:

Minimum: 0.292114 Maximum: 1.400538 Average: 0.7171 seconds

Number of cars 1
bbox:car_number ((1044, 400), (1230, 498)) : 1
The minimum distance from car: 1 is 9.219544457292887
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21% | 268/1261 [11:57<44:20, 2.68s/it]

The times for each task are: [0.733737, 0.855057, 1.399313, 0.307453, 0.481662] with:

Minimum: 0.307453 Maximum: 1.399313 Average: 0.7554 seconds

Number of cars 1
bbox:car_number ((1027, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 10.44030650891055
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

21% | 269/1261 [12:00<44:18, 2.68s/it]

The times for each task are: [0.845594, 0.719103, 0.371852, 1.296588, 0.375065] with:

Minimum: 0.371852 Maximum: 1.296588 Average: 0.7216 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1242, 522)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

21% | 270/1261 [12:03<44:16, 2.68s/it]

The times for each task are: [0.893326, 0.688875, 1.622332, 0.400431, 0.290875] with:

Minimum: 0.290875 Maximum: 1.622332 Average: 0.7792 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1242, 529)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

21% | 271/1261 [12:06<44:13, 2.68s/it]

The times for each task are: [0.86325, 0.654978, 1.432522, 0.298038, 0.386497] with:

Minimum: 0.298038 Maximum: 1.432522 Average: 0.7271 seconds

```
Number of cars 1
bbox:car_number ((1020, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22% | 272/1261 [12:09<44:11, 2.68s/it]

The times for each task are: [0.869007, 0.694152, 1.393655, 0.412932, 0.285762] with:

Minimum: 0.285762 Maximum: 1.393655 Average: 0.7311 seconds

```
Number of cars 1
bbox:car_number ((1020, 400), (1242, 522)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 273/1261 [12:11<44:08, 2.68s/it]

The times for each task are: [0.587912, 1.388323, 0.914392, 0.394754, 0.395069] with:
Minimum: 0.394754 Maximum: 1.388323 Average: 0.7361 seconds

```
Number of cars 1
bbox:car_number ((1044, 400), (1214, 512)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 274/1261 [12:14<44:06, 2.68s/it]

The times for each task are: [0.582473, 1.382207, 1.066455, 0.400575, 0.346502] with:
Minimum: 0.346502 Maximum: 1.382207 Average: 0.7556 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1253, 518)) : 1
The minimum distance from car: 1 is 8.54400374531753
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 275/1261 [12:17<44:04, 2.68s/it]

The times for each task are: [0.877645, 1.413745, 0.623208, 0.296722, 0.44633] with:
Minimum: 0.296722 Maximum: 1.413745 Average: 0.7315 seconds

Number of cars 1

```
bbox:car_number ((1021, 400), (1242, 512)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22% | 276/1261 [12:20<44:02, 2.68s/it]

The times for each task are: [0.940426, 1.439034, 0.733735, 0.347691, 0.411788] with:
Minimum: 0.347691 Maximum: 1.439034 Average: 0.7745 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22% | 277/1261 [12:22<43:59, 2.68s/it]

The times for each task are: [1.557339, 0.896338, 0.624122, 0.311396, 0.457272] with:
Minimum: 0.311396 Maximum: 1.557339 Average: 0.7693 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 522)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22% | 278/1261 [12:25<43:56, 2.68s/it]

The times for each task are: [0.869084, 0.630231, 0.395164, 1.341701, 0.293324] with:
Minimum: 0.293324 Maximum: 1.341701 Average: 0.7059 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 522)) : 1
```

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

22% | 279/1261 [12:28<43:54, 2.68s/it]

The times for each task are: [0.624466, 0.974801, 1.427723, 0.383137, 0.330002] with:

Minimum: 0.330002 Maximum: 1.427723 Average: 0.748 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 516)) : 1

The minimum distance from car: 1 is 14.317821063276353

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

22% | 280/1261 [12:31<43:52, 2.68s/it]

The times for each task are: [1.037101, 0.651438, 1.477605, 0.44559, 0.371621] with:

Minimum: 0.371621 Maximum: 1.477605 Average: 0.7967 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 516)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

22% | 281/1261 [12:33<43:49, 2.68s/it]

The times for each task are: [1.390726, 0.71755, 0.977014, 0.391847, 0.343153] with:

Minimum: 0.343153 Maximum: 1.390726 Average: 0.7641 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1230, 518)) : 1

The minimum distance from car: 1 is 8.06225774829855

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 282/1261 [12:36<43:46, 2.68s/it]

The times for each task are: [1.026598, 0.63064, 1.452651, 0.30614, 0.426201] with:

Minimum: 0.30614 Maximum: 1.452651 Average: 0.7684 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

22%| 283/1261 [12:39<43:43, 2.68s/it]

The times for each task are: [0.899314, 1.385196, 0.733684, 0.300066, 0.377936] with:

Minimum: 0.300066 Maximum: 1.385196 Average: 0.7392 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 516)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

23%| 284/1261 [12:41<43:41, 2.68s/it]

The times for each task are: [0.903605, 0.60028, 1.434569, 0.391394, 0.301582] with:

Minimum: 0.301582 Maximum: 1.434569 Average: 0.7263 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1211, 518)) : 1
Length of task list: 5
Number of processes used: 3
```

23% | 285/1261 [12:44<43:38, 2.68s/it]

The times for each task are: [0.610382, 0.894033, 0.383322, 1.568049, 0.289983] with:

Minimum: 0.289983 Maximum: 1.568049 Average: 0.7492 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 15.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1211, 518)) : 1
Length of task list: 5
Number of processes used: 3
```

23% | 286/1261 [12:47<43:35, 2.68s/it]

The times for each task are: [0.623758, 1.39325, 0.878152, 0.490853, 0.300004] with:

Minimum: 0.300004 Maximum: 1.39325 Average: 0.7372 seconds

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 512)) : 1
The minimum distance from car: 1 is 15.297058540778355
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1241, 512)) : 1
Length of task list: 5
Number of processes used: 3
```

23% | 287/1261 [12:49<43:32, 2.68s/it]

The times for each task are: [0.681122, 1.393142, 0.988065, 0.57131, 0.294108] with:

Minimum: 0.294108 Maximum: 1.393142 Average: 0.7855 seconds

```
Number of cars 1
bbox:car_number ((1020, 400), (1242, 518)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1242, 518)) : 1
```

```
Length of task list: 5
Number of processes used: 3
```

```
23%|      | 288/1261 [12:52<43:29, 2.68s/it]
```

```
The times for each task are: [0.612314, 1.523343, 1.055799, 0.470737, 0.399585] with:
```

```
Minimum: 0.399585 Maximum: 1.523343 Average: 0.8124 seconds
```

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
23%|      | 289/1261 [12:55<43:27, 2.68s/it]
```

```
The times for each task are: [1.565636, 0.660299, 1.125639, 0.349636, 0.468365] with:
```

```
Minimum: 0.349636 Maximum: 1.565636 Average: 0.8339 seconds
```

```
Number of cars 1
bbox:car_number ((1021, 400), (1241, 518)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
23%|      | 290/1261 [12:57<43:24, 2.68s/it]
```

```
The times for each task are: [0.959872, 1.382407, 0.645105, 0.300758, 0.515725] with:
```

```
Minimum: 0.300758 Maximum: 1.382407 Average: 0.7608 seconds
```

```
Number of cars 1
bbox:car_number ((1021, 400), (1214, 518)) : 1
The minimum distance from car: 1 is 14.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

23%| 291/1261 [13:00<43:21, 2.68s/it]

The times for each task are: [1.361848, 1.002734, 0.642797, 0.41605, 0.493442] with:

Minimum: 0.41605 Maximum: 1.361848 Average: 0.7834 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 518)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

23%| 292/1261 [13:03<43:18, 2.68s/it]

The times for each task are: [0.887038, 0.724174, 1.38304, 0.414737, 0.302514] with:

Minimum: 0.302514 Maximum: 1.38304 Average: 0.7423 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 522)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

23%| 293/1261 [13:05<43:16, 2.68s/it]

The times for each task are: [0.585243, 0.907119, 1.376967, 0.413164, 0.323359] with:

Minimum: 0.323359 Maximum: 1.376967 Average: 0.7212 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 518)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

23%| 294/1261 [13:08<43:13, 2.68s/it]

The times for each task are: [0.884728, 0.661427, 1.603065, 0.41469, 0.299708] with:

Minimum: 0.299708 Maximum: 1.603065 Average: 0.7727 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1214, 529)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

23%| 295/1261 [13:11<43:10, 2.68s/it]

The times for each task are: [0.651568, 0.899386, 1.452859, 0.310267, 0.475627] with:

Minimum: 0.310267 Maximum: 1.452859 Average: 0.7579 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1211, 522)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

23%| 296/1261 [13:13<43:08, 2.68s/it]

The times for each task are: [0.666588, 0.901881, 1.433104, 0.436572, 0.307049] with:

Minimum: 0.307049 Maximum: 1.433104 Average: 0.749 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1214, 529)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

24% | 297/1261 [13:16<43:06, 2.68s/it]

The times for each task are: [0.577248, 0.857913, 1.460099, 0.286121, 0.532906] with:

Minimum: 0.286121 Maximum: 1.460099 Average: 0.7429 seconds

Number of cars 2

bbox:car_number ((1021, 400), (1214, 529)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1008, 438), (1017, 462)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 298/1261 [13:19<43:03, 2.68s/it]

The times for each task are: [0.633613, 0.95553, 1.369531, 0.286405, 0.377216] with:

Minimum: 0.286405 Maximum: 1.369531 Average: 0.7245 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 529)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 299/1261 [13:22<43:00, 2.68s/it]

The times for each task are: [0.837745, 0.686299, 1.405006, 0.308365, 0.466449] with:

Minimum: 0.308365 Maximum: 1.405006 Average: 0.7408 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1211, 522)) : 1

The minimum distance from car: 1 is 3.1622776601683795

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24%| 300/1261 [13:24<42:57, 2.68s/it]

The times for each task are: [0.926496, 1.439871, 0.706378, 0.406793, 0.299928] with:

Minimum: 0.299928 Maximum: 1.439871 Average: 0.7559 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1211, 522)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24%| 301/1261 [13:27<42:54, 2.68s/it]

The times for each task are: [0.670815, 1.004618, 1.455197, 0.311511, 0.452507] with:

Minimum: 0.311511 Maximum: 1.455197 Average: 0.7789 seconds

Number of cars 1

bbox:car_number ((1021, 400), (1214, 512)) : 1

The minimum distance from car: 1 is 5.0990195135927845

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24%| 302/1261 [13:29<42:51, 2.68s/it]

The times for each task are: [0.605773, 1.076569, 1.403773, 0.38979, 0.339218] with:

Minimum: 0.339218 Maximum: 1.403773 Average: 0.763 seconds

Number of cars 1

bbox:car_number ((1020, 400), (1211, 516)) : 1

The minimum distance from car: 1 is 2.8284271247461903

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24%| 303/1261 [13:32<42:49, 2.68s/it]

The times for each task are: [0.922088, 0.690087, 1.750233, 0.458255, 0.402156] with:

Minimum: 0.402156 Maximum: 1.750233 Average: 0.8446 seconds

Number of cars 1

bbox:car_number ((1020, 400), (1214, 518)) : 1

The minimum distance from car: 1 is 2.23606797749979

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 304/1261 [13:35<42:46, 2.68s/it]

The times for each task are: [0.979836, 1.731242, 0.707368, 0.401978, 0.367645] with:

Minimum: 0.367645 Maximum: 1.731242 Average: 0.8376 seconds

Number of cars 1

bbox:car_number ((1015, 400), (1214, 518)) : 1

The minimum distance from car: 1 is 3.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 305/1261 [13:38<42:44, 2.68s/it]

The times for each task are: [0.993189, 1.523489, 0.668818, 0.449167, 0.387591] with:

Minimum: 0.387591 Maximum: 1.523489 Average: 0.8045 seconds

Number of cars 1

bbox:car_number ((1008, 400), (1208, 518)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

24% | 306/1261 [13:41<42:42, 2.68s/it]

The times for each task are: [0.659865, 1.04798, 1.409796, 0.494679, 0.336073] with:

Minimum: 0.336073 Maximum: 1.409796 Average: 0.7897 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1212, 516)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

24% | 307/1261 [13:43<42:39, 2.68s/it]

The times for each task are: [0.605766, 0.989424, 1.482424, 0.324558, 0.526528] with:

Minimum: 0.324558 Maximum: 1.482424 Average: 0.7857 seconds

Number of cars 1
bbox:car_number ((994, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 14.035668847618199
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

24% | 308/1261 [13:46<42:37, 2.68s/it]

The times for each task are: [0.630901, 0.944071, 1.617187, 0.360132, 0.435005] with:

Minimum: 0.360132 Maximum: 1.617187 Average: 0.7975 seconds

Number of cars 2
bbox:car_number ((1021, 400), (1208, 512)) : 1
The minimum distance from car: 1 is 12.36931687685298
bbox:car_number ((994, 430), (1017, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25% | 309/1261 [13:49<42:35, 2.68s/it]

The times for each task are: [0.609743, 1.045016, 1.40896, 0.467036, 0.30028] with:

Minimum: 0.30028 Maximum: 1.40896 Average: 0.7662 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (970, 400))]
Length of task list: 5
Number of processes used: 3

25%| 310/1261 [13:51<42:32, 2.68s/it]

The times for each task are: [0.586519, 1.383095, 1.050244, 0.401433, 0.292903] with:

Minimum: 0.292903 Maximum: 1.383095 Average: 0.7428 seconds

Number of cars 1
bbox:car_number ((970, 400), (1212, 518)) : 1
The minimum distance from car: 1 is 25.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (970, 400))]
Length of task list: 5
Number of processes used: 3

25%| 311/1261 [13:54<42:29, 2.68s/it]

The times for each task are: [0.915326, 0.616574, 1.38009, 0.429136, 0.377548] with:

Minimum: 0.377548 Maximum: 1.38009 Average: 0.7437 seconds

Number of cars 2
bbox:car_number ((1027, 400), (1211, 512)) : 1
The minimum distance from car: 1 is 28.160255680657446
bbox:car_number ((994, 427), (1010, 462)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (970, 400))]
Length of task list: 5
Number of processes used: 3

25%| 312/1261 [13:57<42:26, 2.68s/it]

The times for each task are: [0.859997, 0.696639, 1.602044, 0.386311, 0.37475] with:

Minimum: 0.37475 Maximum: 1.602044 Average: 0.7839 seconds

Number of cars 2
bbox:car_number ((1021, 400), (1025, 472)) : 1
bbox:car_number ((1027, 400), (1211, 518)) : 2
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1214, 518))]
Length of task list: 5
Number of processes used: 3

25% | 313/1261 [13:59<42:23, 2.68s/it]

The times for each task are: [0.625813, 1.416319, 0.982288, 0.40239, 0.290201] with:

Minimum: 0.290201 Maximum: 1.416319 Average: 0.7434 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1214, 518)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1214, 518))]
Length of task list: 5
Number of processes used: 3

25% | 314/1261 [14:02<42:20, 2.68s/it]

The times for each task are: [0.890159, 0.619196, 1.580256, 0.410158, 0.371392] with:

Minimum: 0.371392 Maximum: 1.580256 Average: 0.7742 seconds

Number of cars 2
bbox:car_number ((1021, 400), (1211, 529)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1008, 445), (1017, 492)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1211, 529))]
Length of task list: 5
Number of processes used: 3

25% | 315/1261 [14:04<42:16, 2.68s/it]

The times for each task are: [1.355176, 0.938143, 0.611183, 0.410254, 0.313911] with:

Minimum: 0.313911 Maximum: 1.355176 Average: 0.7257 seconds

Number of cars 1
bbox:car_number ((1021, 400), (1208, 522)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25%| 316/1261 [14:07<42:13, 2.68s/it]

The times for each task are: [0.90578, 0.643508, 1.616095, 0.294891, 0.432405] with:

Minimum: 0.294891 Maximum: 1.616095 Average: 0.7785 seconds

Number of cars 1
bbox:car_number ((1027, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25%| 317/1261 [14:09<42:09, 2.68s/it]

The times for each task are: [0.576196, 1.431069, 0.86449, 0.391045, 0.282201] with:

Minimum: 0.282201 Maximum: 1.431069 Average: 0.709 seconds

Number of cars 1
bbox:car_number ((1008, 400), (1208, 512)) : 1
The minimum distance from car: 1 is 11.40175425099138
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

25%| 318/1261 [14:11<42:06, 2.68s/it]

The times for each task are: [1.070977, 0.707169, 1.344638, 0.304341, 0.455696] with:

Minimum: 0.304341 Maximum: 1.344638 Average: 0.7766 seconds

```
Number of cars 2
bbox:car_number ((1021, 400), (1208, 518)) : 1
The minimum distance from car: 1 is 6.708203932499369
bbox:car_number ((1008, 438), (1010, 478)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

25%| 319/1261 [14:14<42:03, 2.68s/it]

The times for each task are: [0.9573, 1.424441, 0.69826, 0.315886, 0.453746] with:

Minimum: 0.315886 Maximum: 1.424441 Average: 0.7699 seconds

```
Number of cars 1
bbox:car_number ((952, 400), (1187, 518)) : 1
The minimum distance from car: 1 is 45.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

25%| 320/1261 [14:17<42:00, 2.68s/it]

The times for each task are: [0.910341, 0.630302, 1.421023, 0.303957, 0.443327] with:

Minimum: 0.303957 Maximum: 1.421023 Average: 0.7418 seconds

```
Number of cars 2
bbox:car_number ((989, 400), (1187, 522)) : 1
The minimum distance from car: 1 is 19.1049731745428
bbox:car_number ((1195, 445), (1211, 492)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

25%| 321/1261 [14:19<41:57, 2.68s/it]

The times for each task are: [1.319946, 0.62908, 1.02718, 0.294875, 0.434701] with:

Minimum: 0.294875 Maximum: 1.319946 Average: 0.7412 seconds

Number of cars 2
bbox:car_number ((989, 400), (1187, 512)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1195, 445), (1211, 492)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26% | 322/1261 [14:22<41:54, 2.68s/it]

The times for each task are: [0.888558, 0.669579, 1.513432, 0.315674, 0.498784] with:

Minimum: 0.315674 Maximum: 1.513432 Average: 0.7772 seconds

Number of cars 1
bbox:car_number ((989, 400), (1197, 507)) : 1
The minimum distance from car: 1 is 5.830951894845301
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26% | 323/1261 [14:24<41:51, 2.68s/it]

The times for each task are: [0.853161, 0.635946, 0.414811, 0.314614, 1.681059] with:

Minimum: 0.314614 Maximum: 1.681059 Average: 0.7799 seconds

Number of cars 2
bbox:car_number ((994, 400), (1187, 516)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1195, 445), (1200, 492)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26% | 324/1261 [14:27<41:49, 2.68s/it]

The times for each task are: [0.86876, 1.472125, 0.708381, 0.338522, 0.383028] with:

Minimum: 0.338522 Maximum: 1.472125 Average: 0.7542 seconds

Number of cars 1
bbox:car_number ((998, 400), (1200, 512)) : 1
The minimum distance from car: 1 is 9.219544457292887
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26%| 325/1261 [14:30<41:46, 2.68s/it]

The times for each task are: [0.621854, 0.881862, 0.404589, 1.472628, 0.305379] with:

Minimum: 0.305379 Maximum: 1.472628 Average: 0.7373 seconds

Number of cars 1
bbox:car_number ((998, 400), (1187, 518)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26%| 326/1261 [14:33<41:44, 2.68s/it]

The times for each task are: [0.877908, 0.565249, 1.628723, 0.398945, 0.287561] with:

Minimum: 0.287561 Maximum: 1.628723 Average: 0.7517 seconds

Number of cars 1
bbox:car_number ((998, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 12.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

26%| 327/1261 [14:35<41:41, 2.68s/it]

The times for each task are: [0.695029, 0.952295, 1.452167, 0.44975, 0.294526] with:

Minimum: 0.294526 Maximum: 1.452167 Average: 0.7688 seconds

```
Number of cars 2
bbox:car_number ((998, 400), (1187, 512)) : 1
The minimum distance from car: 1 is 12.36931687685298
bbox:car_number ((1195, 445), (1211, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

26%| 328/1261 [14:38<41:38, 2.68s/it]

The times for each task are: [0.586633, 0.937623, 1.610322, 0.399086, 0.320993] with:

Minimum: 0.320993 Maximum: 1.610322 Average: 0.7709 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 4.242640687119285
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

26%| 329/1261 [14:41<41:36, 2.68s/it]

The times for each task are: [0.700325, 1.469572, 0.907361, 0.318445, 0.436528] with:

Minimum: 0.318445 Maximum: 1.469572 Average: 0.7664 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1187, 512)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40
Length of task list: 5
Number of processes used: 3
```

26%| 330/1261 [14:43<41:32, 2.68s/it]

The times for each task are: [0.700445, 0.859966, 1.453959, 0.290338, 0.484948] with:

Minimum: 0.290338 Maximum: 1.453959 Average: 0.7579 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1180, 518)) : 1
The minimum distance from car: 1 is 4.242640687119285
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

26%| 331/1261 [14:46<41:29, 2.68s/it]

The times for each task are: [0.626964, 1.016566, 1.432001, 0.413228, 0.315158] with:

Minimum: 0.315158 Maximum: 1.432001 Average: 0.7608 seconds

```
Number of cars 1
bbox:car_number ((998, 400), (1211, 518)) : 1
The minimum distance from car: 1 is 17.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

26%| 332/1261 [14:48<41:26, 2.68s/it]

The times for each task are: [1.415619, 0.854815, 0.60735, 0.37544, 0.434179] with:

Minimum: 0.37544 Maximum: 1.415619 Average: 0.7375 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1187, 518)) : 1
The minimum distance from car: 1 is 14.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

26%| 333/1261 [14:51<41:24, 2.68s/it]

The times for each task are: [0.638469, 1.351562, 1.014748, 0.290467, 0.489138] with:

Minimum: 0.290467 Maximum: 1.351562 Average: 0.7569 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1181, 518)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

26%| 334/1261 [14:54<41:21, 2.68s/it]

The times for each task are: [0.624714, 0.905141, 0.436708, 0.299896, 1.478368] with:
Minimum: 0.299896 Maximum: 1.478368 Average: 0.749 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1182, 512)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27%| 335/1261 [14:56<41:18, 2.68s/it]

The times for each task are: [0.675404, 0.816233, 1.379247, 0.28902, 0.439068] with:
Minimum: 0.28902 Maximum: 1.379247 Average: 0.7198 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27%| 336/1261 [14:59<41:16, 2.68s/it]

The times for each task are: [0.618037, 1.036444, 1.432003, 0.301234, 0.388139] with:
Minimum: 0.301234 Maximum: 1.432003 Average: 0.7552 seconds

Number of cars 1

```
bbox:car_number ((994, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27% | 337/1261 [15:02<41:14, 2.68s/it]

The times for each task are: [0.710949, 1.399628, 0.920803, 0.299953, 0.44933] with:

Minimum: 0.299953 Maximum: 1.399628 Average: 0.7561 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1182, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27% | 338/1261 [15:05<41:11, 2.68s/it]

The times for each task are: [0.607138, 0.885795, 1.384479, 0.294383, 0.389713] with:

Minimum: 0.294383 Maximum: 1.384479 Average: 0.7123 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27% | 339/1261 [15:07<41:09, 2.68s/it]

The times for each task are: [0.697841, 0.822954, 1.328589, 0.296619, 0.520455] with:

Minimum: 0.296619 Maximum: 1.328589 Average: 0.7333 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1212, 518)) : 1
```

The minimum distance from car: 1 is 14.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

27% | 340/1261 [15:10<41:06, 2.68s/it]

The times for each task are: [0.736847, 1.476256, 0.947845, 0.495899, 0.357464] with:

Minimum: 0.357464 Maximum: 1.476256 Average: 0.8029 seconds

Number of cars 1

bbox:car_number ((994, 400), (1180, 518)) : 1

The minimum distance from car: 1 is 16.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

27% | 341/1261 [15:13<41:03, 2.68s/it]

The times for each task are: [1.003299, 0.605359, 1.371953, 0.466933, 0.302452] with:

Minimum: 0.302452 Maximum: 1.371953 Average: 0.75 seconds

Number of cars 2

bbox:car_number ((994, 400), (1162, 512)) : 1

The minimum distance from car: 1 is 9.486832980505138

bbox:car_number ((1165, 431), (1180, 477)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

27% | 342/1261 [15:15<41:00, 2.68s/it]

The times for each task are: [0.849125, 0.707427, 1.496014, 0.378551, 0.44086] with:

Minimum: 0.378551 Maximum: 1.496014 Average: 0.7744 seconds

Number of cars 3

bbox:car_number ((994, 400), (1162, 518)) : 1

```
The minimum distance from car: 1 is 3.0
bbox:car_number ((1165, 431), (1180, 478)) : 2
bbox:car_number ((1183, 445), (1212, 477)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27% | 343/1261 [15:18<40:58, 2.68s/it]

The times for each task are: [0.99583, 0.693826, 1.419013, 0.501801, 0.395994] with:

Minimum: 0.395994 Maximum: 1.419013 Average: 0.8013 seconds

```
Number of cars 1
bbox:car_number ((994, 400), (1230, 518)) : 1
The minimum distance from car: 1 is 34.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27% | 344/1261 [15:21<40:56, 2.68s/it]

The times for each task are: [0.874778, 0.671036, 1.512423, 0.384734, 0.387772] with:

Minimum: 0.384734 Maximum: 1.512423 Average: 0.7661 seconds

```
Number of cars 2
bbox:car_number ((994, 400), (1182, 512)) : 1
The minimum distance from car: 1 is 24.186773244895647
bbox:car_number ((1204, 419), (1219, 459)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

27% | 345/1261 [15:24<40:53, 2.68s/it]

The times for each task are: [0.633432, 1.037895, 0.404752, 1.546603, 0.35595] with:

Minimum: 0.35595 Maximum: 1.546603 Average: 0.7957 seconds

```
Number of cars 2
bbox:car_number ((994, 400), (1160, 512)) : 1
The minimum distance from car: 1 is 11.0
bbox:car_number ((1165, 431), (1180, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1165, 431))]
Length of task list: 5
Number of processes used: 3
```

27% | 346/1261 [15:26<40:50, 2.68s/it]

The times for each task are: [0.947722, 0.606832, 1.366133, 0.304629, 0.411221] with:

Minimum: 0.304629 Maximum: 1.366133 Average: 0.7273 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1180, 512)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1165, 431))]
Length of task list: 5
Number of processes used: 3
```

28% | 347/1261 [15:29<40:47, 2.68s/it]

The times for each task are: [1.39028, 0.899955, 0.725002, 0.288811, 0.387738] with:

Minimum: 0.288811 Maximum: 1.39028 Average: 0.7384 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1167, 507)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1165, 431))]
Length of task list: 5
Number of processes used: 3
```

28% | 348/1261 [15:31<40:45, 2.68s/it]

The times for each task are: [1.021417, 0.620576, 0.305959, 1.676275, 0.448331] with:

Minimum: 0.305959 Maximum: 1.676275 Average: 0.8145 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1167, 507)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28%| 349/1261 [15:34<40:41, 2.68s/it]

The times for each task are: [0.906111, 1.4308, 0.705186, 0.304434, 0.421324] with:

Minimum: 0.304434 Maximum: 1.4308 Average: 0.7536 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1162, 507)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28%| 350/1261 [15:37<40:39, 2.68s/it]

The times for each task are: [1.450815, 1.112971, 0.580686, 0.418878, 0.349257] with:

Minimum: 0.349257 Maximum: 1.450815 Average: 0.7825 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1160, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28%| 351/1261 [15:39<40:36, 2.68s/it]

The times for each task are: [0.910688, 0.608494, 1.468089, 0.427764, 0.340091] with:

Minimum: 0.340091 Maximum: 1.468089 Average: 0.751 seconds

Number of cars 1

```
bbox:car_number ((989, 400), (1162, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28% | 352/1261 [15:42<40:33, 2.68s/it]

The times for each task are: [0.660369, 0.886854, 1.375484, 0.414448, 0.284423] with:

Minimum: 0.284423 Maximum: 1.375484 Average: 0.7243 seconds

```
Number of cars 1
bbox:car_number ((989, 400), (1160, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

28% | 353/1261 [15:45<40:31, 2.68s/it]

The times for each task are: [0.878504, 0.66063, 0.399441, 0.3061, 1.379486] with:

Minimum: 0.3061 Maximum: 1.379486 Average: 0.7248 seconds

```
Number of cars 3
bbox:car_number ((989, 400), (1133, 498)) : 1
The minimum distance from car: 1 is 13.0
bbox:car_number ((1158, 400), (1230, 485)) : 2
The minimum distance from car: 1 is 38.91015291668744
bbox:car_number ((1144, 427), (1149, 485)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

28% | 354/1261 [15:48<40:28, 2.68s/it]

The times for each task are: [0.885357, 0.631752, 1.441239, 0.331323, 0.452817] with:

Minimum: 0.331323 Maximum: 1.441239 Average: 0.7485 seconds

Number of cars 2
bbox:car_number ((989, 400), (1149, 507)) : 1
The minimum distance from car: 1 is 8.94427190999916
bbox:car_number ((985, 430), (987, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

28% | 355/1261 [15:50<40:26, 2.68s/it]

The times for each task are: [0.857209, 0.75924, 1.379817, 0.345716, 0.390306] with:

Minimum: 0.345716 Maximum: 1.379817 Average: 0.7465 seconds

Number of cars 3
bbox:car_number ((989, 400), (1143, 498)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1195, 415), (1242, 462)) : 2
bbox:car_number ((1150, 445), (1160, 477)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

28% | 356/1261 [15:53<40:23, 2.68s/it]

The times for each task are: [1.323045, 0.847049, 0.732238, 0.299068, 0.453566] with:

Minimum: 0.299068 Maximum: 1.323045 Average: 0.731 seconds

Number of cars 1
bbox:car_number ((989, 400), (1152, 507)) : 1
The minimum distance from car: 1 is 5.656854249492381
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

28% | 357/1261 [15:56<40:21, 2.68s/it]

The times for each task are: [0.882947, 1.378565, 0.715047, 0.363306, 0.40785] with:

Minimum: 0.363306 Maximum: 1.378565 Average: 0.7495 seconds

Number of cars 1

bbox:car_number ((989, 400), (1139, 507)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

28% | 358/1261 [15:58<40:18, 2.68s/it]

The times for each task are: [0.867099, 0.642313, 1.532261, 0.293524, 0.383034] with:

Minimum: 0.293524 Maximum: 1.532261 Average: 0.7436 seconds

Number of cars 1

bbox:car_number ((985, 415), (1143, 507)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

28% | 359/1261 [16:01<40:15, 2.68s/it]

The times for each task are: [0.629947, 1.650945, 1.063615, 0.443359, 0.320613] with:

Minimum: 0.320613 Maximum: 1.650945 Average: 0.8217 seconds

Number of cars 1

bbox:car_number ((975, 400), (1143, 507)) : 1

The minimum distance from car: 1 is 9.433981132056603

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

29% | 360/1261 [16:04<40:13, 2.68s/it]

The times for each task are: [0.872496, 0.683907, 0.288254, 1.676144, 0.378548] with:

Minimum: 0.288254 Maximum: 1.676144 Average: 0.7799 seconds

Number of cars 1
bbox:car_number ((985, 400), (1200, 512)) : 1
The minimum distance from car: 1 is 33.13608305156178
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

29% | 361/1261 [16:06<40:10, 2.68s/it]

The times for each task are: [0.930669, 1.484916, 0.734943, 0.313395, 0.433945] with:

Minimum: 0.313395 Maximum: 1.484916 Average: 0.7796 seconds

Number of cars 1
bbox:car_number ((975, 400), (1160, 512)) : 1
The minimum distance from car: 1 is 25.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

29% | 362/1261 [16:09<40:07, 2.68s/it]

The times for each task are: [0.91687, 0.62825, 1.647945, 0.349261, 0.397899] with:

Minimum: 0.349261 Maximum: 1.647945 Average: 0.788 seconds

Number of cars 1
bbox:car_number ((970, 400), (1143, 512)) : 1
The minimum distance from car: 1 is 11.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

29% | 363/1261 [16:12<40:04, 2.68s/it]

The times for each task are: [0.925815, 0.631587, 1.587184, 0.307394, 0.387171] with:

Minimum: 0.307394 Maximum: 1.587184 Average: 0.7678 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 512)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 364/1261 [16:14<40:01, 2.68s/it]

The times for each task are: [0.599261, 1.396475, 0.939405, 0.399043, 0.301389] with:

Minimum: 0.301389 Maximum: 1.396475 Average: 0.7271 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 365/1261 [16:17<39:59, 2.68s/it]

The times for each task are: [0.906093, 0.672306, 1.39244, 0.389568, 0.296192] with:

Minimum: 0.296192 Maximum: 1.39244 Average: 0.7313 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 512)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 366/1261 [16:20<39:57, 2.68s/it]

The times for each task are: [0.620359, 0.862313, 1.36081, 0.389428, 0.296285] with:

Minimum: 0.296285 Maximum: 1.36081 Average: 0.7058 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 518)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29%| 367/1261 [16:22<39:54, 2.68s/it]

The times for each task are: [0.617769, 1.418152, 0.900536, 0.404573, 0.301493] with:
Minimum: 0.301493 Maximum: 1.418152 Average: 0.7285 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1185, 518)) : 1
The minimum distance from car: 1 is 18.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29%| 368/1261 [16:25<39:51, 2.68s/it]

The times for each task are: [1.379523, 0.617098, 1.012831, 0.335642, 0.392576] with:
Minimum: 0.335642 Maximum: 1.379523 Average: 0.7475 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1149, 512)) : 1
The minimum distance from car: 1 is 18.24828759089466
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29%| 369/1261 [16:28<39:49, 2.68s/it]

The times for each task are: [0.831605, 1.376953, 0.70466, 0.404078, 0.308949] with:
Minimum: 0.308949 Maximum: 1.376953 Average: 0.7252 seconds

Number of cars 1

```
bbox:car_number ((975, 400), (1139, 516)) : 1
The minimum distance from car: 1 is 2.8284271247461903
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 370/1261 [16:30<39:45, 2.68s/it]

The times for each task are: [0.878797, 0.629263, 0.421013, 1.423197, 0.301998] with:
Minimum: 0.301998 Maximum: 1.423197 Average: 0.7309 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1137, 516)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

29% | 371/1261 [16:33<39:43, 2.68s/it]

The times for each task are: [0.592839, 1.074764, 1.659893, 0.390152, 0.345057] with:
Minimum: 0.345057 Maximum: 1.659893 Average: 0.8125 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1139, 512)) : 1
The minimum distance from car: 1 is 2.23606797749979
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

30% | 372/1261 [16:36<39:40, 2.68s/it]

The times for each task are: [0.895334, 0.571243, 1.593099, 0.301559, 0.396375] with:
Minimum: 0.301559 Maximum: 1.593099 Average: 0.7515 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1124, 507)) : 1
```

```
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1124, 498))]
Length of task list: 5
Number of processes used: 3
```

30% | 373/1261 [16:39<39:39, 2.68s/it]

The times for each task are: [0.584123, 0.436426, 0.958533, 1.491183, 0.389996] with:

Minimum: 0.389996 Maximum: 1.491183 Average: 0.7721 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1124, 498)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1124, 498))]
Length of task list: 5
Number of processes used: 3
```

30% | 374/1261 [16:42<39:36, 2.68s/it]

The times for each task are: [0.945391, 0.675183, 1.535599, 0.297, 0.462583] with:

Minimum: 0.297 Maximum: 1.535599 Average: 0.7832 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1122, 498)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1122, 498))]
Length of task list: 5
Number of processes used: 3
```

30% | 375/1261 [16:44<39:33, 2.68s/it]

The times for each task are: [0.961495, 1.402545, 0.672524, 0.291, 0.381385] with:

Minimum: 0.291 Maximum: 1.402545 Average: 0.7418 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1124, 507)) : 1
The minimum distance from car: 1 is 4.123105625617661
```

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1118, 507))]
Length of task list: 5
Number of processes used: 3
```

30%| 376/1261 [16:47<39:30, 2.68s/it]

The times for each task are: [0.620876, 1.010733, 1.46414, 0.385381, 0.489372] with:

Minimum: 0.385381 Maximum: 1.46414 Average: 0.7941 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1124, 498)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1118, 507))]
Length of task list: 5
Number of processes used: 3
```

30%| 377/1261 [16:49<39:27, 2.68s/it]

The times for each task are: [0.621007, 1.392487, 1.09737, 0.407451, 0.299437] with:

Minimum: 0.299437 Maximum: 1.392487 Average: 0.7636 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1122, 512)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1118, 507))]
Length of task list: 5
Number of processes used: 3
```

30%| 378/1261 [16:52<39:24, 2.68s/it]

The times for each task are: [0.928353, 1.405545, 0.737462, 0.391348, 0.360389] with:

Minimum: 0.360389 Maximum: 1.405545 Average: 0.7646 seconds

```
Number of cars 1
bbox:car_number ((970, 400), (1118, 507)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
```

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1118, 512))]
Length of task list: 5
Number of processes used: 3

30% | 379/1261 [16:54<39:21, 2.68s/it]

The times for each task are: [0.842148, 0.738835, 1.398855, 0.41565, 0.444148] with:

Minimum: 0.41565 Maximum: 1.398855 Average: 0.7679 seconds

Number of cars 1
bbox:car_number ((958, 400), (1118, 512)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1118, 512))]
Length of task list: 5
Number of processes used: 3

30% | 380/1261 [16:57<39:18, 2.68s/it]

The times for each task are: [0.663146, 1.396114, 0.875223, 0.516879, 0.304077] with:

Minimum: 0.304077 Maximum: 1.396114 Average: 0.7511 seconds

Number of cars 1
bbox:car_number ((958, 400), (1116, 498)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1116, 498))]
Length of task list: 5
Number of processes used: 3

30% | 381/1261 [16:59<39:15, 2.68s/it]

The times for each task are: [1.37928, 0.664151, 0.889774, 0.407246, 0.386442] with:

Minimum: 0.386442 Maximum: 1.37928 Average: 0.7454 seconds

Number of cars 1
bbox:car_number ((955, 400), (1116, 507)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1116, 507))]

```
Length of task list: 5
Number of processes used: 3
```

```
30%|      | 382/1261 [17:02<39:12, 2.68s/it]
```

```
The times for each task are: [0.631308, 0.966649, 0.395179, 1.522528, 0.310582] with:
```

```
Minimum: 0.310582 Maximum: 1.522528 Average: 0.7652 seconds
```

```
Number of cars 1
bbox:car_number ((955, 400), (1107, 498)) : 1
The minimum distance from car: 1 is 5.656854249492381
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
30%|      | 383/1261 [17:05<39:09, 2.68s/it]
```

```
The times for each task are: [0.613274, 0.914101, 1.384878, 0.386817, 0.395378] with:
```

```
Minimum: 0.386817 Maximum: 1.384878 Average: 0.7389 seconds
```

```
Number of cars 1
bbox:car_number ((955, 400), (1107, 507)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
30%|      | 384/1261 [17:07<39:07, 2.68s/it]
```

```
The times for each task are: [0.667762, 1.020813, 1.579493, 0.41695, 0.32168] with:
```

```
Minimum: 0.32168 Maximum: 1.579493 Average: 0.8013 seconds
```

```
Number of cars 1
bbox:car_number ((955, 400), (1107, 498)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

31%| 385/1261 [17:10<39:04, 2.68s/it]

The times for each task are: [0.890954, 0.607982, 1.509732, 0.298942, 0.400471] with:

Minimum: 0.298942 Maximum: 1.509732 Average: 0.7416 seconds

Number of cars 1

bbox:car_number ((952, 400), (1107, 498)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

31%| 386/1261 [17:13<39:01, 2.68s/it]

The times for each task are: [0.591397, 0.959532, 0.349228, 1.599135, 0.485471] with:

Minimum: 0.349228 Maximum: 1.599135 Average: 0.797 seconds

Number of cars 1

bbox:car_number ((952, 400), (1106, 507)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

31%| 387/1261 [17:15<38:58, 2.68s/it]

The times for each task are: [0.626871, 0.867427, 0.378562, 0.345013, 1.592055] with:

Minimum: 0.345013 Maximum: 1.592055 Average: 0.762 seconds

Number of cars 1

bbox:car_number ((951, 400), (1106, 497)) : 1

The minimum distance from car: 1 is 5.0990195135927845

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

31% | 388/1261 [17:17<38:55, 2.68s/it]

The times for each task are: [1.304934, 0.885535, 0.729391, 0.302745, 0.406752] with:

Minimum: 0.302745 Maximum: 1.304934 Average: 0.7259 seconds

Number of cars 1

bbox:car_number ((955, 400), (1107, 498)) : 1

The minimum distance from car: 1 is 3.1622776601683795

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

31% | 389/1261 [17:20<38:52, 2.68s/it]

The times for each task are: [1.399216, 1.093238, 0.632133, 0.378942, 0.343221] with:

Minimum: 0.343221 Maximum: 1.399216 Average: 0.7693 seconds

Number of cars 1

bbox:car_number ((955, 400), (1106, 498)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

31% | 390/1261 [17:23<38:50, 2.68s/it]

The times for each task are: [0.541777, 1.002636, 1.323583, 0.400026, 0.402995] with:

Minimum: 0.400026 Maximum: 1.323583 Average: 0.7342 seconds

Number of cars 1

bbox:car_number ((958, 400), (1105, 498)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

31% | 391/1261 [17:26<38:48, 2.68s/it]

The times for each task are: [0.568323, 1.054453, 0.436194, 1.534456, 0.337403] with:

Minimum: 0.337403 Maximum: 1.534456 Average: 0.7862 seconds

Number of cars 1

bbox:car_number ((958, 400), (1106, 497)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 392/1261 [17:29<38:45, 2.68s/it]

The times for each task are: [0.665411, 0.983083, 1.53305, 0.466359, 0.355829] with:

Minimum: 0.355829 Maximum: 1.53305 Average: 0.8007 seconds

Number of cars 1

bbox:car_number ((951, 400), (1079, 495)) : 1

The minimum distance from car: 1 is 17.029386365926403

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 393/1261 [17:31<38:42, 2.68s/it]

The times for each task are: [0.872459, 0.654592, 1.584756, 0.40477, 0.480582] with:

Minimum: 0.40477 Maximum: 1.584756 Average: 0.7994 seconds

Number of cars 1

bbox:car_number ((940, 400), (1092, 498)) : 1

The minimum distance from car: 1 is 2.23606797749979

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 394/1261 [17:34<38:39, 2.68s/it]

The times for each task are: [0.905876, 1.372969, 0.674152, 0.442749, 0.347405] with:

Minimum: 0.347405 Maximum: 1.372969 Average: 0.7486 seconds

Number of cars 1

bbox:car_number ((940, 400), (1092, 497)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 395/1261 [17:36<38:36, 2.67s/it]

The times for each task are: [0.860221, 0.652798, 1.403883, 0.430452, 0.386264] with:

Minimum: 0.386264 Maximum: 1.403883 Average: 0.7467 seconds

Number of cars 1

bbox:car_number ((932, 400), (1087, 495)) : 1

The minimum distance from car: 1 is 7.0710678118654755

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 396/1261 [17:39<38:33, 2.68s/it]

The times for each task are: [0.574884, 1.398469, 0.894616, 0.355999, 0.4298] with:

Minimum: 0.355999 Maximum: 1.398469 Average: 0.7308 seconds

Number of cars 1

bbox:car_number ((932, 400), (1092, 497)) : 1

The minimum distance from car: 1 is 3.1622776601683795

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

31% | 397/1261 [17:41<38:30, 2.67s/it]

The times for each task are: [1.0178, 0.618806, 0.40432, 1.585112, 0.396715] with:

Minimum: 0.396715 Maximum: 1.585112 Average: 0.8046 seconds

Number of cars 1
bbox:car_number ((932, 400), (1087, 495)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

32% | 398/1261 [17:44<38:28, 2.68s/it]

The times for each task are: [0.650944, 0.965608, 1.578642, 0.391619, 0.353982] with:

Minimum: 0.353982 Maximum: 1.578642 Average: 0.7882 seconds

Number of cars 1
bbox:car_number ((932, 400), (1086, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

32% | 399/1261 [17:47<38:26, 2.68s/it]

The times for each task are: [0.950249, 0.61633, 1.628225, 0.295894, 0.402135] with:

Minimum: 0.295894 Maximum: 1.628225 Average: 0.7786 seconds

Number of cars 1
bbox:car_number ((940, 400), (1086, 497)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

32% | 400/1261 [17:49<38:22, 2.67s/it]

The times for each task are: [0.579078, 1.394814, 1.06354, 0.394205, 0.331142] with:

Minimum: 0.331142 Maximum: 1.394814 Average: 0.7526 seconds

```
Number of cars 1
bbox:car_number ((932, 400), (1079, 497)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 401/1261 [17:52<38:19, 2.67s/it]

The times for each task are: [0.637537, 0.889874, 1.518673, 0.291828, 0.389354] with:

Minimum: 0.291828 Maximum: 1.518673 Average: 0.7455 seconds

```
Number of cars 1
bbox:car_number ((932, 400), (1077, 497)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 402/1261 [17:55<38:17, 2.67s/it]

The times for each task are: [1.415413, 0.660343, 0.925524, 0.368031, 0.483216] with:

Minimum: 0.368031 Maximum: 1.415413 Average: 0.7705 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1079, 495)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 403/1261 [17:57<38:14, 2.67s/it]

The times for each task are: [0.832729, 0.691221, 1.379835, 0.408444, 0.497391] with:

Minimum: 0.408444 Maximum: 1.379835 Average: 0.7619 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1086, 498)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32%| 404/1261 [18:00<38:12, 2.67s/it]

The times for each task are: [0.864485, 1.321476, 0.656948, 0.434346, 0.300536] with:
Minimum: 0.300536 Maximum: 1.321476 Average: 0.7156 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1077, 495)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32%| 405/1261 [18:03<38:09, 2.67s/it]

The times for each task are: [0.865131, 0.622691, 1.3718, 0.401583, 0.305346] with:
Minimum: 0.305346 Maximum: 1.3718 Average: 0.7133 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1079, 498)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32%| 406/1261 [18:05<38:06, 2.67s/it]

The times for each task are: [1.402719, 1.073984, 0.614635, 0.346755, 0.404867] with:
Minimum: 0.346755 Maximum: 1.402719 Average: 0.7686 seconds

Number of cars 1

```
bbox:car_number ((958, 400), (1077, 498)) : 1
The minimum distance from car: 1 is 13.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 407/1261 [18:08<38:04, 2.67s/it]

The times for each task are: [0.660894, 0.870438, 1.591244, 0.310207, 0.37301] with:

Minimum: 0.310207 Maximum: 1.591244 Average: 0.7612 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1079, 498)) : 1
The minimum distance from car: 1 is 13.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 408/1261 [18:11<38:01, 2.67s/it]

The times for each task are: [0.88523, 1.452028, 0.713345, 0.288984, 0.429104] with:

Minimum: 0.288984 Maximum: 1.452028 Average: 0.7537 seconds

```
Number of cars 1
bbox:car_number ((932, 400), (1077, 498)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

32% | 409/1261 [18:14<37:59, 2.68s/it]

The times for each task are: [0.890675, 1.563474, 0.610109, 0.444055, 0.335797] with:

Minimum: 0.335797 Maximum: 1.563474 Average: 0.7688 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1070, 497)) : 1
```

```
The minimum distance from car: 1 is 1.4142135623730951
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 410/1261 [18:16<37:56, 2.67s/it]

The times for each task are: [0.8863, 0.740075, 1.386029, 0.395831, 0.32653] with:

Minimum: 0.32653 Maximum: 1.386029 Average: 0.747 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1070, 495)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 411/1261 [18:19<37:53, 2.67s/it]

The times for each task are: [0.567403, 0.919113, 1.397395, 0.384445, 0.286308] with:

Minimum: 0.286308 Maximum: 1.397395 Average: 0.7109 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1070, 497)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 412/1261 [18:21<37:50, 2.67s/it]

The times for each task are: [0.882317, 0.685351, 1.415272, 0.496996, 0.37418] with:

Minimum: 0.37418 Maximum: 1.415272 Average: 0.7708 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1070, 497)) : 1
The minimum distance from car: 1 is 7.0
```

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 413/1261 [18:24<37:48, 2.67s/it]

The times for each task are: [1.062955, 0.677952, 1.415998, 0.52176, 0.348774] with:

Minimum: 0.348774 Maximum: 1.415998 Average: 0.8055 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1070, 497)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 414/1261 [18:27<37:45, 2.68s/it]

The times for each task are: [1.372082, 1.078388, 0.705324, 0.354533, 0.373636] with:

Minimum: 0.354533 Maximum: 1.372082 Average: 0.7768 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1062, 495)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 415/1261 [18:30<37:42, 2.67s/it]

The times for each task are: [0.571699, 0.910788, 1.408803, 0.491363, 0.31092] with:

Minimum: 0.31092 Maximum: 1.408803 Average: 0.7387 seconds

```
Number of cars 1
bbox:car_number ((952, 400), (1062, 495)) : 1
The minimum distance from car: 1 is 14.0
totalCars: 2
```

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

33% | 416/1261 [18:32<37:40, 2.67s/it]

The times for each task are: [1.419624, 0.839137, 0.584177, 0.392, 0.322758] with:

Minimum: 0.322758 Maximum: 1.419624 Average: 0.7115 seconds

Number of cars 1
bbox:car_number ((927, 400), (1092, 495)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

33% | 417/1261 [18:35<37:37, 2.67s/it]

The times for each task are: [0.891755, 1.60188, 0.713444, 0.452445, 0.357031] with:

Minimum: 0.357031 Maximum: 1.60188 Average: 0.8033 seconds

Number of cars 2
bbox:car_number ((925, 400), (1062, 495)) : 1
The minimum distance from car: 1 is 16.0
bbox:car_number ((1065, 415), (1079, 459)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

33% | 418/1261 [18:38<37:35, 2.68s/it]

The times for each task are: [0.606488, 0.889503, 0.294124, 1.396679, 0.437665] with:

Minimum: 0.294124 Maximum: 1.396679 Average: 0.7249 seconds

Number of cars 2
bbox:car_number ((925, 400), (1056, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1067, 400), (1087, 462)) : 2

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 419/1261 [18:41<37:32, 2.68s/it]

The times for each task are: [0.55604, 1.027482, 0.390319, 1.620719, 0.346868] with:

Minimum: 0.346868 Maximum: 1.620719 Average: 0.7883 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1056, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 420/1261 [18:43<37:29, 2.68s/it]

The times for each task are: [0.94897, 0.633038, 1.471192, 0.382827, 0.294179] with:

Minimum: 0.294179 Maximum: 1.471192 Average: 0.746 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1048, 495)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

33% | 421/1261 [18:46<37:27, 2.68s/it]

The times for each task are: [0.851052, 0.630579, 1.504008, 0.46339, 0.318742] with:

Minimum: 0.318742 Maximum: 1.504008 Average: 0.7536 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1062, 495)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

33% | 422/1261 [18:48<37:24, 2.68s/it]

The times for each task are: [0.882422, 0.673783, 1.419828, 0.307398, 0.392048] with:

Minimum: 0.307398 Maximum: 1.419828 Average: 0.7351 seconds

```
Number of cars 1  
bbox:car_number ((925, 400), (1052, 495)) : 1  
The minimum distance from car: 1 is 1.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

34% | 423/1261 [18:51<37:21, 2.67s/it]

The times for each task are: [0.613297, 0.863486, 1.488808, 0.400275, 0.305469] with:

Minimum: 0.305469 Maximum: 1.488808 Average: 0.7343 seconds

```
Number of cars 1  
bbox:car_number ((913, 400), (1047, 492)) : 1  
The minimum distance from car: 1 is 8.06225774829855  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

34% | 424/1261 [18:53<37:18, 2.67s/it]

The times for each task are: [0.574154, 0.985845, 1.399287, 0.404627, 0.302457] with:

Minimum: 0.302457 Maximum: 1.399287 Average: 0.7333 seconds

```
Number of cars 1  
bbox:car_number ((913, 400), (1047, 495)) : 1  
The minimum distance from car: 1 is 1.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
34%| 425/1261 [18:56<37:16, 2.67s/it]
```

```
The times for each task are: [1.374394, 0.887937, 0.722836, 0.315717, 0.44167] with:
```

```
Minimum: 0.315717 Maximum: 1.374394 Average: 0.7485 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (1048, 498)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
34%| 426/1261 [18:59<37:13, 2.68s/it]
```

```
The times for each task are: [0.881433, 0.571021, 1.356002, 0.301255, 0.390155] with:
```

```
Minimum: 0.301255 Maximum: 1.356002 Average: 0.7 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (1048, 485)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
34%| 427/1261 [19:02<37:10, 2.67s/it]
```

```
The times for each task are: [0.881571, 0.593115, 1.45011, 0.401403, 0.285567] with:
```

```
Minimum: 0.285567 Maximum: 1.45011 Average: 0.7224 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (1052, 495)) : 1
The minimum distance from car: 1 is 9.433981132056603
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

34% | 428/1261 [19:04<37:07, 2.67s/it]

The times for each task are: [0.817275, 1.427708, 0.733315, 0.45671, 0.29704] with:

Minimum: 0.29704 Maximum: 1.427708 Average: 0.7464 seconds

Number of cars 1

bbox:car_number ((913, 400), (1047, 495)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

34% | 429/1261 [19:06<37:03, 2.67s/it]

The times for each task are: [0.908514, 0.618934, 1.441177, 0.285701, 0.397408] with:

Minimum: 0.285701 Maximum: 1.441177 Average: 0.7303 seconds

Number of cars 1

bbox:car_number ((913, 400), (1047, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

34% | 430/1261 [19:09<37:01, 2.67s/it]

The times for each task are: [1.349027, 0.842916, 0.738833, 0.312471, 0.386262] with:

Minimum: 0.312471 Maximum: 1.349027 Average: 0.7259 seconds

Number of cars 1

bbox:car_number ((925, 400), (1048, 495)) : 1

The minimum distance from car: 1 is 6.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

34% | 431/1261 [19:12<36:58, 2.67s/it]

The times for each task are: [1.369056, 0.622083, 0.882737, 0.307784, 0.382121] with:

Minimum: 0.307784 Maximum: 1.369056 Average: 0.7128 seconds

Number of cars 1

bbox:car_number ((913, 400), (1048, 492)) : 1

The minimum distance from car: 1 is 6.082762530298219

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

34% | 432/1261 [19:14<36:55, 2.67s/it]

The times for each task are: [0.869094, 1.658472, 0.664057, 0.434858, 0.293925] with:

Minimum: 0.293925 Maximum: 1.658472 Average: 0.7841 seconds

Number of cars 1

bbox:car_number ((913, 400), (1047, 492)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

34% | 433/1261 [19:17<36:52, 2.67s/it]

The times for each task are: [0.950755, 0.700303, 1.498172, 0.403818, 0.345259] with:

Minimum: 0.345259 Maximum: 1.498172 Average: 0.7797 seconds

Number of cars 1

bbox:car_number ((940, 400), (1048, 497)) : 1

The minimum distance from car: 1 is 14.142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

34% | 434/1261 [19:19<36:50, 2.67s/it]

The times for each task are: [0.869306, 0.715961, 1.414108, 0.303092, 0.398969] with:

Minimum: 0.303092 Maximum: 1.414108 Average: 0.7403 seconds

Number of cars 1

bbox:car_number ((925, 400), (1048, 497)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

34% | 435/1261 [19:22<36:47, 2.67s/it]

The times for each task are: [0.70153, 0.877795, 1.324082, 0.391039, 0.336012] with:

Minimum: 0.336012 Maximum: 1.324082 Average: 0.7261 seconds

Number of cars 1

bbox:car_number ((910, 400), (1047, 492)) : 1

The minimum distance from car: 1 is 8.246211251235321

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

35% | 436/1261 [19:25<36:44, 2.67s/it]

The times for each task are: [0.831272, 1.525911, 0.744447, 0.293378, 0.431433] with:

Minimum: 0.293378 Maximum: 1.525911 Average: 0.7653 seconds

Number of cars 1

bbox:car_number ((913, 400), (1047, 492)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

35% | 437/1261 [19:27<36:42, 2.67s/it]

The times for each task are: [0.886143, 0.606712, 1.578273, 0.299375, 0.444047] with:

Minimum: 0.299375 Maximum: 1.578273 Average: 0.7629 seconds

Number of cars 1

bbox:car_number ((910, 400), (1048, 495)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

35% | 438/1261 [19:30<36:39, 2.67s/it]

The times for each task are: [0.938662, 1.385752, 0.576081, 0.432433, 0.286166] with:

Minimum: 0.286166 Maximum: 1.385752 Average: 0.7238 seconds

Number of cars 1

bbox:car_number ((940, 400), (1047, 492)) : 1

The minimum distance from car: 1 is 14.035668847618199

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

35% | 439/1261 [19:33<36:36, 2.67s/it]

The times for each task are: [0.909135, 0.644599, 1.439286, 0.443581, 0.388124] with:

Minimum: 0.388124 Maximum: 1.439286 Average: 0.7649 seconds

Number of cars 1

bbox:car_number ((940, 400), (1052, 492)) : 1

The minimum distance from car: 1 is 3.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

35% | 440/1261 [19:35<36:33, 2.67s/it]

The times for each task are: [0.914474, 0.59633, 1.64513, 0.285399, 0.391747] with:

Minimum: 0.285399 Maximum: 1.64513 Average: 0.7666 seconds

Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 9.055385138137417
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

35% | 441/1261 [19:38<36:30, 2.67s/it]

The times for each task are: [0.62567, 1.390378, 1.00838, 0.421904, 0.30981] with:

Minimum: 0.30981 Maximum: 1.390378 Average: 0.7512 seconds

Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

35% | 442/1261 [19:40<36:27, 2.67s/it]

The times for each task are: [0.587834, 0.969401, 1.509618, 0.387358, 0.282827] with:

Minimum: 0.282827 Maximum: 1.509618 Average: 0.7474 seconds

Number of cars 1
bbox:car_number ((913, 400), (1056, 492)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

35% | 443/1261 [19:43<36:25, 2.67s/it]

The times for each task are: [0.607619, 0.836347, 0.433632, 0.302777, 1.630218] with:

Minimum: 0.302777 Maximum: 1.630218 Average: 0.7621 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 6.082762530298219
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

35% | 444/1261 [19:46<36:23, 2.67s/it]

The times for each task are: [0.623756, 0.96836, 1.402624, 0.513628, 0.299017] with:

Minimum: 0.299017 Maximum: 1.402624 Average: 0.7615 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 9.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

35% | 445/1261 [19:49<36:20, 2.67s/it]

The times for each task are: [0.866489, 0.699719, 1.453849, 0.402522, 0.336802] with:

Minimum: 0.336802 Maximum: 1.453849 Average: 0.7519 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

35% | 446/1261 [19:51<36:17, 2.67s/it]

The times for each task are: [0.847609, 0.598724, 1.400823, 0.397919, 0.429425] with:

Minimum: 0.397919 Maximum: 1.400823 Average: 0.7349 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (1029, 495)) : 1
The minimum distance from car: 1 is 10.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

35%| 447/1261 [19:54<36:15, 2.67s/it]

The times for each task are: [0.581714, 0.904253, 1.437615, 0.409719, 0.315922] with:
Minimum: 0.315922 Maximum: 1.437615 Average: 0.7298 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1029, 495)) : 1
The minimum distance from car: 1 is 9.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

36%| 448/1261 [19:56<36:12, 2.67s/it]

The times for each task are: [1.335542, 0.848779, 0.677343, 0.404132, 0.305695] with:
Minimum: 0.305695 Maximum: 1.335542 Average: 0.7143 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1029, 485)) : 1
The minimum distance from car: 1 is 7.810249675906654
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

36%| 449/1261 [19:59<36:09, 2.67s/it]

The times for each task are: [0.904714, 0.590907, 1.5204, 0.404744, 0.297419] with:
Minimum: 0.297419 Maximum: 1.5204 Average: 0.7436 seconds

Number of cars 1

```
bbox:car_number ((927, 400), (1052, 495)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 450/1261 [20:02<36:06, 2.67s/it]

The times for each task are: [0.628815, 0.988986, 1.419441, 0.349735, 0.548555] with:
Minimum: 0.349735 Maximum: 1.419441 Average: 0.7871 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1056, 495)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 451/1261 [20:04<36:04, 2.67s/it]

The times for each task are: [0.654956, 1.324203, 0.826961, 0.426116, 0.447607] with:
Minimum: 0.426116 Maximum: 1.324203 Average: 0.736 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1052, 498)) : 1
The minimum distance from car: 1 is 9.219544457292887
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 452/1261 [20:07<36:01, 2.67s/it]

The times for each task are: [0.684038, 1.029611, 1.375885, 0.468171, 0.389874] with:
Minimum: 0.389874 Maximum: 1.375885 Average: 0.7895 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1077, 498)) : 1
```

```
The minimum distance from car: 1 is 26.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 453/1261 [20:10<35:59, 2.67s/it]

The times for each task are: [1.47772, 0.683448, 1.001916, 0.2983, 0.53317] with:

Minimum: 0.2983 Maximum: 1.47772 Average: 0.7989 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 28.071337695236398
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 454/1261 [20:13<35:56, 2.67s/it]

The times for each task are: [0.96562, 0.667098, 1.653901, 0.525026, 0.42836] with:

Minimum: 0.42836 Maximum: 1.653901 Average: 0.848 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 455/1261 [20:15<35:53, 2.67s/it]

The times for each task are: [0.914277, 0.592952, 1.647586, 0.385253, 0.36044] with:

Minimum: 0.36044 Maximum: 1.647586 Average: 0.7801 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 7.0
```

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 456/1261 [20:18<35:50, 2.67s/it]

The times for each task are: [0.903778, 0.639859, 0.411404, 0.301705, 1.527037] with:

Minimum: 0.301705 Maximum: 1.527037 Average: 0.7568 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 457/1261 [20:20<35:47, 2.67s/it]

The times for each task are: [0.580084, 0.929242, 1.398505, 0.354148, 0.522419] with:

Minimum: 0.354148 Maximum: 1.398505 Average: 0.7569 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1032, 497)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

36% | 458/1261 [20:23<35:45, 2.67s/it]

The times for each task are: [0.986161, 0.684344, 1.379572, 0.294824, 0.455204] with:

Minimum: 0.294824 Maximum: 1.379572 Average: 0.76 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (1029, 498)) : 1
The minimum distance from car: 1 is 10.04987562112089
totalCars: 2
```

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

36% | 459/1261 [20:26<35:43, 2.67s/it]

The times for each task are: [0.992717, 0.607682, 1.379936, 0.453827, 0.374465] with:

Minimum: 0.374465 Maximum: 1.379936 Average: 0.7617 seconds

Number of cars 1
bbox:car_number ((927, 400), (1032, 497)) : 1
The minimum distance from car: 1 is 10.04987562112089
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

36% | 460/1261 [20:29<35:40, 2.67s/it]

The times for each task are: [0.899099, 0.701187, 1.392661, 0.401682, 0.38701] with:

Minimum: 0.38701 Maximum: 1.392661 Average: 0.7563 seconds

Number of cars 1
bbox:car_number ((910, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 8.06225774829855
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

37% | 461/1261 [20:31<35:37, 2.67s/it]

The times for each task are: [0.611695, 0.908253, 1.554362, 0.383649, 0.299757] with:

Minimum: 0.299757 Maximum: 1.554362 Average: 0.7515 seconds

Number of cars 1
bbox:car_number ((927, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

```
Length of task list: 5
Number of processes used: 3
```

```
37%|      | 462/1261 [20:34<35:35,  2.67s/it]
```

```
The times for each task are: [0.656673, 0.879945, 1.495134, 0.434027, 0.295145] with:
```

```
Minimum: 0.295145 Maximum: 1.495134 Average: 0.7522 seconds
```

```
Number of cars 1
bbox:car_number ((927, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
37%|      | 463/1261 [20:36<35:31,  2.67s/it]
```

```
The times for each task are: [0.876869, 0.676656, 0.29717, 1.352561, 0.435539] with:
```

```
Minimum: 0.29717 Maximum: 1.352561 Average: 0.7278 seconds
```

```
Number of cars 1
bbox:car_number ((896, 400), (1056, 495)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
37%|      | 464/1261 [20:39<35:29,  2.67s/it]
```

```
The times for each task are: [1.029157, 0.697706, 1.335372, 0.467047, 0.299347] with:
```

```
Minimum: 0.299347 Maximum: 1.335372 Average: 0.7657 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (1032, 495)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

37% | 465/1261 [20:42<35:26, 2.67s/it]

The times for each task are: [0.63835, 0.85814, 1.410063, 0.393234, 0.300416] with:

Minimum: 0.300416 Maximum: 1.410063 Average: 0.72 seconds

Number of cars 1

bbox:car_number ((913, 400), (1029, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

37% | 466/1261 [20:44<35:23, 2.67s/it]

The times for each task are: [0.622727, 0.846495, 1.619074, 0.339738, 0.396252] with:

Minimum: 0.339738 Maximum: 1.619074 Average: 0.7649 seconds

Number of cars 1

bbox:car_number ((905, 400), (1029, 497)) : 1

The minimum distance from car: 1 is 4.123105625617661

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

37% | 467/1261 [20:47<35:20, 2.67s/it]

The times for each task are: [0.859269, 0.567437, 0.296525, 1.59682, 0.38633] with:

Minimum: 0.296525 Maximum: 1.59682 Average: 0.7413 seconds

Number of cars 1

bbox:car_number ((905, 400), (1029, 495)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

37% | 468/1261 [20:50<35:18, 2.67s/it]

The times for each task are: [0.966023, 1.368362, 0.711634, 0.330795, 0.432529] with:

Minimum: 0.330795 Maximum: 1.368362 Average: 0.7619 seconds

Number of cars 1
bbox:car_number ((927, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 9.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

37% | 469/1261 [20:52<35:15, 2.67s/it]

The times for each task are: [0.851775, 0.73291, 1.394924, 0.46813, 0.335902] with:

Minimum: 0.335902 Maximum: 1.394924 Average: 0.7567 seconds

Number of cars 1
bbox:car_number ((927, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

37% | 470/1261 [20:55<35:13, 2.67s/it]

The times for each task are: [0.893664, 1.61954, 0.670431, 0.39479, 0.331606] with:

Minimum: 0.331606 Maximum: 1.61954 Average: 0.782 seconds

Number of cars 1
bbox:car_number ((913, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 8.602325267042627
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3

37% | 471/1261 [20:58<35:10, 2.67s/it]

The times for each task are: [0.917788, 0.644467, 1.578696, 0.291379, 0.381398] with:

Minimum: 0.291379 Maximum: 1.578696 Average: 0.7627 seconds

Number of cars 1

bbox:car_number ((929, 400), (1024, 495)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

37% | 472/1261 [21:00<35:07, 2.67s/it]

The times for each task are: [0.618237, 0.889001, 1.395725, 0.391046, 0.415763] with:

Minimum: 0.391046 Maximum: 1.395725 Average: 0.742 seconds

Number of cars 1

bbox:car_number ((927, 400), (1025, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 473/1261 [21:03<35:05, 2.67s/it]

The times for each task are: [0.867942, 0.646742, 1.595895, 0.364406, 0.41055] with:

Minimum: 0.364406 Maximum: 1.595895 Average: 0.7771 seconds

Number of cars 1

bbox:car_number ((913, 400), (1025, 495)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 474/1261 [21:06<35:02, 2.67s/it]

The times for each task are: [0.616868, 1.022905, 1.389227, 0.395133, 0.334175] with:

Minimum: 0.334175 Maximum: 1.389227 Average: 0.7517 seconds

Number of cars 1

bbox:car_number ((910, 400), (1024, 495)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 475/1261 [21:08<34:59, 2.67s/it]

The times for each task are: [0.921772, 0.620047, 0.42009, 0.293056, 1.519291] with:

Minimum: 0.293056 Maximum: 1.519291 Average: 0.7549 seconds

Number of cars 1

bbox:car_number ((927, 400), (1025, 485)) : 1

The minimum distance from car: 1 is 10.295630140987

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 476/1261 [21:11<34:57, 2.67s/it]

The times for each task are: [0.585721, 0.928589, 1.393548, 0.464376, 0.329862] with:

Minimum: 0.329862 Maximum: 1.393548 Average: 0.7404 seconds

Number of cars 1

bbox:car_number ((910, 400), (1025, 495)) : 1

The minimum distance from car: 1 is 10.295630140987

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

38% | 477/1261 [21:14<34:54, 2.67s/it]

The times for each task are: [0.901808, 0.621369, 1.636497, 0.398991, 0.347865] with:

Minimum: 0.347865 Maximum: 1.636497 Average: 0.7813 seconds

Number of cars 2
bbox:car_number ((913, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((896, 430), (908, 462)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 478/1261 [21:17<34:52, 2.67s/it]

The times for each task are: [0.610112, 1.047024, 0.407092, 1.552166, 0.394218] with:

Minimum: 0.394218 Maximum: 1.552166 Average: 0.8021 seconds

Number of cars 1
bbox:car_number ((896, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 10.295630140987
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 479/1261 [21:19<34:49, 2.67s/it]

The times for each task are: [0.885773, 0.613867, 1.607561, 0.304898, 0.407799] with:

Minimum: 0.304898 Maximum: 1.607561 Average: 0.764 seconds

Number of cars 1
bbox:car_number ((896, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 480/1261 [21:22<34:46, 2.67s/it]

The times for each task are: [1.019295, 0.674049, 1.462358, 0.470653, 0.387724] with:

Minimum: 0.387724 Maximum: 1.462358 Average: 0.8028 seconds

Number of cars 1
bbox:car_number ((895, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 481/1261 [21:24<34:43, 2.67s/it]

The times for each task are: [0.988584, 0.674314, 1.356071, 0.287862, 0.430326] with:

Minimum: 0.287862 Maximum: 1.356071 Average: 0.7474 seconds

Number of cars 1
bbox:car_number ((895, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 482/1261 [21:27<34:40, 2.67s/it]

The times for each task are: [0.915444, 1.521878, 0.605885, 0.334808, 0.391056] with:

Minimum: 0.334808 Maximum: 1.521878 Average: 0.7538 seconds

Number of cars 1
bbox:car_number ((895, 400), (1024, 495)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

38% | 483/1261 [21:30<34:37, 2.67s/it]

The times for each task are: [0.575587, 0.957547, 0.465189, 1.621925, 0.3384] with:

Minimum: 0.3384 Maximum: 1.621925 Average: 0.7917 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

38% | 484/1261 [21:32<34:35, 2.67s/it]

The times for each task are: [0.668999, 1.474393, 0.877761, 0.410899, 0.291838] with:

Minimum: 0.291838 Maximum: 1.474393 Average: 0.7448 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1024, 492)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

38% | 485/1261 [21:35<34:32, 2.67s/it]

The times for each task are: [0.857156, 0.652591, 0.464189, 1.690849, 0.336186] with:

Minimum: 0.336186 Maximum: 1.690849 Average: 0.8002 seconds

```
Number of cars 2
bbox:car_number ((895, 400), (1025, 492)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1225, 445), (1227, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 486/1261 [21:38<34:30, 2.67s/it]

The times for each task are: [0.630196, 0.948481, 1.604357, 0.477451, 0.456916] with:

Minimum: 0.456916 Maximum: 1.604357 Average: 0.8235 seconds

```
Number of cars 1
bbox:car_number ((895, 400), (1025, 485)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 487/1261 [21:40<34:27, 2.67s/it]

The times for each task are: [0.969876, 0.629783, 1.403177, 0.389696, 0.302843] with:

Minimum: 0.302843 Maximum: 1.403177 Average: 0.7391 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 488/1261 [21:43<34:24, 2.67s/it]

The times for each task are: [0.613816, 0.888379, 1.629992, 0.315407, 0.41832] with:

Minimum: 0.315407 Maximum: 1.629992 Average: 0.7732 seconds

```
Number of cars 1
bbox:car_number ((927, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

39% | 489/1261 [21:46<34:22, 2.67s/it]

The times for each task are: [0.620777, 1.393479, 1.023144, 0.472942, 0.282267] with:

Minimum: 0.282267 Maximum: 1.393479 Average: 0.7585 seconds

```
Number of cars 2
bbox:car_number ((925, 400), (1024, 492)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((896, 430), (915, 459)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

39% | 490/1261 [21:48<34:19, 2.67s/it]

The times for each task are: [0.955068, 0.614807, 1.451223, 0.45961, 0.375421] with:

Minimum: 0.375421 Maximum: 1.451223 Average: 0.7712 seconds

```
Number of cars 4
bbox:car_number ((925, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((895, 415), (912, 447)) : 2
bbox:car_number ((1180, 445), (1197, 477)) : 3
bbox:car_number ((1210, 445), (1227, 477)) : 4
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

39% | 491/1261 [21:51<34:16, 2.67s/it]

The times for each task are: [0.601626, 1.416338, 1.031375, 0.381151, 0.344644] with:

Minimum: 0.344644 Maximum: 1.416338 Average: 0.755 seconds

```
Number of cars 1
bbox:car_number ((895, 400), (1047, 495)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]
Length of task list: 5
Number of processes used: 3
```

39% | 492/1261 [21:54<34:14, 2.67s/it]

The times for each task are: [0.876075, 0.635242, 1.38929, 0.39789, 0.335926] with:

Minimum: 0.335926 Maximum: 1.38929 Average: 0.7269 seconds

Number of cars 2
bbox:car_number ((895, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 13.0
bbox:car_number ((1165, 445), (1181, 477)) : 2
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1175, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400), (1219, 492))]
Length of task list: 5
Number of processes used: 3

39% | 493/1261 [21:57<34:12, 2.67s/it]

The times for each task are: [0.96535, 0.715429, 1.367173, 0.449993, 0.335583] with:

Minimum: 0.335583 Maximum: 1.367173 Average: 0.7667 seconds
Number of cars 3
bbox:car_number ((913, 400), (1025, 495)) : 1
The minimum distance from car: 1 is 11.180339887498949
bbox:car_number ((1180, 438), (1200, 477)) : 2
bbox:car_number ((1210, 445), (1227, 477)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1175, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400), (1219, 492))]
Length of task list: 5
Number of processes used: 3

39% | 494/1261 [22:00<34:09, 2.67s/it]

The times for each task are: [0.963816, 1.396944, 0.695246, 0.401378, 0.29382] with:

Minimum: 0.29382 Maximum: 1.396944 Average: 0.7502 seconds
Number of cars 2
bbox:car_number ((896, 400), (1024, 485)) : 1
The minimum distance from car: 1 is 10.295630140987
bbox:car_number ((1160, 438), (1219, 492)) : 2
The minimum distance from car: 1 is 23.53720459187964
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1175, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400), (1219, 492))]
Length of task list: 5
Number of processes used: 3

39% | 495/1261 [22:02<34:07, 2.67s/it]

The times for each task are: [1.498763, 0.597484, 0.864963, 0.293605, 0.458144] with:

Minimum: 0.293605 Maximum: 1.498763 Average: 0.7426 seconds

Number of cars 2

bbox:car_number ((896, 400), (1024, 495)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1179, 445), (1212, 477)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

39% | 496/1261 [22:05<34:04, 2.67s/it]

The times for each task are: [1.444183, 0.90824, 0.726047, 0.305643, 0.40893] with:

Minimum: 0.305643 Maximum: 1.444183 Average: 0.7586 seconds

Number of cars 2

bbox:car_number ((927, 400), (1010, 485)) : 1

The minimum distance from car: 1 is 9.433981132056603

bbox:car_number ((1210, 445), (1212, 477)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

39% | 497/1261 [22:08<34:01, 2.67s/it]

The times for each task are: [1.48147, 0.950703, 0.693345, 0.298473, 0.436472] with:

Minimum: 0.298473 Maximum: 1.48147 Average: 0.7721 seconds

Number of cars 1

bbox:car_number ((927, 400), (1017, 485)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

39% | 498/1261 [22:10<33:59, 2.67s/it]

The times for each task are: [1.346821, 0.893189, 0.678336, 0.399664, 0.33248] with:

Minimum: 0.33248 Maximum: 1.346821 Average: 0.7301 seconds

Number of cars 1

bbox:car_number ((913, 400), (1025, 495)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

40% | 499/1261 [22:13<33:56, 2.67s/it]

The times for each task are: [0.897186, 0.625304, 1.405569, 0.316583, 0.392013] with:

Minimum: 0.316583 Maximum: 1.405569 Average: 0.7273 seconds

Number of cars 3

bbox:car_number ((913, 400), (1025, 485)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1180, 445), (1197, 477)) : 2

bbox:car_number ((1210, 445), (1227, 477)) : 3

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

Length of task list: 5

Number of processes used: 3

40% | 500/1261 [22:16<33:54, 2.67s/it]

The times for each task are: [1.377801, 0.947774, 0.686575, 0.442614, 0.358542] with:

Minimum: 0.358542 Maximum: 1.377801 Average: 0.7627 seconds

Number of cars 1

bbox:car_number ((940, 400), (1024, 485)) : 1

The minimum distance from car: 1 is 13.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 485))]

```
Length of task list: 5
Number of processes used: 3
```

```
40%|      | 501/1261 [22:19<33:51,  2.67s/it]
```

```
The times for each task are: [0.983668, 0.669539, 0.396791, 1.632892, 0.369931] with:
```

```
Minimum: 0.369931 Maximum: 1.632892 Average: 0.8106 seconds
```

```
Number of cars 0
Length of task list: 5
Number of processes used: 3
```

```
40%|      | 502/1261 [22:22<33:49,  2.67s/it]
```

```
The times for each task are: [0.697327, 0.897809, 1.52535, 0.298507, 0.427229] with:
```

```
Minimum: 0.298507 Maximum: 1.52535 Average: 0.7692 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (998, 459)) : 1
The minimum distance from car: 1 is 29.966648127543394
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
40%|      | 503/1261 [22:24<33:46,  2.67s/it]
```

```
The times for each task are: [0.873521, 0.688441, 1.392583, 0.478111, 0.273624] with:
```

```
Minimum: 0.273624 Maximum: 1.392583 Average: 0.7413 seconds
```

```
Number of cars 3
bbox:car_number ((929, 415), (942, 459)) : 1
bbox:car_number ((951, 419), (953, 459)) : 2
bbox:car_number ((955, 419), (998, 462)) : 3
Length of task list: 5
Number of processes used: 3
```

```
40%|      | 504/1261 [22:27<33:43,  2.67s/it]
```

The times for each task are: [0.996855, 1.483991, 0.650635, 0.430737, 0.296328] with:

Minimum: 0.296328 Maximum: 1.483991 Average: 0.7717 seconds

Number of cars 2

bbox:car_number ((951, 400), (998, 459)) : 1

bbox:car_number ((929, 415), (942, 459)) : 2

Length of task list: 5

Number of processes used: 3

40% | 505/1261 [22:30<33:40, 2.67s/it]

The times for each task are: [0.897197, 1.368251, 0.737528, 0.341351, 0.39383] with:

Minimum: 0.341351 Maximum: 1.368251 Average: 0.7476 seconds

Number of cars 2

bbox:car_number ((929, 415), (942, 447)) : 1

bbox:car_number ((951, 419), (953, 459)) : 2

Length of task list: 5

Number of processes used: 3

40% | 506/1261 [22:32<33:38, 2.67s/it]

The times for each task are: [0.608758, 1.335508, 1.003664, 0.398468, 0.331935] with:

Minimum: 0.331935 Maximum: 1.335508 Average: 0.7357 seconds

Number of cars 2

bbox:car_number ((929, 400), (942, 459)) : 1

bbox:car_number ((951, 400), (991, 459)) : 2

Length of task list: 5

Number of processes used: 3

40% | 507/1261 [22:35<33:35, 2.67s/it]

The times for each task are: [0.980319, 1.395081, 0.582241, 0.399529, 0.285144] with:

Minimum: 0.285144 Maximum: 1.395081 Average: 0.7285 seconds

Number of cars 1

bbox:car_number ((913, 400), (1002, 472)) : 1

The minimum distance from car: 1 is 7.280109889280518

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (929, 400))]
Length of task list: 5
Number of processes used: 3
```

40% | 508/1261 [22:38<33:32, 2.67s/it]

The times for each task are: [0.634049, 0.862774, 1.411363, 0.429114, 0.301004] with:

Minimum: 0.301004 Maximum: 1.411363 Average: 0.7277 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (998, 472)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (929, 400))]
Length of task list: 5
Number of processes used: 3
```

40% | 509/1261 [22:40<33:30, 2.67s/it]

The times for each task are: [0.684132, 0.840753, 1.331504, 0.491997, 0.296887] with:

Minimum: 0.296887 Maximum: 1.331504 Average: 0.7291 seconds

```
Number of cars 2
bbox:car_number ((929, 415), (942, 459)) : 1
bbox:car_number ((951, 419), (953, 459)) : 2
Length of task list: 5
Number of processes used: 3
```

40% | 510/1261 [22:43<33:27, 2.67s/it]

The times for each task are: [0.63982, 0.91218, 1.404867, 0.297363, 0.388529] with:

Minimum: 0.297363 Maximum: 1.404867 Average: 0.7286 seconds

```
Number of cars 2
bbox:car_number ((932, 400), (942, 447)) : 1
bbox:car_number ((951, 400), (998, 459)) : 2
Length of task list: 5
Number of processes used: 3
```

41% | 511/1261 [22:45<33:24, 2.67s/it]

The times for each task are: [0.850712, 0.674041, 1.586178, 0.359208, 0.415363] with:

Minimum: 0.359208 Maximum: 1.586178 Average: 0.7771 seconds

Number of cars 1

bbox:car_number ((929, 400), (998, 462)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 512/1261 [22:48<33:22, 2.67s/it]

The times for each task are: [0.981236, 0.647766, 1.468739, 0.290314, 0.443808] with:

Minimum: 0.290314 Maximum: 1.468739 Average: 0.7664 seconds

Number of cars 1

bbox:car_number ((925, 400), (991, 472)) : 1

The minimum distance from car: 1 is 7.0710678118654755

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 513/1261 [22:51<33:19, 2.67s/it]

The times for each task are: [0.704025, 0.914315, 1.412499, 0.298988, 0.450399] with:

Minimum: 0.298988 Maximum: 1.412499 Average: 0.756 seconds

Number of cars 1

bbox:car_number ((932, 400), (1001, 462)) : 1

The minimum distance from car: 1 is 9.433981132056603

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 514/1261 [22:54<33:16, 2.67s/it]

The times for each task are: [1.330964, 0.87923, 0.605934, 0.41339, 0.307685] with:

Minimum: 0.307685 Maximum: 1.330964 Average: 0.7074 seconds

Number of cars 1

bbox:car_number ((925, 400), (1002, 472)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 515/1261 [22:56<33:14, 2.67s/it]

The times for each task are: [0.707804, 0.901286, 1.676769, 0.415968, 0.358152] with:

Minimum: 0.358152 Maximum: 1.676769 Average: 0.812 seconds

Number of cars 1

bbox:car_number ((929, 400), (1001, 478)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 516/1261 [22:59<33:11, 2.67s/it]

The times for each task are: [0.856618, 1.63445, 0.654904, 0.405037, 0.350496] with:

Minimum: 0.350496 Maximum: 1.63445 Average: 0.7803 seconds

Number of cars 1

bbox:car_number ((913, 400), (1010, 485)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

41% | 517/1261 [23:02<33:08, 2.67s/it]

The times for each task are: [1.019223, 0.601272, 1.416994, 0.44097, 0.296004] with:

Minimum: 0.296004 Maximum: 1.416994 Average: 0.7549 seconds

Number of cars 1
bbox:car_number ((913, 400), (1001, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

41% | 518/1261 [23:04<33:06, 2.67s/it]

The times for each task are: [0.92036, 0.676487, 0.41011, 1.715579, 0.290722] with:

Minimum: 0.290722 Maximum: 1.715579 Average: 0.8027 seconds

Number of cars 1
bbox:car_number ((925, 400), (1001, 478)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

41% | 519/1261 [23:07<33:04, 2.67s/it]

The times for each task are: [0.619261, 1.037094, 0.344694, 0.456885, 1.61943] with:

Minimum: 0.344694 Maximum: 1.61943 Average: 0.8155 seconds

Number of cars 1
bbox:car_number ((913, 400), (1010, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

41% | 520/1261 [23:10<33:01, 2.67s/it]

The times for each task are: [0.831326, 0.40172, 0.708896, 1.583906, 0.281507] with:

Minimum: 0.281507 Maximum: 1.583906 Average: 0.7615 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (998, 478))]
Length of task list: 5
Number of processes used: 3
```

41% | 521/1261 [23:13<32:58, 2.67s/it]

The times for each task are: [0.872279, 0.695351, 1.366953, 0.4454, 0.332344] with:

Minimum: 0.332344 Maximum: 1.366953 Average: 0.7425 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (998, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (998, 478))]
Length of task list: 5
Number of processes used: 3
```

41% | 522/1261 [23:15<32:56, 2.67s/it]

The times for each task are: [0.591465, 1.399539, 0.922085, 0.421225, 0.358011] with:

Minimum: 0.358011 Maximum: 1.399539 Average: 0.7385 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (998, 478))]
Length of task list: 5
Number of processes used: 3
```

41% | 523/1261 [23:18<32:53, 2.67s/it]

The times for each task are: [0.924724, 0.621087, 1.682484, 0.402165, 0.279755] with:

Minimum: 0.279755 Maximum: 1.682484 Average: 0.782 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1010, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42%| 524/1261 [23:21<32:50, 2.67s/it]

The times for each task are: [0.618849, 1.38883, 0.991873, 0.471748, 0.278857] with:
Minimum: 0.278857 Maximum: 1.38883 Average: 0.75 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1010, 472)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42%| 525/1261 [23:23<32:47, 2.67s/it]

The times for each task are: [0.866679, 0.667511, 1.534766, 0.414179, 0.283335] with:
Minimum: 0.283335 Maximum: 1.534766 Average: 0.7533 seconds

```
Number of cars 1
bbox:car_number ((940, 415), (991, 462)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42%| 526/1261 [23:26<32:44, 2.67s/it]

The times for each task are: [0.826437, 1.391458, 0.671941, 0.409327, 0.281357] with:
Minimum: 0.281357 Maximum: 1.391458 Average: 0.7161 seconds

Number of cars 1

```
bbox:car_number ((932, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 2.23606797749979
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 527/1261 [23:28<32:42, 2.67s/it]

The times for each task are: [1.40401, 0.714249, 0.948234, 0.394871, 0.374426] with:

Minimum: 0.374426 Maximum: 1.40401 Average: 0.7672 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1010, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 528/1261 [23:31<32:39, 2.67s/it]

The times for each task are: [0.879024, 1.34436, 0.702753, 0.40773, 0.389535] with:

Minimum: 0.389535 Maximum: 1.34436 Average: 0.7447 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1010, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 529/1261 [23:34<32:37, 2.67s/it]

The times for each task are: [0.921222, 0.625425, 0.347838, 0.474078, 1.472507] with:

Minimum: 0.347838 Maximum: 1.472507 Average: 0.7682 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1010, 478)) : 1
```

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

42% | 530/1261 [23:37<32:34, 2.67s/it]

The times for each task are: [0.948354, 1.422759, 0.667152, 0.472976, 0.419062] with:

Minimum: 0.419062 Maximum: 1.422759 Average: 0.7861 seconds

Number of cars 2

bbox:car_number ((925, 400), (1010, 477)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((552, 490), (567, 507)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

42% | 531/1261 [23:39<32:31, 2.67s/it]

The times for each task are: [0.927403, 1.399633, 0.595267, 0.49767, 0.414173] with:

Minimum: 0.414173 Maximum: 1.399633 Average: 0.7668 seconds

Number of cars 2

bbox:car_number ((925, 400), (998, 472)) : 1

The minimum distance from car: 1 is 6.324555320336759

bbox:car_number ((1000, 419), (1001, 462)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

42% | 532/1261 [23:42<32:29, 2.67s/it]

The times for each task are: [0.892801, 0.727808, 1.454202, 0.38937, 0.328458] with:

Minimum: 0.328458 Maximum: 1.454202 Average: 0.7585 seconds

Number of cars 1

```
bbox:car_number ((913, 400), (1010, 485)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 533/1261 [23:45<32:26, 2.67s/it]

The times for each task are: [0.633017, 0.885946, 1.493424, 0.300407, 0.375326] with:
Minimum: 0.300407 Maximum: 1.493424 Average: 0.7376 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (994, 462)) : 1
The minimum distance from car: 1 is 11.180339887498949
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 534/1261 [23:47<32:23, 2.67s/it]

The times for each task are: [1.304625, 0.946296, 0.715957, 0.341682, 0.393698] with:
Minimum: 0.341682 Maximum: 1.304625 Average: 0.7405 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1010, 485)) : 1
The minimum distance from car: 1 is 12.529964086141668
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

42% | 535/1261 [23:50<32:20, 2.67s/it]

The times for each task are: [0.671648, 0.930511, 1.555098, 0.367614, 0.546533] with:
Minimum: 0.367614 Maximum: 1.555098 Average: 0.8143 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (1010, 478)) : 1
```

```
The minimum distance from car: 1 is 8.54400374531753
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (932, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 536/1261 [23:52<32:18, 2.67s/it]

The times for each task are: [1.382222, 0.684938, 1.07843, 0.436355, 0.306625] with:

Minimum: 0.306625 Maximum: 1.382222 Average: 0.7777 seconds

```
Number of cars 1
bbox:car_number ((932, 400), (1001, 472)) : 1
The minimum distance from car: 1 is 5.830951894845301
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (932, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 537/1261 [23:55<32:15, 2.67s/it]

The times for each task are: [0.683489, 0.87869, 1.553366, 0.426331, 0.38522] with:

Minimum: 0.38522 Maximum: 1.553366 Average: 0.7854 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 478)) : 1
The minimum distance from car: 1 is 13.341664064126334
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 538/1261 [23:58<32:12, 2.67s/it]

The times for each task are: [0.695076, 0.90508, 1.413676, 0.479969, 0.380075] with:

Minimum: 0.380075 Maximum: 1.413676 Average: 0.7748 seconds

```
Number of cars 1
bbox:car_number ((925, 400), (1001, 472)) : 1
The minimum distance from car: 1 is 10.44030650891055
```

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 539/1261 [24:01<32:10, 2.67s/it]

The times for each task are: [0.627222, 1.055559, 1.485529, 0.408448, 0.385185] with:

Minimum: 0.385185 Maximum: 1.485529 Average: 0.7924 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 472)) : 1
The minimum distance from car: 1 is 10.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 540/1261 [24:03<32:07, 2.67s/it]

The times for each task are: [0.614048, 1.021006, 1.410194, 0.376014, 0.286813] with:

Minimum: 0.286813 Maximum: 1.410194 Average: 0.7416 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

43% | 541/1261 [24:06<32:04, 2.67s/it]

The times for each task are: [0.628577, 0.89048, 0.291894, 1.605225, 0.431123] with:

Minimum: 0.291894 Maximum: 1.605225 Average: 0.7695 seconds

```
Number of cars 1
bbox:car_number ((929, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 13.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (925, 400), (1002, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 542/1261 [24:09<32:02, 2.67s/it]
```

```
The times for each task are: [0.673003, 0.969107, 1.555661, 0.41088, 0.317027] with:
```

```
Minimum: 0.317027 Maximum: 1.555661 Average: 0.7851 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (1002, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (1002, 485)) : 1
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 543/1261 [24:11<31:59, 2.67s/it]
```

```
The times for each task are: [0.634175, 1.46946, 0.918536, 0.30282, 0.406658] with:
```

```
Minimum: 0.30282 Maximum: 1.46946 Average: 0.7463 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (1002, 485)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (1002, 485)) : 1
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 544/1261 [24:14<31:57, 2.67s/it]
```

```
The times for each task are: [0.544109, 0.90041, 1.593363, 0.288089, 0.429448] with:
```

```
Minimum: 0.288089 Maximum: 1.593363 Average: 0.7511 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (998, 478)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (998, 478)) : 1
```

```
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 545/1261 [24:17<31:54,  2.67s/it]
```

```
The times for each task are: [1.515204, 0.89619, 0.615727, 0.459085, 0.381011] with:
```

```
Minimum: 0.381011 Maximum: 1.515204 Average: 0.7734 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (998, 477)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 546/1261 [24:20<31:52,  2.67s/it]
```

```
The times for each task are: [0.972881, 0.568197, 0.278288, 1.647114, 0.445903] with:
```

```
Minimum: 0.278288 Maximum: 1.647114 Average: 0.7825 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (991, 472)) : 1
The minimum distance from car: 1 is 11.180339887498949
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
43%|      | 547/1261 [24:22<31:49,  2.67s/it]
```

```
The times for each task are: [0.859545, 1.395554, 0.66053, 0.317135, 0.434316] with:
```

```
Minimum: 0.317135 Maximum: 1.395554 Average: 0.7334 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (991, 472)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

43% | 548/1261 [24:25<31:46, 2.67s/it]

The times for each task are: [1.35686, 0.841882, 0.418665, 0.7419, 0.3366] with:

Minimum: 0.3366 Maximum: 1.35686 Average: 0.7392 seconds

Number of cars 1

bbox:car_number ((896, 400), (998, 478)) : 1

The minimum distance from car: 1 is 11.40175425099138

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 549/1261 [24:28<31:43, 2.67s/it]

The times for each task are: [0.854305, 0.674949, 1.581817, 0.423548, 0.384729] with:

Minimum: 0.384729 Maximum: 1.581817 Average: 0.7839 seconds

Number of cars 1

bbox:car_number ((896, 400), (994, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 550/1261 [24:30<31:40, 2.67s/it]

The times for each task are: [0.827852, 1.375587, 0.694934, 0.411624, 0.315523] with:

Minimum: 0.315523 Maximum: 1.375587 Average: 0.7251 seconds

Number of cars 1

bbox:car_number ((896, 400), (991, 472)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 551/1261 [24:32<31:37, 2.67s/it]

The times for each task are: [0.843146, 0.675025, 0.410881, 1.40307, 0.328912] with:

Minimum: 0.328912 Maximum: 1.40307 Average: 0.7322 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 472)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (989, 459))]

Length of task list: 5

Number of processes used: 3

44% | 552/1261 [24:35<31:35, 2.67s/it]

The times for each task are: [1.047845, 1.408136, 0.689744, 0.303209, 0.452185] with:

Minimum: 0.303209 Maximum: 1.408136 Average: 0.7802 seconds

Number of cars 1

bbox:car_number ((896, 400), (991, 472)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (989, 459))]

Length of task list: 5

Number of processes used: 3

44% | 553/1261 [24:38<31:32, 2.67s/it]

The times for each task are: [1.378016, 0.877684, 0.67411, 0.308089, 0.383854] with:

Minimum: 0.308089 Maximum: 1.378016 Average: 0.7244 seconds

Number of cars 2

bbox:car_number ((913, 400), (987, 472)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((989, 400), (991, 459)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (989, 459))]

Length of task list: 5

Number of processes used: 3

44% | 554/1261 [24:41<31:30, 2.67s/it]

The times for each task are: [0.901265, 0.617767, 1.709372, 0.304904, 0.407413] with:

Minimum: 0.304904 Maximum: 1.709372 Average: 0.7881 seconds

Number of cars 1

bbox:car_number ((932, 400), (987, 462)) : 1

The minimum distance from car: 1 is 10.295630140987

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 555/1261 [24:43<31:27, 2.67s/it]

The times for each task are: [0.586425, 0.904369, 1.42461, 0.422491, 0.3079] with:

Minimum: 0.3079 Maximum: 1.42461 Average: 0.7292 seconds

Number of cars 1

bbox:car_number ((896, 400), (977, 472)) : 1

The minimum distance from car: 1 is 23.53720459187964

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 556/1261 [24:46<31:25, 2.67s/it]

The times for each task are: [0.631564, 0.923105, 0.30917, 1.650313, 0.440771] with:

Minimum: 0.30917 Maximum: 1.650313 Average: 0.791 seconds

Number of cars 2

bbox:car_number ((913, 400), (991, 472)) : 1

The minimum distance from car: 1 is 16.0

bbox:car_number ((994, 400), (994, 472)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

```
44%|      | 557/1261 [24:49<31:22,  2.67s/it]
```

The times for each task are: [0.917038, 1.406779, 0.673819, 0.418776, 0.312217] with:

Minimum: 0.312217 Maximum: 1.406779 Average: 0.7457 seconds

Number of cars 1

bbox:car_number ((913, 400), (987, 462)) : 1

The minimum distance from car: 1 is 5.385164807134504

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

```
44%|      | 558/1261 [24:52<31:20,  2.67s/it]
```

The times for each task are: [0.84704, 0.705035, 0.418341, 1.447417, 0.335987] with:

Minimum: 0.335987 Maximum: 1.447417 Average: 0.7508 seconds

Number of cars 1

bbox:car_number ((913, 400), (991, 462)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

```
44%|      | 559/1261 [24:55<31:17,  2.67s/it]
```

The times for each task are: [0.852444, 1.425092, 0.699479, 0.309932, 0.388887] with:

Minimum: 0.309932 Maximum: 1.425092 Average: 0.7352 seconds

Number of cars 1

bbox:car_number ((905, 400), (1001, 472)) : 1

The minimum distance from car: 1 is 5.0990195135927845

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

44% | 560/1261 [24:57<31:15, 2.67s/it]

The times for each task are: [0.913439, 1.413256, 0.689397, 0.330786, 0.43665] with:

Minimum: 0.330786 Maximum: 1.413256 Average: 0.7567 seconds

Number of cars 1

bbox:car_number ((929, 400), (987, 459)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

44% | 561/1261 [25:00<31:12, 2.68s/it]

The times for each task are: [0.906142, 0.654071, 1.69276, 0.504891, 0.339291] with:

Minimum: 0.339291 Maximum: 1.69276 Average: 0.8194 seconds

Number of cars 1

bbox:car_number ((913, 400), (1001, 459)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

45% | 562/1261 [25:03<31:10, 2.68s/it]

The times for each task are: [1.39612, 1.108527, 0.710804, 0.386487, 0.327507] with:

Minimum: 0.327507 Maximum: 1.39612 Average: 0.7859 seconds

Number of cars 0

Length of task list: 5

Number of processes used: 3

45% | 563/1261 [25:06<31:07, 2.68s/it]

The times for each task are: [0.677329, 0.891902, 1.412683, 0.301713, 0.45128] with:

Minimum: 0.301713 Maximum: 1.412683 Average: 0.747 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (994, 472)) : 1
The minimum distance from car: 1 is 13.892443989449804
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

45% | 564/1261 [25:09<31:05, 2.68s/it]

The times for each task are: [0.620534, 0.978743, 1.335825, 0.411292, 0.394261] with:

Minimum: 0.394261 Maximum: 1.335825 Average: 0.7481 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (998, 477)) : 1
The minimum distance from car: 1 is 2.8284271247461903
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

45% | 565/1261 [25:12<31:02, 2.68s/it]

The times for each task are: [0.707845, 1.424177, 0.957283, 0.429073, 0.365585] with:

Minimum: 0.365585 Maximum: 1.424177 Average: 0.7768 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (998, 478)) : 1
The minimum distance from car: 1 is 4.123105625617661
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

45% | 566/1261 [25:14<30:59, 2.68s/it]

The times for each task are: [0.655624, 1.411386, 1.06835, 0.409152, 0.301113] with:

Minimum: 0.301113 Maximum: 1.411386 Average: 0.7691 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (998, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

45%| 567/1261 [25:17<30:57, 2.68s/it]

The times for each task are: [0.888152, 0.695733, 1.447796, 0.400374, 0.383177] with:
Minimum: 0.383177 Maximum: 1.447796 Average: 0.763 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

45%| 568/1261 [25:19<30:54, 2.68s/it]

The times for each task are: [0.892026, 0.653624, 1.46451, 0.49084, 0.353438] with:
Minimum: 0.353438 Maximum: 1.46451 Average: 0.7709 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 478)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

45%| 569/1261 [25:22<30:51, 2.68s/it]

The times for each task are: [0.634329, 0.908013, 1.429016, 0.415287, 0.372131] with:
Minimum: 0.372131 Maximum: 1.429016 Average: 0.7518 seconds

Number of cars 1

```
bbox:car_number ((913, 400), (994, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

45% | 570/1261 [25:25<30:48, 2.68s/it]

The times for each task are: [0.602337, 0.892679, 0.419596, 1.453523, 0.350116] with:
Minimum: 0.350116 Maximum: 1.453523 Average: 0.7437 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

45% | 571/1261 [25:27<30:45, 2.68s/it]

The times for each task are: [1.382901, 0.971777, 0.631527, 0.28914, 0.367009] with:
Minimum: 0.28914 Maximum: 1.382901 Average: 0.7285 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

45% | 572/1261 [25:30<30:43, 2.68s/it]

The times for each task are: [0.707978, 0.805728, 1.442288, 0.28686, 0.504442] with:
Minimum: 0.28686 Maximum: 1.442288 Average: 0.7495 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 478)) : 1
```

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

45% | 573/1261 [25:33<30:40, 2.68s/it]

The times for each task are: [1.662134, 0.976098, 0.647258, 0.318293, 0.549681] with:

Minimum: 0.318293 Maximum: 1.662134 Average: 0.8307 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

46% | 574/1261 [25:35<30:38, 2.68s/it]

The times for each task are: [1.430789, 0.887181, 0.690128, 0.406214, 0.313635] with:

Minimum: 0.313635 Maximum: 1.430789 Average: 0.7456 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

46% | 575/1261 [25:38<30:35, 2.68s/it]

The times for each task are: [0.990761, 1.460515, 0.641209, 0.484735, 0.322067] with:

Minimum: 0.322067 Maximum: 1.460515 Average: 0.7799 seconds

Number of cars 1

bbox:car_number ((913, 400), (991, 472)) : 1

The minimum distance from car: 1 is 5.0

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (994, 478))]
Length of task list: 5
Number of processes used: 3
```

46%| 576/1261 [25:41<30:33, 2.68s/it]

The times for each task are: [0.626725, 1.454268, 1.017527, 0.406392, 0.299733] with:

Minimum: 0.299733 Maximum: 1.454268 Average: 0.7609 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (994, 478)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (994, 478))]
Length of task list: 5
Number of processes used: 3
```

46%| 577/1261 [25:44<30:30, 2.68s/it]

The times for each task are: [0.853242, 0.696894, 1.411784, 0.406067, 0.385767] with:

Minimum: 0.385767 Maximum: 1.411784 Average: 0.7508 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (994, 478))]
Length of task list: 5
Number of processes used: 3
```

46%| 578/1261 [25:46<30:27, 2.68s/it]

The times for each task are: [0.616158, 1.004744, 1.619859, 0.317069, 0.410816] with:

Minimum: 0.317069 Maximum: 1.619859 Average: 0.7937 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (994, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 579/1261 [25:49<30:25, 2.68s/it]
```

```
The times for each task are: [1.442931, 1.077246, 0.623851, 0.424956, 0.37708] with:
```

```
Minimum: 0.37708 Maximum: 1.442931 Average: 0.7892 seconds
```

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (913, 400))]
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 580/1261 [25:52<30:22, 2.68s/it]
```

```
The times for each task are: [0.947293, 1.399957, 0.623422, 0.353122, 0.406025] with:
```

```
Minimum: 0.353122 Maximum: 1.399957 Average: 0.746 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 581/1261 [25:54<30:19, 2.68s/it]
```

```
The times for each task are: [0.572931, 1.41566, 0.403526, 1.056368, 0.29703] with:
```

```
Minimum: 0.29703 Maximum: 1.41566 Average: 0.7491 seconds
```

```
Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 582/1261 [25:57<30:17, 2.68s/it]
```

```
The times for each task are: [0.757494, 0.883824, 1.42587, 0.292262, 0.463905] with:
```

```
Minimum: 0.292262 Maximum: 1.42587 Average: 0.7647 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 583/1261 [26:00<30:14, 2.68s/it]
```

```
The times for each task are: [0.578015, 1.029323, 1.450996, 0.351345, 0.404471] with:
```

```
Minimum: 0.351345 Maximum: 1.450996 Average: 0.7628 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
46%|      | 584/1261 [26:03<30:11, 2.68s/it]
```

```
The times for each task are: [0.638977, 0.99046, 1.405314, 0.307656, 0.398778] with:
```

```
Minimum: 0.307656 Maximum: 1.405314 Average: 0.7482 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

46% | 585/1261 [26:06<30:09, 2.68s/it]

The times for each task are: [0.685107, 0.888215, 1.622723, 0.393895, 0.285465] with:

Minimum: 0.285465 Maximum: 1.622723 Average: 0.7751 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

46% | 586/1261 [26:08<30:06, 2.68s/it]

The times for each task are: [0.843617, 1.361396, 0.649363, 0.418108, 0.298456] with:

Minimum: 0.298456 Maximum: 1.361396 Average: 0.7142 seconds

Number of cars 1

bbox:car_number ((910, 400), (991, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 587/1261 [26:11<30:04, 2.68s/it]

The times for each task are: [0.634492, 0.88509, 1.411134, 0.314588, 0.510276] with:

Minimum: 0.314588 Maximum: 1.411134 Average: 0.7511 seconds

Number of cars 1

bbox:car_number ((913, 400), (991, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 588/1261 [26:14<30:01, 2.68s/it]

The times for each task are: [0.632068, 0.918869, 1.499585, 0.416494, 0.320729] with:

Minimum: 0.320729 Maximum: 1.499585 Average: 0.7575 seconds

Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

47% | 589/1261 [26:16<29:58, 2.68s/it]

The times for each task are: [1.065236, 0.638156, 1.447072, 0.454909, 0.300539] with:

Minimum: 0.300539 Maximum: 1.447072 Average: 0.7812 seconds

Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

47% | 590/1261 [26:19<29:56, 2.68s/it]

The times for each task are: [0.882129, 0.627591, 1.625913, 0.387326, 0.299632] with:

Minimum: 0.299632 Maximum: 1.625913 Average: 0.7645 seconds

Number of cars 1
bbox:car_number ((932, 400), (991, 472)) : 1
The minimum distance from car: 1 is 13.341664064126334
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

47% | 591/1261 [26:21<29:52, 2.68s/it]

The times for each task are: [0.618269, 1.396129, 0.902056, 0.380731, 0.331272] with:

Minimum: 0.331272 Maximum: 1.396129 Average: 0.7257 seconds

Number of cars 1

bbox:car_number ((913, 400), (991, 472)) : 1

The minimum distance from car: 1 is 9.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

47% | 592/1261 [26:24<29:50, 2.68s/it]

The times for each task are: [0.5673, 1.008194, 1.38744, 0.384573, 0.37973] with:

Minimum: 0.37973 Maximum: 1.38744 Average: 0.7454 seconds

Number of cars 1

bbox:car_number ((913, 400), (991, 477)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

47% | 593/1261 [26:27<29:47, 2.68s/it]

The times for each task are: [1.455024, 1.026741, 0.742989, 0.51379, 0.333562] with:

Minimum: 0.333562 Maximum: 1.455024 Average: 0.8144 seconds

Number of cars 1

bbox:car_number ((905, 400), (994, 485)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

47% | 594/1261 [26:29<29:44, 2.68s/it]

The times for each task are: [0.848395, 0.722697, 1.386912, 0.406613, 0.349992] with:

Minimum: 0.349992 Maximum: 1.386912 Average: 0.7429 seconds

Number of cars 2

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 3.1622776601683795

bbox:car_number ((896, 419), (896, 472)) : 2

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 595/1261 [26:32<29:42, 2.68s/it]

The times for each task are: [1.361362, 0.892697, 0.680745, 0.392193, 0.295407] with:

Minimum: 0.295407 Maximum: 1.361362 Average: 0.7245 seconds

Number of cars 1

bbox:car_number ((910, 400), (991, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 596/1261 [26:35<29:39, 2.68s/it]

The times for each task are: [0.882599, 1.388364, 0.725666, 0.357351, 0.405769] with:

Minimum: 0.357351 Maximum: 1.388364 Average: 0.7519 seconds

Number of cars 1

bbox:car_number ((905, 400), (991, 478)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

47% | 597/1261 [26:37<29:36, 2.68s/it]

The times for each task are: [0.584607, 1.414929, 0.91719, 0.405632, 0.352821] with:

Minimum: 0.352821 Maximum: 1.414929 Average: 0.735 seconds

Number of cars 2

bbox:car_number ((886, 400), (896, 472)) : 1

bbox:car_number ((905, 400), (991, 478)) : 2

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

47% | 598/1261 [26:40<29:34, 2.68s/it]

The times for each task are: [0.923594, 1.384061, 0.715483, 0.296364, 0.383068] with:

Minimum: 0.296364 Maximum: 1.384061 Average: 0.7405 seconds

Number of cars 1

bbox:car_number ((882, 400), (991, 485)) : 1

The minimum distance from car: 1 is 12.36931687685298

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

48% | 599/1261 [26:43<29:31, 2.68s/it]

The times for each task are: [0.641383, 0.901462, 1.413955, 0.38773, 0.297866] with:

Minimum: 0.297866 Maximum: 1.413955 Average: 0.7285 seconds

Number of cars 1

bbox:car_number ((896, 400), (991, 485)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

48% | 600/1261 [26:46<29:29, 2.68s/it]

The times for each task are: [0.671769, 0.903185, 0.402551, 1.531869, 0.279712] with:

Minimum: 0.279712 Maximum: 1.531869 Average: 0.7578 seconds

Number of cars 1

bbox:car_number ((886, 400), (994, 495)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

48% | 601/1261 [26:48<29:26, 2.68s/it]

The times for each task are: [1.100411, 0.661296, 1.443525, 0.442369, 0.289794] with:

Minimum: 0.289794 Maximum: 1.443525 Average: 0.7875 seconds

Number of cars 1

bbox:car_number ((896, 400), (991, 485)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

48% | 602/1261 [26:51<29:23, 2.68s/it]

The times for each task are: [0.882634, 0.718483, 1.394072, 0.390525, 0.355106] with:

Minimum: 0.355106 Maximum: 1.394072 Average: 0.7482 seconds

Number of cars 1

bbox:car_number ((886, 400), (991, 485)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

48% | 603/1261 [26:53<29:20, 2.68s/it]

The times for each task are: [0.945284, 0.64832, 1.389792, 0.280145, 0.405098] with:

Minimum: 0.280145 Maximum: 1.389792 Average: 0.7337 seconds

Number of cars 1
bbox:car_number ((894, 400), (994, 485)) : 1
The minimum distance from car: 1 is 6.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

48% | 604/1261 [26:56<29:18, 2.68s/it]

The times for each task are: [0.914303, 1.364326, 0.733882, 0.387128, 0.339034] with:

Minimum: 0.339034 Maximum: 1.364326 Average: 0.7477 seconds

Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

48% | 605/1261 [26:59<29:15, 2.68s/it]

The times for each task are: [0.678365, 1.580088, 0.842214, 0.353243, 0.471311] with:

Minimum: 0.353243 Maximum: 1.580088 Average: 0.785 seconds

Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

48% | 606/1261 [27:01<29:13, 2.68s/it]

The times for each task are: [0.898595, 1.406536, 0.779562, 0.46689, 0.314644] with:

Minimum: 0.314644 Maximum: 1.406536 Average: 0.7732 seconds

```
Number of cars 1
bbox:car_number ((913, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

48% | 607/1261 [27:04<29:10, 2.68s/it]

The times for each task are: [0.675722, 1.425701, 1.01291, 0.302896, 0.435062] with:

Minimum: 0.302896 Maximum: 1.425701 Average: 0.7705 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 485)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

48% | 608/1261 [27:07<29:07, 2.68s/it]

The times for each task are: [0.866299, 1.394271, 0.749828, 0.389315, 0.293141] with:

Minimum: 0.293141 Maximum: 1.394271 Average: 0.7386 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

48% | 609/1261 [27:09<29:05, 2.68s/it]

The times for each task are: [0.893488, 1.381277, 0.701063, 0.297699, 0.395658] with:

Minimum: 0.297699 Maximum: 1.381277 Average: 0.7338 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 477)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

48%| 610/1261 [27:12<29:02, 2.68s/it]

The times for each task are: [0.862938, 0.651575, 1.524395, 0.479175, 0.299136] with:
Minimum: 0.299136 Maximum: 1.524395 Average: 0.7634 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

48%| 611/1261 [27:15<28:59, 2.68s/it]

The times for each task are: [0.975523, 0.750737, 1.397998, 0.322709, 0.3895] with:
Minimum: 0.322709 Maximum: 1.397998 Average: 0.7673 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (991, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400))]
Length of task list: 5
Number of processes used: 3
```

49%| 612/1261 [27:18<28:57, 2.68s/it]

The times for each task are: [0.881196, 0.385888, 1.382891, 0.700167, 0.390596] with:
Minimum: 0.385888 Maximum: 1.382891 Average: 0.7481 seconds

Number of cars 1

```
bbox:car_number ((910, 400), (987, 478)) : 1
```

```
The minimum distance from car: 1 is 2.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
49% | 613/1261 [27:20<28:54, 2.68s/it]
```

```
The times for each task are: [0.866776, 0.61355, 1.349885, 0.396644, 0.437737] with:
```

```
Minimum: 0.396644 Maximum: 1.349885 Average: 0.7329 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((910, 400), (987, 478)) : 1
```

```
The minimum distance from car: 1 is 0.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
49% | 614/1261 [27:23<28:51, 2.68s/it]
```

```
The times for each task are: [0.884924, 0.641709, 1.426998, 0.419543, 0.386507] with:
```

```
Minimum: 0.386507 Maximum: 1.426998 Average: 0.7519 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((905, 400), (987, 478)) : 1
```

```
The minimum distance from car: 1 is 2.0
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
49% | 615/1261 [27:26<28:49, 2.68s/it]
```

```
The times for each task are: [0.911988, 1.400702, 0.612092, 0.309743, 0.387638] with:
```

```
Minimum: 0.309743 Maximum: 1.400702 Average: 0.7244 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((905, 400), (987, 478)) : 1
```

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

49% | 616/1261 [27:29<28:46, 2.68s/it]

The times for each task are: [0.633251, 0.907193, 0.358338, 0.494433, 1.630168] with:

Minimum: 0.358338 Maximum: 1.630168 Average: 0.8047 seconds

Number of cars 1

bbox:car_number ((905, 400), (987, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

49% | 617/1261 [27:31<28:43, 2.68s/it]

The times for each task are: [0.599378, 1.467622, 0.956337, 0.400543, 0.408068] with:

Minimum: 0.400543 Maximum: 1.467622 Average: 0.7664 seconds

Number of cars 1

bbox:car_number ((905, 400), (987, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

49% | 618/1261 [27:34<28:41, 2.68s/it]

The times for each task are: [0.916193, 0.645575, 1.464292, 0.320301, 0.404561] with:

Minimum: 0.320301 Maximum: 1.464292 Average: 0.7502 seconds

Number of cars 1

bbox:car_number ((905, 400), (987, 478)) : 1

The minimum distance from car: 1 is 0.0

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

49%| 619/1261 [27:37<28:38, 2.68s/it]

The times for each task are: [0.954241, 0.689245, 1.462385, 0.369936, 0.412503] with:

Minimum: 0.369936 Maximum: 1.462385 Average: 0.7777 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (987, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

49%| 620/1261 [27:40<28:36, 2.68s/it]

The times for each task are: [0.936571, 0.641626, 1.461727, 0.401457, 0.383309] with:

Minimum: 0.383309 Maximum: 1.461727 Average: 0.7649 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (991, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400))]
Length of task list: 5
Number of processes used: 3
```

49%| 621/1261 [27:42<28:33, 2.68s/it]

The times for each task are: [1.06198, 1.389819, 0.626153, 0.474054, 0.291013] with:

Minimum: 0.291013 Maximum: 1.389819 Average: 0.7686 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (977, 478)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (905, 400), (977, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

49% | 622/1261 [27:45<28:30, 2.68s/it]

The times for each task are: [0.884448, 1.371449, 0.765982, 0.410598, 0.291528] with:

Minimum: 0.291528 Maximum: 1.371449 Average: 0.7448 seconds

```
Number of cars 1
bbox:car_number ((905, 400), (977, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (977, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

49% | 623/1261 [27:47<28:27, 2.68s/it]

The times for each task are: [0.937054, 0.675165, 1.631364, 0.426524, 0.400136] with:

Minimum: 0.400136 Maximum: 1.631364 Average: 0.814 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (977, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (977, 478)) : 1
Length of task list: 5
Number of processes used: 3
```

49% | 624/1261 [27:50<28:25, 2.68s/it]

The times for each task are: [0.853723, 0.408559, 0.617358, 1.409389, 0.303466] with:

Minimum: 0.303466 Maximum: 1.409389 Average: 0.7185 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (977, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (910, 400), (977, 478)) : 1
```

```
Length of task list: 5
Number of processes used: 3
```

```
50%| 625/1261 [27:53<28:22, 2.68s/it]
```

```
The times for each task are: [0.601006, 1.064195, 1.413624, 0.407385, 0.451832] with:
```

```
Minimum: 0.407385 Maximum: 1.413624 Average: 0.7876 seconds
```

```
Number of cars 1
bbox:car_number ((910, 415), (972, 478)) : 1
The minimum distance from car: 1 is 7.280109889280518
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
50%| 626/1261 [27:56<28:20, 2.68s/it]
```

```
The times for each task are: [0.889414, 0.636942, 1.371174, 0.409841, 0.286809] with:
```

```
Minimum: 0.286809 Maximum: 1.371174 Average: 0.7188 seconds
```

```
Number of cars 1
bbox:car_number ((910, 415), (977, 478)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
50%| 627/1261 [27:58<28:17, 2.68s/it]
```

```
The times for each task are: [0.996119, 0.652042, 1.682836, 0.524977, 0.387801] with:
```

```
Minimum: 0.387801 Maximum: 1.682836 Average: 0.8488 seconds
```

```
Number of cars 1
bbox:car_number ((905, 400), (987, 478)) : 1
The minimum distance from car: 1 is 7.615773105863909
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

50% | 628/1261 [28:01<28:14, 2.68s/it]

The times for each task are: [0.84842, 1.404668, 0.724091, 0.428969, 0.405855] with:

Minimum: 0.405855 Maximum: 1.404668 Average: 0.7624 seconds

Number of cars 1

bbox:car_number ((910, 415), (972, 477)) : 1

The minimum distance from car: 1 is 8.602325267042627

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

50% | 629/1261 [28:03<28:11, 2.68s/it]

The times for each task are: [0.925787, 0.6213, 1.440616, 0.302238, 0.413494] with:

Minimum: 0.302238 Maximum: 1.440616 Average: 0.7407 seconds

Number of cars 1

bbox:car_number ((910, 400), (972, 477)) : 1

The minimum distance from car: 1 is 8.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

50% | 630/1261 [28:06<28:09, 2.68s/it]

The times for each task are: [0.629987, 0.902662, 1.392968, 0.305167, 0.412142] with:

Minimum: 0.305167 Maximum: 1.392968 Average: 0.7286 seconds

Number of cars 1

bbox:car_number ((896, 400), (972, 472)) : 1

The minimum distance from car: 1 is 7.280109889280518

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

50%| 631/1261 [28:09<28:06, 2.68s/it]

The times for each task are: [0.846987, 0.645005, 0.403355, 1.632083, 0.288586] with:

Minimum: 0.288586 Maximum: 1.632083 Average: 0.7632 seconds

Number of cars 1
bbox:car_number ((896, 400), (972, 472)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

50%| 632/1261 [28:12<28:03, 2.68s/it]

The times for each task are: [0.568375, 1.013353, 1.482639, 0.443276, 0.35328] with:

Minimum: 0.35328 Maximum: 1.482639 Average: 0.7722 seconds

Number of cars 1
bbox:car_number ((913, 400), (972, 462)) : 1
The minimum distance from car: 1 is 9.433981132056603
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

50%| 633/1261 [28:14<28:01, 2.68s/it]

The times for each task are: [0.920343, 1.45331, 0.675106, 0.410107, 0.29139] with:

Minimum: 0.29139 Maximum: 1.45331 Average: 0.7501 seconds

Number of cars 1
bbox:car_number ((896, 400), (972, 478)) : 1
The minimum distance from car: 1 is 11.313708498984761
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

50% | 634/1261 [28:17<27:58, 2.68s/it]

The times for each task are: [0.889409, 1.360422, 0.740045, 0.430673, 0.386599] with:

Minimum: 0.386599 Maximum: 1.360422 Average: 0.7614 seconds

Number of cars 1

bbox:car_number ((895, 400), (972, 477)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

50% | 635/1261 [28:19<27:55, 2.68s/it]

The times for each task are: [0.62708, 1.027466, 1.455541, 0.405849, 0.36661] with:

Minimum: 0.36661 Maximum: 1.455541 Average: 0.7765 seconds

Number of cars 1

bbox:car_number ((894, 400), (972, 478)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

50% | 636/1261 [28:22<27:52, 2.68s/it]

The times for each task are: [0.574845, 0.991504, 1.59807, 0.307551, 0.439499] with:

Minimum: 0.307551 Maximum: 1.59807 Average: 0.7823 seconds

Number of cars 1

bbox:car_number ((895, 415), (972, 477)) : 1

The minimum distance from car: 1 is 7.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

51% | 637/1261 [28:24<27:49, 2.68s/it]

The times for each task are: [0.851384, 0.620718, 1.418926, 0.300236, 0.384678] with:

Minimum: 0.300236 Maximum: 1.418926 Average: 0.7152 seconds

Number of cars 1

bbox:car_number ((895, 400), (972, 492)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

51% | 638/1261 [28:27<27:47, 2.68s/it]

The times for each task are: [0.632619, 0.893707, 0.392191, 1.673973, 0.393637] with:

Minimum: 0.392191 Maximum: 1.673973 Average: 0.7972 seconds

Number of cars 1

bbox:car_number ((886, 400), (972, 495)) : 1

The minimum distance from car: 1 is 4.123105625617661

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

51% | 639/1261 [28:30<27:44, 2.68s/it]

The times for each task are: [0.615918, 0.883748, 0.406038, 0.291193, 1.372836] with:

Minimum: 0.291193 Maximum: 1.372836 Average: 0.7139 seconds

Number of cars 1

bbox:car_number ((886, 415), (972, 485)) : 1

The minimum distance from car: 1 is 3.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

51% | 640/1261 [28:32<27:41, 2.68s/it]

The times for each task are: [0.892443, 1.361317, 0.763311, 0.420826, 0.386119] with:

Minimum: 0.386119 Maximum: 1.361317 Average: 0.7648 seconds

Number of cars 1
bbox:car_number ((894, 400), (972, 478)) : 1
The minimum distance from car: 1 is 11.704699910719626
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

51% | 641/1261 [28:35<27:39, 2.68s/it]

The times for each task are: [0.880966, 0.709437, 1.403249, 0.410791, 0.319359] with:

Minimum: 0.319359 Maximum: 1.403249 Average: 0.7448 seconds

Number of cars 1
bbox:car_number ((894, 400), (972, 477)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

51% | 642/1261 [28:37<27:36, 2.68s/it]

The times for each task are: [0.938053, 0.596976, 0.442112, 1.704063, 0.385198] with:

Minimum: 0.385198 Maximum: 1.704063 Average: 0.8133 seconds

Number of cars 1
bbox:car_number ((895, 400), (972, 472)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

51% | 643/1261 [28:40<27:33, 2.68s/it]

The times for each task are: [1.448325, 1.039207, 0.623444, 0.410657, 0.373729] with:

Minimum: 0.373729 Maximum: 1.448325 Average: 0.7791 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (972, 477)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (971, 477))]
Length of task list: 5
Number of processes used: 3
```

51% | 644/1261 [28:43<27:31, 2.68s/it]

The times for each task are: [0.972979, 1.503244, 0.695937, 0.412506, 0.301886] with:

Minimum: 0.301886 Maximum: 1.503244 Average: 0.7773 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (971, 477)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (971, 477))]
Length of task list: 5
Number of processes used: 3
```

51% | 645/1261 [28:46<27:28, 2.68s/it]

The times for each task are: [0.932927, 0.63016, 1.397151, 0.418319, 0.389611] with:

Minimum: 0.389611 Maximum: 1.397151 Average: 0.7536 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (957, 477)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (957, 477))]
Length of task list: 5
Number of processes used: 3
```

51% | 646/1261 [28:48<27:25, 2.68s/it]

The times for each task are: [1.401867, 0.858608, 0.631075, 0.352301, 0.526183] with:

Minimum: 0.352301 Maximum: 1.401867 Average: 0.754 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (963, 478)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (963, 478))]
Length of task list: 5
Number of processes used: 3
```

51%| 647/1261 [28:50<27:22, 2.68s/it]

The times for each task are: [0.642412, 1.436467, 0.935721, 0.391108, 0.294712] with:
Minimum: 0.294712 Maximum: 1.436467 Average: 0.7401 seconds

```
Number of cars 1
bbox:car_number ((886, 400), (963, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (963, 478))]
Length of task list: 5
Number of processes used: 3
```

51%| 648/1261 [28:53<27:20, 2.68s/it]

The times for each task are: [0.60491, 1.471369, 1.105167, 0.346504, 0.501617] with:
Minimum: 0.346504 Maximum: 1.471369 Average: 0.8059 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (957, 478)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (957, 478))]
Length of task list: 5
Number of processes used: 3
```

51%| 649/1261 [28:56<27:17, 2.68s/it]

The times for each task are: [0.633988, 0.921541, 1.660689, 0.314255, 0.39383] with:
Minimum: 0.314255 Maximum: 1.660689 Average: 0.7849 seconds

Number of cars 1

bbox:car_number ((886, 400), (957, 478)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (894, 415))]

Length of task list: 5

Number of processes used: 3

52% | 650/1261 [28:58<27:14, 2.68s/it]

The times for each task are: [0.691728, 0.927857, 1.480558, 0.483954, 0.360245] with:

Minimum: 0.360245 Maximum: 1.480558 Average: 0.7889 seconds

Number of cars 1

bbox:car_number ((894, 415), (954, 472)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (894, 415))]

Length of task list: 5

Number of processes used: 3

52% | 651/1261 [29:01<27:11, 2.68s/it]

The times for each task are: [1.414934, 0.872445, 0.628242, 0.297287, 0.378147] with:

Minimum: 0.297287 Maximum: 1.414934 Average: 0.7182 seconds

Number of cars 1

bbox:car_number ((894, 400), (957, 472)) : 1

The minimum distance from car: 1 is 7.0710678118654755

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (894, 415))]

Length of task list: 5

Number of processes used: 3

52% | 652/1261 [29:04<27:09, 2.68s/it]

The times for each task are: [0.657144, 1.41465, 0.84557, 0.314138, 0.440289] with:

Minimum: 0.314138 Maximum: 1.41465 Average: 0.7344 seconds

Number of cars 1

bbox:car_number ((894, 400), (954, 472)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

52% | 653/1261 [29:07<27:06, 2.68s/it]

The times for each task are: [1.363024, 1.008617, 0.638572, 0.422266, 0.354354] with:

Minimum: 0.354354 Maximum: 1.363024 Average: 0.7574 seconds

Number of cars 1

bbox:car_number ((886, 400), (954, 472)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

52% | 654/1261 [29:09<27:03, 2.68s/it]

The times for each task are: [0.866173, 1.454209, 0.621992, 0.394453, 0.358143] with:

Minimum: 0.358143 Maximum: 1.454209 Average: 0.739 seconds

Number of cars 1

bbox:car_number ((894, 415), (953, 462)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

52% | 655/1261 [29:12<27:00, 2.67s/it]

The times for each task are: [0.575547, 1.365297, 1.015674, 0.294642, 0.399757] with:

Minimum: 0.294642 Maximum: 1.365297 Average: 0.7302 seconds

Number of cars 1

bbox:car_number ((894, 415), (957, 477)) : 1

The minimum distance from car: 1 is 8.246211251235321

```
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

52%| 656/1261 [29:14<26:58, 2.68s/it]

The times for each task are: [0.914941, 0.628348, 1.492773, 0.322543, 0.421608] with:

Minimum: 0.322543 Maximum: 1.492773 Average: 0.756 seconds

```
Number of cars 1
bbox:car_number ((886, 400), (954, 472)) : 1
The minimum distance from car: 1 is 11.180339887498949
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

52%| 657/1261 [29:17<26:55, 2.67s/it]

The times for each task are: [1.389842, 0.850861, 0.603621, 0.289155, 0.38902] with:

Minimum: 0.289155 Maximum: 1.389842 Average: 0.7045 seconds

```
Number of cars 1
bbox:car_number ((882, 400), (954, 477)) : 1
The minimum distance from car: 1 is 2.8284271247461903
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

52%| 658/1261 [29:20<26:52, 2.67s/it]

The times for each task are: [1.379104, 0.626995, 0.874206, 0.323469, 0.463263] with:

Minimum: 0.323469 Maximum: 1.379104 Average: 0.7334 seconds

```
Number of cars 1
bbox:car_number ((880, 400), (957, 477)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

52% | 659/1261 [29:22<26:50, 2.68s/it]

The times for each task are: [0.996311, 0.61104, 1.458785, 0.472015, 0.288827] with:

Minimum: 0.288827 Maximum: 1.458785 Average: 0.7654 seconds

```
Number of cars 1  
bbox:car_number ((875, 400), (957, 478)) : 1  
The minimum distance from car: 1 is 2.23606797749979  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

52% | 660/1261 [29:25<26:47, 2.68s/it]

The times for each task are: [0.856255, 1.302037, 0.741809, 0.427453, 0.335858] with:

Minimum: 0.335858 Maximum: 1.302037 Average: 0.7327 seconds

```
Number of cars 1  
bbox:car_number ((875, 400), (957, 478)) : 1  
The minimum distance from car: 1 is 0.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]  
Length of task list: 5  
Number of processes used: 3
```

52% | 661/1261 [29:28<26:45, 2.68s/it]

The times for each task are: [1.37964, 0.826863, 0.686814, 0.298902, 0.374078] with:

Minimum: 0.298902 Maximum: 1.37964 Average: 0.7133 seconds

```
Number of cars 1  
bbox:car_number ((880, 400), (957, 478)) : 1  
The minimum distance from car: 1 is 2.0  
totalCars: 2  
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
52%|     | 662/1261 [29:31<26:42, 2.68s/it]
```

```
The times for each task are: [0.83134, 0.624474, 1.551105, 0.406886, 0.295637] with:
```

```
Minimum: 0.295637 Maximum: 1.551105 Average: 0.7419 seconds
```

```
Number of cars 1
bbox:car_number ((880, 400), (957, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
53%|     | 663/1261 [29:33<26:39, 2.68s/it]
```

```
The times for each task are: [0.941967, 1.483991, 0.666567, 0.308744, 0.405237] with:
```

```
Minimum: 0.308744 Maximum: 1.483991 Average: 0.7613 seconds
```

```
Number of cars 1
bbox:car_number ((894, 400), (957, 477)) : 1
The minimum distance from car: 1 is 7.0710678118654755
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

```
53%|     | 664/1261 [29:36<26:37, 2.68s/it]
```

```
The times for each task are: [0.619954, 1.370523, 1.023237, 0.388731, 0.389322] with:
```

```
Minimum: 0.388731 Maximum: 1.370523 Average: 0.7584 seconds
```

```
Number of cars 1
bbox:car_number ((882, 400), (954, 477)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
```

Number of processes used: 3

53% | 665/1261 [29:39<26:34, 2.68s/it]

The times for each task are: [1.015919, 0.641294, 1.436403, 0.528714, 0.36607] with:

Minimum: 0.36607 Maximum: 1.436403 Average: 0.7977 seconds

Number of cars 1

bbox:car_number ((880, 400), (954, 478)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53% | 666/1261 [29:41<26:32, 2.68s/it]

The times for each task are: [0.981821, 0.587957, 1.489036, 0.287071, 0.491543] with:

Minimum: 0.287071 Maximum: 1.489036 Average: 0.7675 seconds

Number of cars 1

bbox:car_number ((880, 400), (954, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53% | 667/1261 [29:44<26:29, 2.68s/it]

The times for each task are: [0.901713, 0.572323, 1.549051, 0.412729, 0.403108] with:

Minimum: 0.403108 Maximum: 1.549051 Average: 0.7678 seconds

Number of cars 1

bbox:car_number ((880, 400), (954, 477)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53%| 668/1261 [29:47<26:26, 2.68s/it]

The times for each task are: [0.932437, 0.683546, 1.373214, 0.394369, 0.347809] with:

Minimum: 0.347809 Maximum: 1.373214 Average: 0.7463 seconds

Number of cars 1

bbox:car_number ((880, 400), (953, 478)) : 1

The minimum distance from car: 1 is 1.4142135623730951

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53%| 669/1261 [29:49<26:23, 2.68s/it]

The times for each task are: [0.696199, 0.894515, 1.382222, 0.292537, 0.456359] with:

Minimum: 0.292537 Maximum: 1.382222 Average: 0.7444 seconds

Number of cars 1

bbox:car_number ((875, 400), (953, 472)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53%| 670/1261 [29:52<26:21, 2.68s/it]

The times for each task are: [0.882239, 1.400604, 0.672816, 0.394367, 0.343436] with:

Minimum: 0.343436 Maximum: 1.400604 Average: 0.7387 seconds

Number of cars 1

bbox:car_number ((880, 400), (944, 472)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

53% | 671/1261 [29:55<26:18, 2.68s/it]

The times for each task are: [0.680124, 1.342032, 1.022719, 0.451726, 0.357024] with:

Minimum: 0.357024 Maximum: 1.342032 Average: 0.7707 seconds

Number of cars 1

bbox:car_number ((875, 400), (953, 477)) : 1

The minimum distance from car: 1 is 2.8284271247461903

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

53% | 672/1261 [29:57<26:15, 2.67s/it]

The times for each task are: [1.316714, 0.610619, 1.068747, 0.294396, 0.405332] with:

Minimum: 0.294396 Maximum: 1.316714 Average: 0.7392 seconds

Number of cars 1

bbox:car_number ((880, 400), (942, 472)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

53% | 673/1261 [30:00<26:12, 2.67s/it]

The times for each task are: [0.564971, 0.868677, 1.516262, 0.398129, 0.347093] with:

Minimum: 0.347093 Maximum: 1.516262 Average: 0.739 seconds

Number of cars 1

bbox:car_number ((880, 400), (953, 472)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

53% | 674/1261 [30:02<26:10, 2.67s/it]

The times for each task are: [0.562204, 0.895089, 1.432029, 0.390062, 0.305459] with:

Minimum: 0.305459 Maximum: 1.432029 Average: 0.717 seconds

Number of cars 1

bbox:car_number ((865, 400), (944, 472)) : 1

The minimum distance from car: 1 is 12.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

54% | 675/1261 [30:05<26:07, 2.68s/it]

The times for each task are: [0.871882, 1.393672, 0.706991, 0.502205, 0.275388] with:

Minimum: 0.275388 Maximum: 1.393672 Average: 0.75 seconds

Number of cars 1

bbox:car_number ((865, 400), (944, 472)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

54% | 676/1261 [30:08<26:05, 2.68s/it]

The times for each task are: [0.619439, 1.072281, 0.401474, 1.39169, 0.332096] with:

Minimum: 0.332096 Maximum: 1.39169 Average: 0.7634 seconds

Number of cars 1

bbox:car_number ((880, 400), (944, 477)) : 1

The minimum distance from car: 1 is 8.246211251235321

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

54% | 677/1261 [30:11<26:02, 2.68s/it]

The times for each task are: [0.935099, 0.649861, 1.482608, 0.304589, 0.530881] with:

Minimum: 0.304589 Maximum: 1.482608 Average: 0.7806 seconds

Number of cars 1
bbox:car_number ((865, 400), (944, 477)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

54% | 678/1261 [30:14<25:59, 2.68s/it]

The times for each task are: [0.594758, 1.115897, 1.456252, 0.377166, 0.377644] with:

Minimum: 0.377166 Maximum: 1.456252 Average: 0.7843 seconds

Number of cars 1
bbox:car_number ((865, 400), (953, 477)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

54% | 679/1261 [30:16<25:57, 2.68s/it]

The times for each task are: [1.373139, 0.611302, 0.982089, 0.279614, 0.394557] with:

Minimum: 0.279614 Maximum: 1.373139 Average: 0.7281 seconds

Number of cars 1
bbox:car_number ((880, 400), (944, 472)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3

54% | 680/1261 [30:19<25:54, 2.68s/it]

The times for each task are: [1.36454, 0.853791, 0.604029, 0.471409, 0.300391] with:

Minimum: 0.300391 Maximum: 1.36454 Average: 0.7188 seconds

```
Number of cars 1
bbox:car_number ((880, 400), (942, 472)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54% | 681/1261 [30:22<25:51, 2.68s/it]

The times for each task are: [0.679068, 1.376886, 0.867997, 0.291477, 0.381848] with:

Minimum: 0.291477 Maximum: 1.376886 Average: 0.7195 seconds

```
Number of cars 1
bbox:car_number ((880, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54% | 682/1261 [30:24<25:49, 2.68s/it]

The times for each task are: [0.603477, 1.019969, 1.347662, 0.384317, 0.338975] with:

Minimum: 0.338975 Maximum: 1.347662 Average: 0.7389 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (944, 472)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54% | 683/1261 [30:27<25:46, 2.68s/it]

The times for each task are: [1.300626, 0.94309, 0.700921, 0.405354, 0.286732] with:

Minimum: 0.286732 Maximum: 1.300626 Average: 0.7273 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (953, 478)) : 1
The minimum distance from car: 1 is 5.830951894845301
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54%| 684/1261 [30:30<25:43, 2.68s/it]

The times for each task are: [0.895838, 1.354608, 0.629325, 0.334814, 0.392497] with:
Minimum: 0.334814 Maximum: 1.354608 Average: 0.7214 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54%| 685/1261 [30:32<25:40, 2.68s/it]

The times for each task are: [1.021898, 0.669817, 0.335201, 0.471035, 1.664928] with:
Minimum: 0.335201 Maximum: 1.664928 Average: 0.8326 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]
Length of task list: 5
Number of processes used: 3
```

54%| 686/1261 [30:34<25:37, 2.67s/it]

The times for each task are: [0.599916, 0.932185, 1.361881, 0.29386, 0.377256] with:
Minimum: 0.29386 Maximum: 1.361881 Average: 0.713 seconds

Number of cars 1

bbox:car_number ((875, 400), (942, 472)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

54% | 687/1261 [30:37<25:35, 2.67s/it]

The times for each task are: [0.944332, 1.426139, 0.665968, 0.443155, 0.281293] with:

Minimum: 0.281293 Maximum: 1.426139 Average: 0.7522 seconds

Number of cars 1

bbox:car_number ((880, 400), (942, 472)) : 1

The minimum distance from car: 1 is 3.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

55% | 688/1261 [30:40<25:32, 2.67s/it]

The times for each task are: [0.887118, 0.639808, 1.412776, 0.484443, 0.298932] with:

Minimum: 0.298932 Maximum: 1.412776 Average: 0.7446 seconds

Number of cars 1

bbox:car_number ((875, 400), (942, 472)) : 1

The minimum distance from car: 1 is 3.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (1133, 495))]

Length of task list: 5

Number of processes used: 3

55% | 689/1261 [30:42<25:29, 2.67s/it]

The times for each task are: [0.8559, 0.653925, 1.665162, 0.289751, 0.382381] with:

Minimum: 0.289751 Maximum: 1.665162 Average: 0.7694 seconds

Number of cars 1

bbox:car_number ((875, 400), (942, 472)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

55% | 690/1261 [30:45<25:27, 2.67s/it]

The times for each task are: [0.703327, 1.427851, 0.877366, 0.356448, 0.431991] with:

Minimum: 0.356448 Maximum: 1.427851 Average: 0.7594 seconds

Number of cars 1

bbox:car_number ((875, 400), (942, 472)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

55% | 691/1261 [30:47<25:24, 2.67s/it]

The times for each task are: [1.357672, 0.890861, 0.655662, 0.30926, 0.408325] with:

Minimum: 0.30926 Maximum: 1.357672 Average: 0.7244 seconds

Number of cars 1

bbox:car_number ((865, 400), (944, 472)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 40

Length of task list: 5

Number of processes used: 3

55% | 692/1261 [30:50<25:21, 2.67s/it]

The times for each task are: [0.552601, 0.901582, 1.339632, 0.39107, 0.277874] with:

Minimum: 0.277874 Maximum: 1.339632 Average: 0.6926 seconds

Number of cars 1

bbox:car_number ((875, 400), (942, 472)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2
Car Number: 1 Car Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (942, 472))]
Length of task list: 5
Number of processes used: 3

55% | 693/1261 [30:53<25:19, 2.67s/it]

The times for each task are: [0.837153, 0.682392, 1.318521, 0.376182, 0.337697] with:
Minimum: 0.337697 Maximum: 1.318521 Average: 0.7104 seconds

Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 2
((875, 400), (942, 472)) Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (942, 472))]
Length of task list: 5
Number of processes used: 3

55% | 694/1261 [30:56<25:16, 2.67s/it]

The times for each task are: [1.035089, 1.371364, 0.755388, 0.464512, 0.342738] with:
Minimum: 0.342738 Maximum: 1.371364 Average: 0.7938 seconds

Number of cars 1
bbox:car_number ((875, 400), (944, 472)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2
((875, 400), (944, 472)) Positions: [((1075, 427), (1133, 485)), ((1051, 400), (1133, 495)), ((1051, 400), (944, 472))]
Length of task list: 5
Number of processes used: 3

55% | 695/1261 [30:58<25:13, 2.67s/it]

The times for each task are: [0.86966, 0.637112, 1.518779, 0.416527, 0.301851] with:
Minimum: 0.301851 Maximum: 1.518779 Average: 0.7488 seconds

Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 2

((875, 400), ((875, 400), (10754, 4272)), ((1133, 495)), ((1051, 400), ((1051, 400), (10754, 4272)), ((1133, 495))), ((9425, 147200)), Length of task list: 5
Number of processes used: 3

55% | 696/1261 [31:01<25:11, 2.67s/it]

The times for each task are: [0.900591, 0.590284, 1.405001, 0.432376, 0.332006] with:

Minimum: 0.332006 Maximum: 1.405001 Average: 0.7321 seconds

Number of cars 1
bbox:car_number ((875, 400), (934, 472)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 2

((875, 400), ((875, 400), (10754, 4272)), ((1133, 495)), ((9425, 147200)), ((18733, 4095)), (9341, 05472400))

Length of task list: 5

Number of processes used: 3

55% | 697/1261 [31:04<25:08, 2.67s/it]

The times for each task are: [0.625436, 1.534473, 0.930302, 0.345635, 0.510876] with:

Minimum: 0.345635 Maximum: 1.534473 Average: 0.7893 seconds

Number of cars 1
bbox:car_number ((865, 400), (942, 478)) : 1

The minimum distance from car: 1 is 3.1622776601683795

totalCars: 2

((875, 400), ((875, 400), (10754, 4272)), ((1133, 495)), ((9425, 147200)), ((18733, 4095)), (9341, 05472400))

Length of task list: 5

Number of processes used: 3

55% | 698/1261 [31:06<25:05, 2.67s/it]

The times for each task are: [0.658715, 1.007638, 1.458113, 0.372341, 0.424743] with:

Minimum: 0.372341 Maximum: 1.458113 Average: 0.7843 seconds

Number of cars 1
bbox:car_number ((865, 400), (942, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 2

((875, 400), ((875, 400), (10754, 4272)), ((1133, 495)), ((9425, 147200)), ((18733, 4095)), (9341, 05472400))

```
Length of task list: 5
Number of processes used: 3
```

```
55%|    | 699/1261 [31:09<25:03, 2.67s/it]
```

```
The times for each task are: [0.63039, 0.86618, 1.642943, 0.315458, 0.393725] with:
```

```
Minimum: 0.315458 Maximum: 1.642943 Average: 0.7697 seconds
```

```
Number of cars 1
```

```
bbox:car_number ((875, 400), (942, 477)) : 1
```

```
The minimum distance from car: 1 is 5.0990195135927845
```

```
totalCars: 2
```

```
((875, 400), ((875, 400), (10754, 4472)), ((11375, 4850)), ((9405, 4720)), ((18753, 4495)), (9341, 05472)400
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
56%|    | 700/1261 [31:12<25:00, 2.67s/it]
```

```
The times for each task are: [1.018203, 1.433601, 0.621619, 0.418987, 0.316604] with:
```

```
Minimum: 0.316604 Maximum: 1.433601 Average: 0.7618 seconds
```

```
Number of cars 2
```

```
bbox:car_number ((875, 415), (934, 472)) : 1
```

```
The minimum distance from car: 1 is 6.4031242374328485
```

```
bbox:car_number ((1181, 431), (1253, 495)) : 2
```

```
The minimum distance from car: 1 is 28.071337695236398
```

```
totalCars: 2
```

```
((875, 400), ((875, 400), (10754, 4472)), ((11375, 4850)), ((9405, 4720)), ((18753, 4495)), (9341, 05472)400
```

```
totalCars: 2
```

```
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
56%|    | 701/1261 [31:14<24:57, 2.67s/it]
```

```
The times for each task are: [0.842085, 1.360419, 0.7174, 0.298131, 0.407636] with:
```

```
Minimum: 0.298131 Maximum: 1.360419 Average: 0.7251 seconds
```

```
Number of cars 2
```

```
bbox:car_number ((865, 400), (944, 478)) : 1
```

```
The minimum distance from car: 1 is 4.0
bbox:car_number ((1195, 460), (1212, 462)) : 2
totalCars: 2
72)), ((875, 400) Num(842, 472) car ,Pos(875, 400)((10754, 4272)),,(11375, 4850)),, ((9425, 4720)),, (((18753, 4005)),,(9341, 0547, 2)40
Length of task list: 5
Number of processes used: 3
```

56% | 702/1261 [31:17<24:55, 2.67s/it]

The times for each task are: [0.930469, 1.358865, 0.675306, 0.434649, 0.399214] with:

Minimum: 0.399214 Maximum: 1.358865 Average: 0.7597 seconds

```
Number of cars 1
bbox:car_number ((865, 400), (942, 477)) : 1
The minimum distance from car: 1 is 1.4142135623730951
totalCars: 2
72)), ((875, 400) Num(842, 472) car ,Pos(875, 400)((10754, 4272)),,(11375, 4850)),, ((9425, 4720)),, (((18753, 4005)),,(9341, 0547, 2)40
Length of task list: 5
Number of processes used: 3
```

56% | 703/1261 [31:20<24:52, 2.67s/it]

The times for each task are: [0.613457, 1.096794, 1.616539, 0.444211, 0.376803] with:

Minimum: 0.376803 Maximum: 1.616539 Average: 0.8296 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 2
72)), ((875, 400) Num(842, 472) car ,Pos(875, 400)((10754, 4272)),,(11375, 4850)),, ((9425, 4720)),, (((18753, 4005)),,(9341, 0547, 2)40
Length of task list: 5
Number of processes used: 3
```

56% | 704/1261 [31:22<24:49, 2.67s/it]

The times for each task are: [0.95202, 0.689695, 1.437393, 0.30225, 0.476243] with:

Minimum: 0.30225 Maximum: 1.437393 Average: 0.7715 seconds

```
Number of cars 2
bbox:car_number ((875, 415), (934, 472)) : 1
```

```
The minimum distance from car: 1 is 8.06225774829855
bbox:car_number ((1210, 445), (1241, 492)) : 2
totalCars: 2
((875, 410), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512)))
Length of task list: 5
Number of processes used: 3
```

56% | 705/1261 [31:25<24:46, 2.67s/it]

The times for each task are: [0.864335, 0.724038, 1.383529, 0.43355, 0.303081] with:

Minimum: 0.303081 Maximum: 1.383529 Average: 0.7417 seconds

Number of cars 2

bbox:car_number ((875, 415), (934, 472)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1204, 427), (1257, 477)) : 2

The minimum distance from car: 1 is 17.029386365926403

totalCars: 2

((875, 410), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512)))
Length of task list: 5
Number of processes used: 3

56% | 706/1261 [31:28<24:44, 2.67s/it]

The times for each task are: [0.917048, 0.611168, 1.398347, 0.432168, 0.320899] with:

Minimum: 0.320899 Maximum: 1.398347 Average: 0.7359 seconds

Number of cars 2

bbox:car_number ((875, 415), (934, 472)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1158, 419), (1268, 529)) : 2

The minimum distance from car: 1 is 27.80287754891569

totalCars: 2

((875, 410), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512)))
Length of task list: 5
Number of processes used: 3

56% | 707/1261 [31:30<24:41, 2.67s/it]

The times for each task are: [0.896248, 1.380468, 0.701361, 0.440423, 0.406604] with:

Minimum: 0.406604 Maximum: 1.380468 Average: 0.765 seconds

Number of cars 3

bbox:car_number ((875, 415), (934, 477)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((1175, 419), (1253, 498)) : 2

The minimum distance from car: 1 is 16.0312195418814

bbox:car_number ((1144, 481), (1160, 498)) : 3

totalCars: 2

((875, 415), (1175, 419), (1144, 481)), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1175, 400), (1273, 478)), ((1144, 481), (1160, 498))

totalCars: 2

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1175, 400), (1273, 478)), ((1144, 481), (1160, 498))]

Length of task list: 5

Number of processes used: 3

56% | 708/1261 [31:33<24:38, 2.67s/it]

The times for each task are: [0.867633, 0.626343, 1.446589, 0.407227, 0.31813] with:

Minimum: 0.31813 Maximum: 1.446589 Average: 0.7332 seconds

Number of cars 3

bbox:car_number ((1175, 400), (1257, 497)) : 1

The minimum distance from car: 1 is 10.198039027185569

bbox:car_number ((875, 415), (934, 477)) : 2

The minimum distance from car: 1 is 0.0

bbox:car_number ((1158, 445), (1167, 485)) : 3

totalCars: 2

((875, 415), (1175, 400), (1257, 497)), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1175, 400), (1273, 478)), ((1158, 445), (1167, 485))

totalCars: 2

((875, 415), (1175, 400), (1257, 497)), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1175, 400), (1273, 478)), ((1158, 445), (1167, 485))

Length of task list: 5

Number of processes used: 3

56% | 709/1261 [31:36<24:36, 2.67s/it]

The times for each task are: [0.605683, 1.377323, 0.998609, 0.387563, 0.349071] with:

Minimum: 0.349071 Maximum: 1.377323 Average: 0.7436 seconds

Number of cars 2

```
bbox:car_number ((875, 415), (934, 472)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1156, 419), (1241, 495)) : 2
The minimum distance from car: 1 is 20.12461179749811
totalCars: 2
((875, 400), (942, 472)), ((1136, 400), (1238, 495))
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 495))]
Length of task list: 5
Number of processes used: 3
```

56% | 710/1261 [31:38<24:33, 2.67s/it]

The times for each task are: [0.634753, 1.458658, 0.860363, 0.303077, 0.39689] with:
Minimum: 0.303077 Maximum: 1.458658 Average: 0.7307 seconds

```
Number of cars 2
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 8.06225774829855
bbox:car_number ((1136, 400), (1238, 495)) : 2
The minimum distance from car: 1 is 14.866068747318506
totalCars: 2
((875, 400), (942, 472)), ((1136, 400), (1238, 495))
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 495))]
Length of task list: 5
Number of processes used: 3
```

56% | 711/1261 [31:41<24:30, 2.67s/it]

The times for each task are: [0.853756, 0.670124, 1.567577, 0.388955, 0.379643] with:
Minimum: 0.379643 Maximum: 1.567577 Average: 0.772 seconds

```
Number of cars 4
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1129, 400), (1230, 495)) : 2
The minimum distance from car: 1 is 8.0
bbox:car_number ((1113, 445), (1124, 492)) : 3
bbox:car_number ((1105, 490), (1107, 492)) : 4
totalCars: 2
((875, 400), (942, 472)), ((1129, 400), (1230, 495)), ((1113, 445), (1124, 492)), ((1105, 490), (1107, 492))
totalCars: 2
```

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 478))]
Length of task list: 5
Number of processes used: 3

56% | 712/1261 [31:43<24:28, 2.67s/it]

The times for each task are: [0.930298, 0.735196, 1.353456, 0.337763, 0.393047] with:

Minimum: 0.337763 Maximum: 1.353456 Average: 0.75 seconds

Number of cars 2
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1113, 400), (1219, 495)) : 2
The minimum distance from car: 1 is 13.0
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 478))]
Length of task list: 5
Number of processes used: 3

57% | 713/1261 [31:46<24:25, 2.67s/it]

The times for each task are: [0.801893, 1.348542, 0.674047, 0.48848, 0.38819] with:

Minimum: 0.38819 Maximum: 1.348542 Average: 0.7402 seconds

Number of cars 2
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1082, 400), (1214, 522)) : 2
The minimum distance from car: 1 is 22.80350850198276
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 478))]
Length of task list: 5
Number of processes used: 3

57% | 714/1261 [31:49<24:22, 2.67s/it]

The times for each task are: [0.630378, 0.906998, 1.603834, 0.299531, 0.473786] with:

Minimum: 0.299531 Maximum: 1.603834 Average: 0.7829 seconds

Number of cars 2
bbox:car_number ((875, 400), (934, 472)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1060, 419), (1208, 529)) : 2
The minimum distance from car: 1 is 19.1049731745428
totalCars: 2
((875, 400), (934, 472)), Pos((875, 400), (1060, 419), (1208, 529))
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1150, 520)), ((1197, 512), (1082, 415))]
Length of task list: 5
Number of processes used: 3

57% | 715/1261 [31:52<24:20, 2.67s/it]

The times for each task are: [0.893958, 0.702755, 1.511005, 0.406834, 0.337541] with:

Minimum: 0.337541 Maximum: 1.511005 Average: 0.7704 seconds

Number of cars 3
bbox:car_number ((875, 400), (934, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1060, 400), (1208, 529)) : 2
The minimum distance from car: 1 is 10.0
bbox:car_number ((1150, 520), (1180, 529)) : 3
totalCars: 2
((875, 400), (934, 472)), Pos((875, 400), (1060, 419), (1208, 529))
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1150, 520)), ((1197, 512), (1082, 415))]
Length of task list: 5
Number of processes used: 3

57% | 716/1261 [31:55<24:17, 2.67s/it]

The times for each task are: [0.579873, 1.480459, 1.0331, 0.432091, 0.287593] with:

Minimum: 0.287593 Maximum: 1.480459 Average: 0.7626 seconds

Number of cars 2
bbox:car_number ((1082, 400), (1197, 512)) : 1
The minimum distance from car: 1 is 9.433981132056603
bbox:car_number ((875, 415), (934, 462)) : 2
The minimum distance from car: 1 is 2.0
totalCars: 2

```

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
totalCars: 2
Length of task list: 5
Number of processes used: 3

57% | 717/1261 [31:57<24:15, 2.67s/it]

The times for each task are: [1.039622, 0.638799, 1.415466, 0.399688, 0.485779] with:

Minimum: 0.399688 Maximum: 1.415466 Average: 0.7959 seconds

Number of cars 3
bbox:car_number ((875, 400), (934, 472)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1075, 400), (1200, 512)) : 2
The minimum distance from car: 1 is 2.0
bbox:car_number ((1198, 520), (1200, 567)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3

57% | 718/1261 [32:00<24:12, 2.67s/it]

The times for each task are: [0.973989, 0.684481, 1.376145, 0.345036, 0.446545] with:

Minimum: 0.345036 Maximum: 1.376145 Average: 0.7652 seconds

Number of cars 3
bbox:car_number ((875, 400), (934, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1102, 400), (1185, 495)) : 2
The minimum distance from car: 1 is 10.816653826391969
bbox:car_number ((1045, 460), (1092, 507)) : 3
totalCars: 2
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3

```

57% | 719/1261 [32:03<24:09, 2.68s/it]

The times for each task are: [1.338322, 1.041424, 0.612184, 0.407693, 0.508835] with:

Minimum: 0.407693 Maximum: 1.338322 Average: 0.7817 seconds

Number of cars 5

bbox:car_number ((875, 400), (934, 472)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1030, 400), (1185, 522)) : 2

The minimum distance from car: 1 is 38.62641583165593

bbox:car_number ((1175, 493), (1180, 495)) : 3

bbox:car_number ((1179, 514), (1182, 567)) : 4

bbox:car_number ((1204, 514), (1238, 541)) : 5

totalCars: 2

((875, 400), ((1030, 400), (1185, 522)), ((1175, 493), (1180, 495)), ((1179, 514), (1182, 567)), ((1204, 514), (1238, 541)))

totalCars: 2

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

57% | 720/1261 [32:06<24:07, 2.68s/it]

The times for each task are: [0.893929, 0.616037, 1.414561, 0.29215, 0.429341] with:

Minimum: 0.29215 Maximum: 1.414561 Average: 0.7292 seconds

Number of cars 2

bbox:car_number ((875, 400), (934, 472)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1030, 400), (1181, 507)) : 2

The minimum distance from car: 1 is 8.246211251235321

totalCars: 2

((875, 400), ((1030, 400), (1181, 507)), ((1175, 493), (1180, 495)), ((1179, 514), (1182, 567)), ((1204, 514), (1238, 541)))

totalCars: 2

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

57% | 721/1261 [32:08<24:04, 2.68s/it]

The times for each task are: [0.63153, 0.977535, 1.375963, 0.449823, 0.278321] with:

Minimum: 0.278321 Maximum: 1.375963 Average: 0.7426 seconds

```
Number of cars 3
bbox:car_number ((880, 400), (934, 462)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1082, 400), (1167, 485)) : 2
The minimum distance from car: 1 is 21.95449840010015
bbox:car_number ((1195, 481), (1253, 537)) : 3
totalCars: 3
((875, 400), ((1060, 400), (1160, 498))), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512)), ((1195, 481), (1253, 537))
totalCars: 3
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
totalCars: 3
Car Number: 3 Car Positions: [((1195, 481), (1253, 537))]
Length of task list: 5
Number of processes used: 3
```

57% | 722/1261 [32:11<24:01, 2.67s/it]

The times for each task are: [0.876256, 1.40116, 0.670857, 0.480732, 0.369311] with:

Minimum: 0.369311 Maximum: 1.40116 Average: 0.7597 seconds

```
Number of cars 3
bbox:car_number ((875, 400), (942, 462)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1060, 400), (1162, 498)) : 2
The minimum distance from car: 1 is 14.7648230602334
bbox:car_number ((1165, 535), (1167, 537)) : 3
totalCars: 3
((875, 400), ((1060, 400), (1160, 498))), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512)), ((1195, 481), (1253, 537))
totalCars: 3
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

57% | 723/1261 [32:13<23:59, 2.67s/it]

The times for each task are: [0.851963, 1.598844, 0.621627, 0.347944, 0.447263] with:

Minimum: 0.347944 Maximum: 1.598844 Average: 0.7735 seconds

```
Number of cars 2
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1060, 400), (1160, 498)) : 2
The minimum distance from car: 1 is 1.0
```

```
totalCars: 3
((875, 400), (942, 472)), Pos((875, 400), ((1075, 472)), (1175, 485)), ((945, 472)), ((1875, 495)), (934, 547, 400)
totalCars: 3
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 472))]
Length of task list: 5
Number of processes used: 3
```

57% | 724/1261 [32:16<23:56, 2.67s/it]

The times for each task are: [0.946702, 0.686541, 1.363525, 0.340956, 0.418226] with:

Minimum: 0.340956 Maximum: 1.363525 Average: 0.7512 seconds

```
Number of cars 3
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1067, 400), (1152, 516)) : 2
The minimum distance from car: 1 is 9.055385138137417
bbox:car_number ((1144, 505), (1200, 573)) : 3
totalCars: 4
((875, 400), (942, 472)), Pos((875, 400), ((1075, 472)), (1175, 485)), ((945, 472)), ((1875, 495)), (934, 547, 400)
totalCars: 4
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 472))]
totalCars: 4
Car Number: 3 Car Positions: [((1144, 505), (1200, 573))]
Length of task list: 5
Number of processes used: 3
```

57% | 725/1261 [32:19<23:54, 2.68s/it]

The times for each task are: [0.804874, 1.544335, 0.70431, 0.374554, 0.332415] with:

Minimum: 0.332415 Maximum: 1.544335 Average: 0.7521 seconds

```
Number of cars 2
bbox:car_number ((875, 400), (942, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1075, 400), (1143, 477)) : 2
The minimum distance from car: 1 is 20.0
totalCars: 4
((875, 400), (942, 472)), Pos((875, 400), ((1075, 472)), (1175, 485)), ((945, 472)), ((1875, 495)), (934, 547, 400)
totalCars: 4
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 472))]
Length of task list: 5
Number of processes used: 3
```

58% | 726/1261 [32:22<23:51, 2.68s/it]

The times for each task are: [0.619395, 0.997358, 1.369839, 0.467627, 0.38139] with:

Minimum: 0.38139 Maximum: 1.369839 Average: 0.7671 seconds

Number of cars 2

bbox:car_number ((880, 400), (942, 462)) : 1

The minimum distance from car: 1 is 5.830951894845301

bbox:car_number ((1048, 400), (1143, 485)) : 2

The minimum distance from car: 1 is 14.560219778561036

totalCars: 4

((875, 400), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1048, 400), (1143, 485)), ((1090, 505), (1107, 507)), ((1129, 508), (1133, 512)), ((1135, 508), (1137, 552)))

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1048, 400), (1143, 485)), ((1090, 505), (1107, 507)), ((1129, 508), (1133, 512)), ((1135, 508), (1137, 552))]

Length of task list: 5

Number of processes used: 3

58% | 727/1261 [32:24<23:48, 2.68s/it]

The times for each task are: [0.63124, 1.389521, 0.996025, 0.386931, 0.331013] with:

Minimum: 0.331013 Maximum: 1.389521 Average: 0.7469 seconds

Number of cars 5

bbox:car_number ((875, 400), (944, 472)) : 1

The minimum distance from car: 1 is 5.385164807134504

bbox:car_number ((1045, 400), (1143, 507)) : 2

The minimum distance from car: 1 is 11.045361017187261

bbox:car_number ((1090, 505), (1107, 507)) : 3

bbox:car_number ((1129, 508), (1133, 512)) : 4

bbox:car_number ((1135, 508), (1137, 552)) : 5

totalCars: 4

((875, 400), ((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1048, 400), (1143, 485)), ((1090, 505), (1107, 507)), ((1129, 508), (1133, 512)), ((1135, 508), (1137, 552)))

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1204, 400)), ((1048, 400), (1143, 485)), ((1090, 505), (1107, 507)), ((1129, 508), (1133, 512)), ((1135, 508), (1137, 552))]

Length of task list: 5

Number of processes used: 3

58% | 728/1261 [32:27<23:45, 2.68s/it]

The times for each task are: [0.619076, 1.386145, 0.893888, 0.28514, 0.36928] with:

Minimum: 0.28514 Maximum: 1.386145 Average: 0.7107 seconds

Number of cars 8

bbox:car_number ((880, 400), (942, 462)) : 1

The minimum distance from car: 1 is 5.385164807134504

bbox:car_number ((1046, 400), (1137, 498)) : 2

The minimum distance from car: 1 is 5.0

bbox:car_number ((1113, 492), (1116, 495)) : 3

bbox:car_number ((1122, 495), (1149, 498)) : 4

bbox:car_number ((1165, 495), (1180, 498)) : 5

bbox:car_number ((1135, 505), (1137, 552)) : 6

bbox:car_number ((1165, 505), (1181, 522)) : 7

bbox:car_number ((1180, 535), (1181, 552)) : 8

totalCars: 4

((72)), ((875, 400), Num(1842, 1472)), Pos(1875, 400), ((10754, 4472)), ((11375, 4850)), ((9425, 14720)), ((18133, 44905)), ((9341, 05472)400

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

58% | 729/1261 [32:30<23:43, 2.68s/it]

The times for each task are: [0.933067, 1.53073, 0.699771, 0.409377, 0.321084] with:

Minimum: 0.321084 Maximum: 1.53073 Average: 0.7788 seconds

Number of cars 4

bbox:car_number ((882, 400), (942, 459)) : 1

The minimum distance from car: 1 is 2.23606797749979

bbox:car_number ((1065, 415), (1133, 485)) : 2

The minimum distance from car: 1 is 8.06225774829855

bbox:car_number ((1113, 493), (1116, 495)) : 3

bbox:car_number ((1120, 495), (1181, 564)) : 4

The minimum distance from car: 3 is 24.166091947189145

totalCars: 4

((72)), ((875, 400), Num(1842, 1472)), Pos(1875, 400), ((10754, 4472)), ((11375, 4850)), ((9425, 14720)), ((18133, 44905)), ((9341, 05472)400

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

totalCars: 4

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

58% | 730/1261 [32:32<23:40, 2.67s/it]

The times for each task are: [0.558256, 1.364794, 0.393963, 0.986624, 0.285341] with:

Minimum: 0.285341 Maximum: 1.364794 Average: 0.7178 seconds

Number of cars 5

bbox:car_number ((880, 400), (942, 462)) : 1

The minimum distance from car: 1 is 2.23606797749979

bbox:car_number ((1060, 400), (1133, 485)) : 2

The minimum distance from car: 1 is 8.54400374531753

bbox:car_number ((1102, 481), (1133, 485)) : 3

bbox:car_number ((1113, 493), (1180, 560)) : 4

The minimum distance from car: 3 is 5.0

bbox:car_number ((1183, 493), (1185, 539)) : 5

totalCars: 4

((875, 400), Num(842, 1472)), Pos(875, 400), ((10754, 4272)), , ((11375, 4850)), , ((9425, 14720)), , ((81753, 4495)), , (9341, 05472)40

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

totalCars: 4

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 493), (1185, 539))]

Length of task list: 5

Number of processes used: 3

58% | 731/1261 [32:35<23:37, 2.67s/it]

The times for each task are: [0.824988, 0.612896, 1.629678, 0.472729, 0.295929] with:

Minimum: 0.295929 Maximum: 1.629678 Average: 0.7672 seconds

Number of cars 5

bbox:car_number ((880, 400), (953, 472)) : 1

The minimum distance from car: 1 is 7.0710678118654755

bbox:car_number ((1044, 400), (1122, 485)) : 2

The minimum distance from car: 1 is 13.0

bbox:car_number ((1144, 446), (1257, 560)) : 3

The minimum distance from car: 3 is 24.73863375370596

bbox:car_number ((1082, 505), (1087, 539)) : 4

bbox:car_number ((1102, 508), (1122, 539)) : 5

totalCars: 4

((875, 400), Num(842, 1472)), Pos(875, 400), ((10754, 4272)), , ((11375, 4850)), , ((9425, 14720)), , ((81753, 4495)), , (9341, 05472)40

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

totalCars: 4

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560))]

Length of task list: 5

Number of processes used: 3

58% | 732/1261 [32:38<23:35, 2.67s/it]

The times for each task are: [0.858225, 1.56949, 0.650442, 0.410356, 0.345748] with:

Minimum: 0.345748 Maximum: 1.56949 Average: 0.7669 seconds

Number of cars 6

bbox:car_number ((882, 400), (953, 462)) : 1

The minimum distance from car: 1 is 5.0990195135927845

bbox:car_number ((1044, 400), (1124, 485)) : 2

The minimum distance from car: 1 is 1.0

bbox:car_number ((1195, 460), (1230, 516)) : 3

bbox:car_number ((1113, 514), (1182, 582)) : 4

The minimum distance from car: 3 is 22.02271554554524

bbox:car_number ((1158, 514), (1162, 518)) : 5

bbox:car_number ((1175, 514), (1185, 518)) : 6

totalCars: 4

((875, 400), ((1044, 400), Pos((875, 400), ((1044, 400), (1124, 485))), ((942, 400), ((1195, 460), (1230, 516))), ((1113, 514), (1182, 582)), ((1158, 514), (1162, 518)), ((1175, 514), (1185, 518)))

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1185, 518))]

totalCars: 4

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1175, 514))]

Length of task list: 5

Number of processes used: 3

58% | 733/1261 [32:40<23:32, 2.67s/it]

The times for each task are: [0.655291, 0.902723, 1.402013, 0.291643, 0.379286] with:

Minimum: 0.291643 Maximum: 1.402013 Average: 0.7262 seconds

Number of cars 4

bbox:car_number ((882, 400), (942, 462)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1044, 400), (1116, 495)) : 2

The minimum distance from car: 1 is 6.4031242374328485

bbox:car_number ((1102, 446), (1257, 586)) : 3

The minimum distance from car: 3 is 24.698178070456937

bbox:car_number ((1090, 492), (1093, 495)) : 4

totalCars: 4

((875, 400), ((1044, 400), Pos((875, 400), ((1044, 400), (1116, 495))), ((942, 400), ((1102, 446), (1257, 586))), ((1090, 492), (1093, 495))))

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1185, 518))]

totalCars: 4

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1183, 518))]

Length of task list: 5

Number of processes used: 3

58%| 734/1261 [32:43<23:29, 2.68s/it]

The times for each task are: [0.625129, 1.015637, 1.389927, 0.384492, 0.343476] with:

Minimum: 0.343476 Maximum: 1.389927 Average: 0.7517 seconds

Number of cars 4

bbox:car_number ((880, 400), (953, 477)) : 1

The minimum distance from car: 1 is 8.06225774829855

bbox:car_number ((1044, 400), (1107, 498)) : 2

The minimum distance from car: 1 is 5.385164807134504

bbox:car_number ((1105, 431), (1253, 586)) : 3

The minimum distance from car: 3 is 8.0

bbox:car_number ((1144, 493), (1149, 498)) : 4

totalCars: 4

((72)), ((875, 400) Num(1842, 1472) Car Pos(1875 on 400)((10754, 4272)), (11375, 4850)), ((9425, 147200)), (((18753, 4005)),), (9341, 05472) 400

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 560))]

totalCars: 4

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1204, 512))]

Length of task list: 5

Number of processes used: 3

58%| 735/1261 [32:46<23:27, 2.68s/it]

The times for each task are: [0.586867, 1.052069, 1.437867, 0.390199, 0.301282] with:

Minimum: 0.301282 Maximum: 1.437867 Average: 0.7537 seconds

Number of cars 3

bbox:car_number ((880, 400), (953, 477)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1021, 400), (1107, 498)) : 2

The minimum distance from car: 1 is 11.0

bbox:car_number ((1102, 446), (1242, 591)) : 3

The minimum distance from car: 3 is 12.206555615733702

totalCars: 4

((72)), ((875, 400) Num(1842, 1472) Car Pos(1875 on 400)((10754, 4272)), (11375, 4850)), ((9425, 147200)), (((18753, 4005)),), (9341, 05472) 400

totalCars: 4

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1257, 560))]

totalCars: 4

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1204, 512))]

Length of task list: 5

Number of processes used: 3

58%| 736/1261 [32:48<23:24, 2.68s/it]

The times for each task are: [0.872361, 0.673992, 1.364075, 0.388901, 0.300251] with:

Minimum: 0.300251 Maximum: 1.364075 Average: 0.7199 seconds

Number of cars 2

bbox:car_number ((880, 400), (954, 477)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1020, 400), (1242, 592)) : 2

The minimum distance from car: 3 is 46.52956049652737

totalCars: 4

((875, 400), ((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44

Length of task list: 5

Number of processes used: 3

58%| 737/1261 [32:51<23:21, 2.68s/it]

The times for each task are: [0.600003, 1.409361, 0.890104, 0.303593, 0.549574] with:

Minimum: 0.303593 Maximum: 1.409361 Average: 0.7505 seconds

Number of cars 4

bbox:car_number ((882, 400), (953, 478)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1021, 400), (1241, 592)) : 2

The minimum distance from car: 3 is 0.0

bbox:car_number ((1210, 550), (1211, 560)) : 3

bbox:car_number ((1225, 550), (1241, 560)) : 4

totalCars: 4

((875, 400), ((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44

Length of task list: 5

Number of processes used: 3

59%| 738/1261 [32:54<23:19, 2.68s/it]

The times for each task are: [0.551641, 0.913567, 0.388335, 1.674443, 0.2917] with:

Minimum: 0.2917 Maximum: 1.674443 Average: 0.7639 seconds

Number of cars 4
bbox:car_number ((880, 400), (957, 478)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1027, 400), (1133, 498)) : 2
The minimum distance from car: 1 is 16.0
bbox:car_number ((1195, 431), (1238, 447)) : 3
bbox:car_number ((1084, 454), (1241, 586)) : 4
The minimum distance from car: 3 is 31.76476034853718
totalCars: 4
((875, 400), ((1021, 400), (1133, 498), (1195, 431), (1084, 454)),
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 586))]
totalCars: 4
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1102, 446))]
Length of task list: 5
Number of processes used: 3

59% | 739/1261 [32:56<23:16, 2.67s/it]

The times for each task are: [0.878489, 0.633045, 1.575979, 0.430598, 0.361588] with:

Minimum: 0.361588 Maximum: 1.575979 Average: 0.7759 seconds

Number of cars 2
bbox:car_number ((882, 400), (957, 477)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1021, 400), (1230, 591)) : 2
The minimum distance from car: 3 is 6.082762530298219
totalCars: 4
((875, 400), ((1021, 400), (1133, 498), (1195, 431), (1084, 454)),
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1181, 564))]
totalCars: 4
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1181, 564))]
Length of task list: 5
Number of processes used: 3

59% | 740/1261 [32:59<23:13, 2.67s/it]

The times for each task are: [0.608259, 1.379284, 0.98812, 0.395596, 0.287741] with:

Minimum: 0.287741 Maximum: 1.379284 Average: 0.7318 seconds

Number of cars 2
bbox:car_number ((880, 400), (957, 478)) : 1

```
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1021, 400), (1242, 582)) : 2
The minimum distance from car: 3 is 7.211102550927978
totalCars: 4
((875, 400), Num(842, 1472), Car Positions(1875, 400), ((10754, 4072)), ((11375, 4850)), ((94051, 4720)), ((18753, 4095)), ((934105472)4072)), 
Car Number: 3
Length of task list: 5
Number of processes used: 3
```

59% | 741/1261 [33:01<23:10, 2.67s/it]

```
The times for each task are: [1.422631, 0.852821, 0.712672, 0.445625, 0.355802] with:
Minimum: 0.355802 Maximum: 1.422631 Average: 0.7579 seconds
```

```
Number of cars 5
bbox:car_number ((882, 400), (957, 477)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1027, 400), (1086, 472)) : 2
The minimum distance from car: 1 is 27.294688127912362
bbox:car_number ((1067, 446), (1214, 586)) : 3
The minimum distance from car: 3 is 22.360679774997898
bbox:car_number ((1082, 462), (1093, 472)) : 4
bbox:car_number ((1051, 493), (1067, 497)) : 5
totalCars: 4
((875, 400), Num(842, 1472), Car Positions(1875, 400), ((10754, 4072)), ((11375, 4850)), ((94051, 4720)), ((18753, 4095)), ((934105472)4072)), 
Car Number: 1
Length of task list: 5
Number of processes used: 3
```

59% | 742/1261 [33:04<23:08, 2.67s/it]

```
The times for each task are: [0.921158, 0.694141, 1.351985, 0.334107, 0.446979] with:
Minimum: 0.334107 Maximum: 1.351985 Average: 0.7497 seconds
```

```
Number of cars 2
bbox:car_number ((886, 400), (971, 512)) : 1
The minimum distance from car: 1 is 20.12461179749811
bbox:car_number ((1008, 400), (1214, 560)) : 2
The minimum distance from car: 3 is 22.825424421026653
```

```
totalCars: 4
((875, 400), ((875, 400), (10754, 4072)), ((11375, 4850)), ((9405, 4720)), ((18753, 4095)), ((9341, 05472)) totalCars: 4
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 4472)), ((875, 400)), ((875, 400)), ((10754, 4072)), ((11375, 4850)), ((9405, 4720)), ((18753, 4095)), ((9341, 05472)) Length of task list: 5
Number of processes used: 3
```

59% | 743/1261 [33:07<23:05, 2.67s/it]

The times for each task are: [0.855447, 0.600728, 0.41203, 1.583467, 0.352721] with:

Minimum: 0.352721 Maximum: 1.583467 Average: 0.7609 seconds

```
Number of cars 2
bbox:car_number ((880, 400), (971, 507)) : 1
The minimum distance from car: 1 is 4.242640687119285
bbox:car_number ((998, 400), (1238, 539)) : 2
The minimum distance from car: 3 is 13.038404810405298
totalCars: 4
((875, 400), ((875, 400), (10754, 4072)), ((11375, 4850)), ((9405, 4720)), ((18753, 4095)), ((9341, 05472)) totalCars: 4
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 4472)), ((875, 400)), ((875, 400)), ((10754, 4072)), ((11375, 4850)), ((9405, 4720)), ((18753, 4095)), ((9341, 05472)) Length of task list: 5
Number of processes used: 3
```

59% | 744/1261 [33:10<23:02, 2.67s/it]

The times for each task are: [0.866403, 0.690042, 1.617633, 0.434421, 0.291931] with:

Minimum: 0.291931 Maximum: 1.617633 Average: 0.7801 seconds

```
Number of cars 3
bbox:car_number ((882, 400), (957, 478)) : 1
The minimum distance from car: 1 is 15.231546211727817
bbox:car_number ((1021, 415), (1242, 560)) : 2
The minimum distance from car: 3 is 22.20360331117452
bbox:car_number ((970, 481), (971, 485)) : 3
totalCars: 4
((875, 400), ((875, 400), (10754, 4072)), ((11375, 4850)), ((9405, 4720)), ((18753, 4095)), ((9341, 05472)) totalCars: 4
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 4472)), ((875, 400)), ((875, 400)), ((10754, 4072)), ((11375, 4850)), ((9405, 4720)), ((18753, 4095)), ((9341, 05472)) Length of task list: 5
Number of processes used: 3
```

59% | 745/1261 [33:12<23:00, 2.68s/it]

The times for each task are: [0.889403, 0.660522, 1.369005, 0.292901, 0.378315] with:

Minimum: 0.292901 Maximum: 1.369005 Average: 0.718 seconds

Number of cars 4

bbox:car_number ((886, 400), (963, 485)) : 1

The minimum distance from car: 1 is 5.830951894845301

bbox:car_number ((967, 400), (1077, 560)) : 2

bbox:car_number ((1060, 431), (1253, 582)) : 3

The minimum distance from car: 3 is 18.867962264113206

bbox:car_number ((1067, 460), (1106, 485)) : 4

totalCars: 5

Car Number: 2 Car Positions: [((967, 400), (1077, 560))]

totalCars: 5

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

Length of task list: 5

Number of processes used: 3

59% | 746/1261 [33:15<22:57, 2.68s/it]

The times for each task are: [0.62994, 0.893442, 1.378583, 0.296664, 0.389262] with:

Minimum: 0.296664 Maximum: 1.378583 Average: 0.7176 seconds

Number of cars 3

bbox:car_number ((886, 400), (963, 485)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1021, 431), (1273, 566)) : 2

The minimum distance from car: 3 is 12.041594578792296

bbox:car_number ((967, 462), (971, 485)) : 3

totalCars: 5

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

Length of task list: 5

Number of processes used: 3

59% | 747/1261 [33:18<22:54, 2.67s/it]

The times for each task are: [0.887607, 0.719204, 1.429004, 0.374518, 0.320613] with:

Minimum: 0.320613 Maximum: 1.429004 Average: 0.7462 seconds

Number of cars 2
bbox:car_number ((886, 400), (957, 478)) : 1
The minimum distance from car: 1 is 4.242640687119285
bbox:car_number ((1015, 415), (1257, 591)) : 2
The minimum distance from car: 3 is 12.083045973594572
totalCars: 5
((875, 400), Num(842, 1472), Pos(875, 400), ((1074, 4072), (1137, 4850)), ((945, 14720), ((1873, 4095), (934, 105472), 4072)), ((875, 400), Num(842, 1472), Pos(875, 400), ((1074, 4072), (1137, 4850)), ((945, 14720), ((1873, 4095), (934, 105472), 4072)))
totalCars: 5
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 537)), ((1102, 446), (1144, 446)), ((1102, 446), (1257, 560))]
Length of task list: 5
Number of processes used: 3

59% | 748/1261 [33:20<22:52, 2.67s/it]

The times for each task are: [0.591289, 1.01549, 0.392144, 1.604519, 0.347778] with:

Minimum: 0.347778 Maximum: 1.604519 Average: 0.7902 seconds
Number of cars 2
bbox:car_number ((882, 400), (963, 485)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((994, 400), (1187, 567)) : 2
The minimum distance from car: 3 is 41.19465984809196
totalCars: 5
((875, 400), Num(842, 1472), Pos(875, 400), ((1074, 4072), (1137, 4850)), ((945, 14720), ((1873, 4095), (934, 105472), 4072)), ((875, 400), Num(842, 1472), Pos(875, 400), ((1074, 4072), (1137, 4850)), ((945, 14720), ((1873, 4095), (934, 105472), 4072)))
totalCars: 5
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 446), (1144, 446)), ((1102, 446), (1257, 560))]
Length of task list: 5
Number of processes used: 3

59% | 749/1261 [33:23<22:49, 2.68s/it]

The times for each task are: [1.390378, 0.888554, 0.745806, 0.297894, 0.379356] with:

Minimum: 0.297894 Maximum: 1.390378 Average: 0.7404 seconds
Number of cars 5
bbox:car_number ((894, 400), (957, 472)) : 1
The minimum distance from car: 1 is 6.708203932499369
bbox:car_number ((994, 400), (1106, 539)) : 2
The minimum distance from car: 2 is 30.083217912982647
bbox:car_number ((1141, 415), (1241, 518)) : 3
bbox:car_number ((1113, 431), (1139, 472)) : 4

```
bbox:car_number ((1075, 508), (1122, 541)) : 5
totalCars: 6
((875, 400), (957, 472)), ((994, 400), (1257, 573))
totalCars: 6
Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539))]
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

59%| 750/1261 [33:26<22:46, 2.68s/it]

The times for each task are: [0.860635, 0.610331, 1.344914, 0.38491, 0.287425] with:
Minimum: 0.287425 Maximum: 1.344914 Average: 0.6976 seconds

```
Number of cars 2
bbox:car_number ((894, 400), (957, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((994, 400), (1257, 573)) : 2
The minimum distance from car: 3 is 20.248456731316587
totalCars: 6
((875, 400), (957, 472)), ((994, 400), (1257, 573))
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]
Length of task list: 5
Number of processes used: 3
```

60%| 751/1261 [33:29<22:44, 2.68s/it]

The times for each task are: [1.052629, 0.64695, 1.369074, 0.375634, 0.477008] with:
Minimum: 0.375634 Maximum: 1.369074 Average: 0.7843 seconds

```
Number of cars 1
bbox:car_number ((910, 415), (953, 459)) : 1
Length of task list: 5
Number of processes used: 3
```

60%| 752/1261 [33:32<22:41, 2.68s/it]

The times for each task are: [0.914037, 1.578675, 0.641451, 0.4357, 0.387732] with:

Minimum: 0.387732 Maximum: 1.578675 Average: 0.7915 seconds

Number of cars 5

bbox:car_number ((895, 400), (953, 462)) : 1

The minimum distance from car: 1 is 5.0990195135927845

bbox:car_number ((1181, 469), (1185, 472)) : 2

bbox:car_number ((1113, 514), (1122, 518)) : 3

bbox:car_number ((1113, 520), (1122, 529)) : 4

bbox:car_number ((1120, 535), (1122, 552)) : 5

totalCars: 6

72)), ((875, 460), Num(842, 1472)), Pos(875, 400), ((10754, 4072)), ((10375, 4850)), ((9405, 14720)), ((18753, 4095)), ((9341, 05472400))

Length of task list: 5

Number of processes used: 3

60% | 753/1261 [33:34<22:39, 2.68s/it]

The times for each task are: [0.925772, 0.598319, 1.559414, 0.32494, 0.588301] with:

Minimum: 0.32494 Maximum: 1.559414 Average: 0.7993 seconds

Number of cars 2

bbox:car_number ((895, 415), (942, 459)) : 1

bbox:car_number ((1136, 446), (1242, 518)) : 2

The minimum distance from car: 3 is 16.1245154965971

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518))]

Length of task list: 5

Number of processes used: 3

60% | 754/1261 [33:37<22:36, 2.68s/it]

The times for each task are: [0.631678, 0.937858, 1.462869, 0.381381, 0.451283] with:

Minimum: 0.381381 Maximum: 1.462869 Average: 0.773 seconds

Number of cars 6

bbox:car_number ((894, 400), (954, 462)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1113, 431), (1182, 485)) : 2

The minimum distance from car: 3 is 35.608987629529715

bbox:car_number ((1102, 476), (1105, 485)) : 3

bbox:car_number ((1090, 514), (1105, 518)) : 4

bbox:car_number ((1075, 520), (1105, 535)) : 5

bbox:car_number ((1120, 535), (1122, 539)) : 6

totalCars: 6

((875, 400), ((1142, 1472)), ((10754, 4272)), ((11375, 4850)), ((9425, 14720)), ((18753, 4005)), ((9341, 05472))
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3

60% | 755/1261 [33:40<22:34, 2.68s/it]

The times for each task are: [0.964243, 0.666477, 1.690254, 0.413227, 0.418396] with:

Minimum: 0.413227 Maximum: 1.690254 Average: 0.8305 seconds

Number of cars 2
bbox:car_number ((895, 400), (954, 459)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1090, 445), (1230, 529)) : 2
The minimum distance from car: 3 is 29.427877939124322
totalCars: 6
((875, 400), ((1142, 1472)), ((10754, 4272)), ((11375, 4850)), ((9425, 14720)), ((18753, 4005)), ((9341, 05472))
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

60% | 756/1261 [33:42<22:31, 2.68s/it]

The times for each task are: [0.622302, 1.386242, 0.98629, 0.353711, 0.433547] with:

Minimum: 0.353711 Maximum: 1.386242 Average: 0.7564 seconds

Number of cars 4
bbox:car_number ((895, 400), (957, 462)) : 1
The minimum distance from car: 1 is 2.8284271247461903
bbox:car_number ((1000, 415), (1070, 472)) : 2
The minimum distance from car: 1 is 22.135943621178654
bbox:car_number ((1090, 431), (1214, 529)) : 3
The minimum distance from car: 3 is 10.63014581273465
bbox:car_number ((1180, 520), (1180, 529)) : 4
totalCars: 6
((875, 400), ((1142, 1472)), ((10754, 4272)), ((11375, 4850)), ((9425, 14720)), ((18753, 4005)), ((9341, 05472))
totalCars: 6
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5

Number of processes used: 3

60% | 757/1261 [33:45<22:28, 2.68s/it]

The times for each task are: [0.921345, 0.701699, 1.385903, 0.396963, 0.332687] with:

Minimum: 0.332687 Maximum: 1.385903 Average: 0.7477 seconds

Number of cars 2

bbox:car_number ((895, 400), (957, 462)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1048, 431), (1238, 552)) : 2

The minimum distance from car: 3 is 14.212670403551895

totalCars: 6

((72)), ((875, 410), Num((842, 472)), Pos((875, 400), ((1075, 4272), (1137, 4850), ((9425, 14720), ((18733, 4495), (9341, 05472)40

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

60% | 758/1261 [33:48<22:25, 2.68s/it]

The times for each task are: [0.940896, 0.665939, 1.533506, 0.340437, 0.500701] with:

Minimum: 0.340437 Maximum: 1.533506 Average: 0.7963 seconds

Number of cars 3

bbox:car_number ((910, 400), (957, 459)) : 1

bbox:car_number ((1051, 445), (1149, 552)) : 2

The minimum distance from car: 3 is 18.027756377319946

bbox:car_number ((1165, 446), (1185, 477)) : 3

totalCars: 6

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44

Length of task list: 5

Number of processes used: 3

60% | 759/1261 [33:51<22:23, 2.68s/it]

The times for each task are: [0.651819, 0.899395, 1.425237, 0.287132, 0.435395] with:

Minimum: 0.287132 Maximum: 1.425237 Average: 0.7398 seconds

Number of cars 2

```
bbox:car_number ((895, 400), (971, 472)) : 1
The minimum distance from car: 1 is 8.602325267042627
bbox:car_number ((1084, 427), (1214, 518)) : 2
The minimum distance from car: 3 is 14.142135623730951
totalCars: 6
((875, 400), ((1075, 400), (1075, 472)), ((1037, 485), (940, 472)), ((1875, 400), (934, 472)), ((1075, 400), (1075, 472)), ((1037, 485), (940, 472)), ((1875, 400), (934, 472)))
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

60% | 760/1261 [33:53<22:20, 2.68s/it]

The times for each task are: [1.011699, 1.361729, 0.698128, 0.457754, 0.313776] with:
Minimum: 0.313776 Maximum: 1.361729 Average: 0.7686 seconds

```
Number of cars 4
bbox:car_number ((905, 400), (972, 472)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1067, 431), (1227, 518)) : 2
The minimum distance from car: 3 is 2.8284271247461903
bbox:car_number ((1021, 492), (1025, 539)) : 3
bbox:car_number ((1180, 520), (1211, 522)) : 4
totalCars: 6
((875, 400), ((1075, 400), (1075, 472)), ((1037, 485), (940, 472)), ((1875, 400), (934, 472)), ((1075, 400), (1075, 472)), ((1037, 485), (940, 472)), ((1875, 400), (934, 472)))
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

60% | 761/1261 [33:56<22:17, 2.68s/it]

The times for each task are: [0.607975, 1.009254, 1.329228, 0.38124, 0.296169] with:
Minimum: 0.296169 Maximum: 1.329228 Average: 0.7248 seconds

```
Number of cars 6
bbox:car_number ((910, 400), (957, 462)) : 1
bbox:car_number ((1030, 400), (1214, 518)) : 2
The minimum distance from car: 3 is 29.154759474226502
bbox:car_number ((1046, 415), (1047, 459)) : 3
bbox:car_number ((1020, 454), (1025, 459)) : 4
bbox:car_number ((1020, 462), (1025, 462)) : 5
bbox:car_number ((1021, 492), (1025, 498)) : 6
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

60% | 762/1261 [33:59<22:15, 2.68s/it]

The times for each task are: [0.829883, 0.70067, 0.390082, 1.365769, 0.325582] with:

Minimum: 0.325582 Maximum: 1.365769 Average: 0.7224 seconds

```
Number of cars 2
bbox:car_number ((910, 400), (972, 472)) : 1
```

The minimum distance from car: 1 is 3.0

```
bbox:car_number ((1048, 400), (1212, 566)) : 2
```

The minimum distance from car: 3 is 15.264337522473747

```
totalCars: 6
```

```
((875, 400), ((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
```

Number of processes used: 3

61% | 763/1261 [34:02<22:12, 2.68s/it]

The times for each task are: [1.396398, 0.933103, 0.720265, 0.37074, 0.442333] with:

Minimum: 0.37074 Maximum: 1.396398 Average: 0.7726 seconds

```
Number of cars 2
bbox:car_number ((905, 400), (972, 472)) : 1
```

The minimum distance from car: 1 is 3.0

```
bbox:car_number ((1020, 400), (1212, 566)) : 2
```

The minimum distance from car: 3 is 14.0

```
totalCars: 6
```

```
((875, 400), ((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
```

Number of processes used: 3

61% | 764/1261 [34:04<22:10, 2.68s/it]

The times for each task are: [0.909969, 0.612401, 1.386262, 0.283006, 0.381664] with:

Minimum: 0.283006 Maximum: 1.386262 Average: 0.7147 seconds

Number of cars 2

bbox:car_number ((910, 400), (972, 472)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((1045, 415), (1212, 539)) : 2

The minimum distance from car: 3 is 13.416407864998739

totalCars: 6

((72)), ((875, 400), Num(1842, 1472)Car, Pos(1875, 400D)((10754, 4272)), (11375, 4850)), ((9405, 147200)), (((18753, 4005)),), (9341, 05472)400

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

61% | 765/1261 [34:07<22:07, 2.68s/it]

The times for each task are: [0.894109, 1.396228, 0.696249, 0.381245, 0.29067] with:

Minimum: 0.29067 Maximum: 1.396228 Average: 0.7317 seconds

Number of cars 3

bbox:car_number ((910, 400), (972, 462)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1021, 400), (1211, 554)) : 2

The minimum distance from car: 3 is 12.0

bbox:car_number ((1021, 462), (1025, 462)) : 3

totalCars: 6

((72)), ((875, 400), Num(1842, 1472)Car, Pos(1875, 400D)((10754, 4272)), (11375, 4850)), ((9405, 147200)), (((18753, 4005)),), (9341, 05472)400

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

61% | 766/1261 [34:10<22:05, 2.68s/it]

The times for each task are: [0.558327, 0.880458, 1.583844, 0.350061, 0.461197] with:

Minimum: 0.350061 Maximum: 1.583844 Average: 0.7668 seconds

Number of cars 3

bbox:car_number ((895, 400), (977, 477)) : 1

The minimum distance from car: 1 is 8.602325267042627

bbox:car_number ((985, 400), (1211, 539)) : 2

```
The minimum distance from car: 3 is 19.697715603592208
bbox:car_number ((1015, 475), (1017, 477)) : 3
totalCars: 6
72)), ((875, 400), Num(842, 1472)), Car, Pos(875, 400), ((10754, 4272)), , ((10375, 4850)), , ((9425, 14720)), , ((18753, 4495)), ,(9341, 05472)40
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

61% | 767/1261 [34:13<22:02, 2.68s/it]

The times for each task are: [1.045051, 0.625372, 1.383262, 0.48852, 0.348309] with:

Minimum: 0.348309 Maximum: 1.383262 Average: 0.7781 seconds

```
Number of cars 3
bbox:car_number ((910, 400), (972, 472)) : 1
The minimum distance from car: 1 is 5.385164807134504
bbox:car_number ((985, 400), (1208, 560)) : 2
The minimum distance from car: 3 is 11.180339887498949
bbox:car_number ((975, 454), (977, 472)) : 3
totalCars: 6
72)), ((875, 400), Num(842, 1472)), Car, Pos(875, 400), ((10754, 4272)), , ((10375, 4850)), , ((9425, 14720)), , ((18753, 4495)), ,(9341, 05472)40
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

61% | 768/1261 [34:16<21:59, 2.68s/it]

The times for each task are: [0.620583, 1.334214, 0.978404, 0.398812, 0.285684] with:

Minimum: 0.285684 Maximum: 1.334214 Average: 0.7235 seconds

```
Number of cars 3
bbox:car_number ((910, 400), (971, 462)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1030, 400), (1197, 518)) : 2
The minimum distance from car: 3 is 9.0
bbox:car_number ((1020, 446), (1032, 462)) : 3
totalCars: 6
72)), ((875, 400), Num(842, 1472)), Car, Pos(875, 400), ((10754, 4272)), , ((10375, 4850)), , ((9425, 14720)), , ((18753, 4495)), ,(9341, 05472)40
totalCars: 6
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
```

Number of processes used: 3

61% | 769/1261 [34:18<21:57, 2.68s/it]

The times for each task are: [0.616308, 0.934918, 1.424075, 0.311591, 0.396135] with:

Minimum: 0.311591 Maximum: 1.424075 Average: 0.7366 seconds

Number of cars 2

bbox:car_number ((905, 400), (972, 478)) : 1

The minimum distance from car: 1 is 8.246211251235321

bbox:car_number ((989, 431), (1197, 552)) : 2

The minimum distance from car: 3 is 9.899494936611665

totalCars: 6

((875, 400) Num(842, 1472)ar ,Pos(875ons400)((10754, 4272)),,(11375,4850)),, ((9405,1,4720)),, (((18753, 4005)),,(9341,05472)400

totalCars: 6

Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44

Length of task list: 5

Number of processes used: 3

61% | 770/1261 [34:21<21:54, 2.68s/it]

The times for each task are: [0.839363, 1.402022, 0.691041, 0.422485, 0.393265] with:

Minimum: 0.393265 Maximum: 1.402022 Average: 0.7496 seconds

Number of cars 2

bbox:car_number ((905, 400), (972, 472)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((998, 400), (1197, 552)) : 2

The minimum distance from car: 3 is 4.123105625617661

totalCars: 6

((875, 400) Num(842, 1472)ar ,Pos(875ons400)((10754, 4272)),,(11375,4850)),, ((9405,1,4720)),, (((18753, 4005)),,(9341,05472)400

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

61% | 771/1261 [34:24<21:52, 2.68s/it]

The times for each task are: [0.908308, 1.50206, 0.700852, 0.405766, 0.309021] with:

Minimum: 0.309021 Maximum: 1.50206 Average: 0.7652 seconds

```
Number of cars 4
bbox:car_number ((905, 400), (971, 472)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((998, 431), (1133, 560)) : 2
The minimum distance from car: 3 is 28.284271247461902
bbox:car_number ((1135, 445), (1137, 462)) : 3
bbox:car_number ((1150, 460), (1197, 516)) : 4
totalCars: 6
((875, 400), ((1084, 412), 1472)), Pos(875, 400), ((1084, 412), 1472), , ((1137, 5, 48500)), , ((9425, 1, 47200)), , ((18133, 4005)), , (9341, 0, 5472)400
totalCars: 6
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44
Length of task list: 5
Number of processes used: 3
```

61% | 772/1261 [34:27<21:49, 2.68s/it]

The times for each task are: [1.39411, 0.866355, 0.71189, 0.339014, 0.400323] with:

Minimum: 0.339014 Maximum: 1.39411 Average: 0.7423 seconds

```
Number of cars 2
bbox:car_number ((910, 400), (972, 462)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1008, 419), (1162, 541)) : 2
The minimum distance from car: 3 is 12.649110640673518
totalCars: 6
((875, 400), ((1084, 412), 1472)), Pos(875, 400), ((1084, 412), 1472), , ((1137, 5, 48500)), , ((9425, 1, 47200)), , ((18133, 4005)), , (9341, 0, 5472)400
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

61% | 773/1261 [34:30<21:46, 2.68s/it]

The times for each task are: [0.570235, 0.90204, 1.556378, 0.438061, 0.306526] with:

Minimum: 0.306526 Maximum: 1.556378 Average: 0.7546 seconds

```
Number of cars 2
bbox:car_number ((910, 400), (972, 462)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((994, 415), (1180, 560)) : 2
The minimum distance from car: 3 is 7.280109889280518
totalCars: 6
```

```
:72)), ((875, 410) Num(842, 1472) r, Pos(875, 1400), (10754, 4272)), (11337, 5, 4850)), ((9105, 1, 4740)), (((18153, 4095)), ), (9341, 0, 5472) 40  
totalCars: 6  
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44  
Length of task list: 5  
Number of processes used: 3
```

61% | 774/1261 [34:33<21:44, 2.68s/it]

The times for each task are: [1.080494, 0.659304, 1.624002, 0.356865, 0.501125] with:

Minimum: 0.356865 Maximum: 1.624002 Average: 0.8444 seconds

Number of cars 2

bbox:car_number ((910, 400), (972, 462)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((989, 415), (1180, 554)) : 2

The minimum distance from car: 3 is 4.242640687119285

totalCars: 6

400 Num 942: 1

totalCars: 6

Car Number:

Car Number: 3 Car Positions: [(1111, 110),
Length of task list: 5

Number of processes used

Number of processes used: 3

61% | 775/1261 [34:35<21:41, 2.68s/it]

The times for each task are: [0.615379, 1.412146, 0.880489, 0.286576, 0.378489] with:

Minimum: 0.286576 Maximum: 1.412146 Average: 0.7146 seconds

Number of cars 2

bbox:car_number ((910, 400), (972, 462)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((998, 400), (1182, 554)) : 2

The minimum distance from car: 3 is 9.219544457292887

totalCars: 6

(4-20) Num(842) 1

totalCars: 6

Car Number:

Car Number: 3 Car Positions: [(1141, 415),
Length of task list: 5

Length of task list: 3
Number of processes used:

Number of processes used: 3

62% | 776/1261 [34:38<21:38, 2.68s/it]

The times for each task are: [0.876051, 0.611766, 1.533399, 0.430093, 0.314276] with:

Minimum: 0.314276 Maximum: 1.533399 Average: 0.7531 seconds

Number of cars 3

bbox:car_number ((910, 400), (972, 472)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1008, 400), (1182, 560)) : 2

The minimum distance from car: 3 is 5.830951894845301

bbox:car_number ((989, 462), (1001, 462)) : 3

totalCars: 6

((875, 400), ((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

62% | 777/1261 [34:41<21:36, 2.68s/it]

The times for each task are: [0.847479, 0.681795, 1.609721, 0.365795, 0.463971] with:

Minimum: 0.365795 Maximum: 1.609721 Average: 0.7938 seconds

Number of cars 2

bbox:car_number ((913, 415), (977, 472)) : 1

The minimum distance from car: 1 is 8.06225774829855

bbox:car_number ((985, 415), (1181, 554)) : 2

The minimum distance from car: 3 is 12.649110640673518

totalCars: 6

((875, 400), ((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

62% | 778/1261 [34:44<21:33, 2.68s/it]

The times for each task are: [0.847434, 0.614552, 1.460248, 0.311177, 0.496556] with:

Minimum: 0.311177 Maximum: 1.460248 Average: 0.746 seconds

Number of cars 2

bbox:car_number ((910, 400), (977, 472)) : 1

The minimum distance from car: 1 is 7.280109889280518

bbox:car_number ((985, 427), (1167, 539)) : 2

```
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 6
((875, 400), ((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

62% | 779/1261 [34:46<21:31, 2.68s/it]

```
The times for each task are: [0.610563, 0.859595, 1.398253, 0.437439, 0.302129] with:
Minimum: 0.302129 Maximum: 1.398253 Average: 0.7216 seconds
```

```
Number of cars 2
bbox:car_number ((913, 415), (972, 462)) : 1
The minimum distance from car: 1 is 2.23606797749979
bbox:car_number ((994, 427), (1160, 552)) : 2
The minimum distance from car: 3 is 6.082762530298219
totalCars: 6
((875, 400), ((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

62% | 780/1261 [34:49<21:28, 2.68s/it]

```
The times for each task are: [1.392159, 0.684329, 0.89785, 0.357433, 0.464179] with:
Minimum: 0.357433 Maximum: 1.392159 Average: 0.7592 seconds
```

```
Number of cars 2
bbox:car_number ((913, 400), (972, 459)) : 1
The minimum distance from car: 1 is 9.0
bbox:car_number ((994, 415), (1160, 539)) : 2
The minimum distance from car: 3 is 12.0
totalCars: 6
((875, 400), ((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

62%| 781/1261 [34:51<21:25, 2.68s/it]

The times for each task are: [0.933629, 1.313084, 0.718576, 0.2864, 0.436693] with:

Minimum: 0.2864 Maximum: 1.313084 Average: 0.7377 seconds

Number of cars 5

bbox:car_number ((985, 400), (1167, 560)) : 1

The minimum distance from car: 3 is 3.1622776601683795

bbox:car_number ((970, 419), (972, 447)) : 2

bbox:car_number ((970, 457), (972, 459)) : 3

bbox:car_number ((989, 462), (991, 478)) : 4

bbox:car_number ((989, 490), (991, 498)) : 5

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

62%| 782/1261 [34:54<21:23, 2.68s/it]

The times for each task are: [0.977966, 0.613876, 0.295826, 0.453987, 1.379842] with:

Minimum: 0.295826 Maximum: 1.379842 Average: 0.7443 seconds

Number of cars 2

bbox:car_number ((970, 400), (1149, 554)) : 1

The minimum distance from car: 2 is 12.041594578792296

bbox:car_number ((913, 415), (957, 447)) : 2

totalCars: 6

Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539)), ((970, 400),

Length of task list: 5

Number of processes used: 3

62%| 783/1261 [34:57<21:20, 2.68s/it]

The times for each task are: [0.957849, 0.687504, 0.430553, 1.424587, 0.296249] with:

Minimum: 0.296249 Maximum: 1.424587 Average: 0.7593 seconds

Number of cars 2

bbox:car_number ((910, 400), (957, 459)) : 1

bbox:car_number ((989, 400), (1152, 560)) : 2

The minimum distance from car: 3 is 6.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

```
Length of task list: 5
Number of processes used: 3
```

```
62%| 784/1261 [34:59<21:17, 2.68s/it]
```

```
The times for each task are: [0.98635, 1.683715, 0.636096, 0.375196, 0.417931] with:
```

```
Minimum: 0.375196 Maximum: 1.683715 Average: 0.8199 seconds
```

```
Number of cars 2
bbox:car_number ((989, 400), (1149, 529)) : 1
The minimum distance from car: 3 is 16.0312195418814
bbox:car_number ((970, 415), (977, 447)) : 2
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
62%| 785/1261 [35:02<21:14, 2.68s/it]
```

```
The times for each task are: [0.900777, 1.629199, 0.731034, 0.405922, 0.314737] with:
```

```
Minimum: 0.314737 Maximum: 1.629199 Average: 0.7963 seconds
```

```
Number of cars 2
bbox:car_number ((975, 400), (1160, 537)) : 1
The minimum distance from car: 3 is 4.47213595499958
bbox:car_number ((970, 427), (972, 447)) : 2
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
62%| 786/1261 [35:05<21:12, 2.68s/it]
```

```
The times for each task are: [0.618692, 0.904891, 1.667022, 0.333369, 0.40467] with:
```

```
Minimum: 0.333369 Maximum: 1.667022 Average: 0.7857 seconds
```

```
Number of cars 1
bbox:car_number ((967, 400), (1152, 529)) : 1
The minimum distance from car: 3 is 8.94427190999916
totalCars: 6
```

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))] Length of task list: 5 Number of processes used: 3

62%| 787/1261 [35:07<21:09, 2.68s/it]

The times for each task are: [0.622404, 1.424894, 0.995919, 0.304072, 0.407526] with:

Minimum: 0.304072 Maximum: 1.424894 Average: 0.751 seconds

Number of cars 2
bbox:car_number ((970, 400), (1152, 552)) : 1
The minimum distance from car: 2 is 2.23606797749979
bbox:car_number ((565, 505), (597, 522)) : 2
totalCars: 6
Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539)), ((970, 400), (1141, 415))] Length of task list: 5 Number of processes used: 3

62%| 788/1261 [35:09<21:06, 2.68s/it]

The times for each task are: [0.839437, 0.665189, 1.342164, 0.476649, 0.385515] with:

Minimum: 0.385515 Maximum: 1.342164 Average: 0.7418 seconds

Number of cars 1
bbox:car_number ((951, 400), (1137, 529)) : 1
The minimum distance from car: 3 is 15.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))] Length of task list: 5 Number of processes used: 3

63%| 789/1261 [35:12<21:04, 2.68s/it]

The times for each task are: [0.910945, 0.688067, 1.599645, 0.408985, 0.303575] with:

Minimum: 0.303575 Maximum: 1.599645 Average: 0.7822 seconds

Number of cars 2
bbox:car_number ((989, 431), (1143, 529)) : 1
The minimum distance from car: 2 is 6.4031242374328485
bbox:car_number ((989, 460), (1010, 478)) : 2

```
totalCars: 6
Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539)), ((970, 400),
Length of task list: 5
Number of processes used: 3
```

63%| 790/1261 [35:15<21:01, 2.68s/it]

The times for each task are: [0.964948, 0.732948, 1.363712, 0.401532, 0.389192] with:

Minimum: 0.389192 Maximum: 1.363712 Average: 0.7705 seconds

```
Number of cars 2
bbox:car_number ((951, 415), (957, 447)) : 1
bbox:car_number ((985, 427), (1143, 552)) : 2
The minimum distance from car: 3 is 6.082762530298219
totalCars: 6
Car Number: 3 Car Positions: [((1195, 481), (1253, 537)), ((1144, 446), (1257, 560)), ((1102, 44
Length of task list: 5
Number of processes used: 3
```

63%| 791/1261 [35:18<20:58, 2.68s/it]

The times for each task are: [0.883497, 0.626338, 1.560543, 0.398289, 0.343999] with:

Minimum: 0.343999 Maximum: 1.560543 Average: 0.7625 seconds

```
Number of cars 1
bbox:car_number ((967, 415), (1143, 529)) : 1
The minimum distance from car: 2 is 13.601470508735444
totalCars: 6
Car Number: 2 Car Positions: [((967, 400), (1077, 560)), ((994, 400), (1106, 539)), ((970, 400),
Length of task list: 5
Number of processes used: 3
```

63%| 792/1261 [35:21<20:56, 2.68s/it]

The times for each task are: [0.991172, 1.524732, 0.662677, 0.433224, 0.353179] with:

Minimum: 0.353179 Maximum: 1.524732 Average: 0.793 seconds

```
Number of cars 1
bbox:car_number ((958, 400), (1139, 529)) : 1
The minimum distance from car: 3 is 4.0
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1137, 535))]
Length of task list: 5
Number of processes used: 3
```

63%| 793/1261 [35:24<20:53, 2.68s/it]

The times for each task are: [0.640502, 1.516207, 1.113878, 0.472184, 0.391449] with:

Minimum: 0.391449 Maximum: 1.516207 Average: 0.8268 seconds

```
Number of cars 1
bbox:car_number ((955, 400), (1137, 535)) : 1
The minimum distance from car: 3 is 3.605551275463989
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1137, 535))]
Length of task list: 5
Number of processes used: 3
```

63%| 794/1261 [35:26<20:50, 2.68s/it]

The times for each task are: [0.596541, 0.956425, 1.553378, 0.414201, 0.314955] with:

Minimum: 0.314955 Maximum: 1.553378 Average: 0.7671 seconds

```
Number of cars 2
bbox:car_number ((955, 400), (1137, 535)) : 1
The minimum distance from car: 3 is 0.0
bbox:car_number ((910, 446), (927, 462)) : 2
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1137, 535))]
Length of task list: 5
Number of processes used: 3
```

63%| 795/1261 [35:29<20:48, 2.68s/it]

The times for each task are: [0.683754, 0.858311, 1.441182, 0.351026, 0.452911] with:

Minimum: 0.351026 Maximum: 1.441182 Average: 0.7574 seconds

```
Number of cars 1
bbox:car_number ((951, 400), (1133, 529)) : 1
The minimum distance from car: 3 is 5.0
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

63%| 796/1261 [35:32<20:45, 2.68s/it]

The times for each task are: [0.889099, 0.648314, 1.382038, 0.441313, 0.328582] with:

Minimum: 0.328582 Maximum: 1.382038 Average: 0.7379 seconds

```
Number of cars 2
bbox:car_number ((940, 400), (1122, 539)) : 1
The minimum distance from car: 3 is 12.083045973594572
bbox:car_number ((913, 457), (927, 462)) : 2
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

63%| 797/1261 [35:34<20:42, 2.68s/it]

The times for each task are: [0.956867, 0.606866, 0.478881, 0.314094, 1.711338] with:

Minimum: 0.314094 Maximum: 1.711338 Average: 0.8136 seconds

```
Number of cars 1
bbox:car_number ((955, 400), (1118, 539)) : 1
The minimum distance from car: 3 is 5.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

63%| 798/1261 [35:37<20:39, 2.68s/it]

The times for each task are: [0.886564, 0.61081, 0.307679, 1.448107, 0.399457] with:

Minimum: 0.307679 Maximum: 1.448107 Average: 0.7305 seconds

```
Number of cars 1
bbox:car_number ((958, 400), (1118, 552)) : 1
The minimum distance from car: 3 is 7.280109889280518
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

63%| 799/1261 [35:39<20:37, 2.68s/it]

The times for each task are: [0.996676, 0.61178, 1.370271, 0.344744, 0.397702] with:

Minimum: 0.344744 Maximum: 1.370271 Average: 0.7442 seconds

```
Number of cars 1
bbox:car_number ((955, 400), (1122, 537)) : 1
The minimum distance from car: 3 is 8.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

63%| 800/1261 [35:42<20:34, 2.68s/it]

The times for each task are: [0.697707, 0.861233, 1.368966, 0.501005, 0.290011] with:

Minimum: 0.290011 Maximum: 1.368966 Average: 0.7438 seconds

```
Number of cars 1
bbox:car_number ((952, 419), (1122, 529)) : 1
The minimum distance from car: 3 is 6.082762530298219
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 801/1261 [35:45<20:31, 2.68s/it]

The times for each task are: [0.855437, 0.640851, 1.412615, 0.433488, 0.41777] with:

Minimum: 0.41777 Maximum: 1.412615 Average: 0.752 seconds

```
Number of cars 2
bbox:car_number ((951, 415), (1122, 535)) : 1
The minimum distance from car: 3 is 1.4142135623730951
bbox:car_number ((940, 454), (942, 462)) : 2
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 802/1261 [35:47<20:29, 2.68s/it]

The times for each task are: [0.903336, 0.612642, 1.521257, 0.39405, 0.299678] with:

Minimum: 0.299678 Maximum: 1.521257 Average: 0.7462 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1093, 537)) : 1
The minimum distance from car: 3 is 21.18962010041709
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 803/1261 [35:51<20:26, 2.68s/it]

The times for each task are: [0.622902, 0.904433, 0.420286, 1.652713, 0.362518] with:

Minimum: 0.362518 Maximum: 1.652713 Average: 0.7926 seconds

```
Number of cars 1
bbox:car_number ((940, 400), (1107, 535)) : 1
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

64%| 804/1261 [35:53<20:24, 2.68s/it]

The times for each task are: [0.911508, 1.350722, 0.672002, 0.383344, 0.4156] with:

Minimum: 0.383344 Maximum: 1.350722 Average: 0.7466 seconds

```
Number of cars 1
bbox:car_number ((940, 415), (1107, 529)) : 1
The minimum distance from car: 3 is 5.0
totalCars: 6
```

```
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

```
64%| 805/1261 [35:56<20:21, 2.68s/it]
```

```
The times for each task are: [0.874357, 0.698201, 1.647812, 0.470617, 0.389213] with:
```

```
Minimum: 0.389213 Maximum: 1.647812 Average: 0.816 seconds
```

```
Number of cars 1
bbox:car_number ((932, 400), (1107, 535)) : 1
The minimum distance from car: 3 is 6.4031242374328485
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

```
64%| 806/1261 [35:58<20:18, 2.68s/it]
```

```
The times for each task are: [0.942908, 1.407033, 0.630253, 0.496652, 0.345977] with:
```

```
Minimum: 0.345977 Maximum: 1.407033 Average: 0.7646 seconds
```

```
Number of cars 1
bbox:car_number ((913, 400), (1107, 539)) : 1
The minimum distance from car: 3 is 9.219544457292887
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

```
64%| 807/1261 [36:01<20:15, 2.68s/it]
```

```
The times for each task are: [0.581059, 0.90941, 1.456262, 0.381617, 0.285041] with:
```

```
Minimum: 0.285041 Maximum: 1.456262 Average: 0.7227 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (1106, 539)) : 1
The minimum distance from car: 3 is 5.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
64%| 808/1261 [36:04<20:13, 2.68s/it]
```

```
The times for each task are: [0.910273, 0.62754, 0.403125, 1.696172, 0.298015] with:
```

```
Minimum: 0.298015 Maximum: 1.696172 Average: 0.787 seconds
```

```
Number of cars 1
bbox:car_number ((925, 400), (1106, 539)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
64%| 809/1261 [36:06<20:10, 2.68s/it]
```

```
The times for each task are: [0.566562, 0.917039, 1.401814, 0.308252, 0.48301] with:
```

```
Minimum: 0.308252 Maximum: 1.401814 Average: 0.7353 seconds
```

```
Number of cars 1
bbox:car_number ((940, 400), (1105, 529)) : 1
The minimum distance from car: 3 is 8.602325267042627
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
64%| 810/1261 [36:09<20:07, 2.68s/it]
```

```
The times for each task are: [0.896013, 0.641877, 1.43219, 0.450355, 0.289383] with:
```

```
Minimum: 0.289383 Maximum: 1.43219 Average: 0.742 seconds
```

```
Number of cars 1
bbox:car_number ((940, 400), (1092, 535)) : 1
The minimum distance from car: 3 is 6.708203932499369
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
```

Number of processes used: 3

64%| 811/1261 [36:11<20:05, 2.68s/it]

The times for each task are: [0.902405, 1.37148, 0.714441, 0.412971, 0.336097] with:

Minimum: 0.336097 Maximum: 1.37148 Average: 0.7475 seconds

Number of cars 1

bbox:car_number ((927, 400), (1092, 535)) : 1

The minimum distance from car: 3 is 7.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

64%| 812/1261 [36:14<20:02, 2.68s/it]

The times for each task are: [0.612605, 1.550999, 1.013817, 0.474827, 0.331315] with:

Minimum: 0.331315 Maximum: 1.550999 Average: 0.7967 seconds

Number of cars 2

bbox:car_number ((951, 400), (1092, 535)) : 1

The minimum distance from car: 3 is 12.0

bbox:car_number ((940, 460), (942, 507)) : 2

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

64%| 813/1261 [36:17<19:59, 2.68s/it]

The times for each task are: [0.984028, 0.617244, 1.382511, 0.463685, 0.327587] with:

Minimum: 0.327587 Maximum: 1.382511 Average: 0.755 seconds

Number of cars 1

bbox:car_number ((925, 400), (1087, 529)) : 1

The minimum distance from car: 3 is 15.297058540778355

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 814/1261 [36:19<19:56, 2.68s/it]

The times for each task are: [0.905629, 0.654824, 0.406019, 1.372198, 0.396415] with:

Minimum: 0.396415 Maximum: 1.372198 Average: 0.747 seconds

Number of cars 1

bbox:car_number ((927, 400), (1087, 535)) : 1

The minimum distance from car: 3 is 3.1622776601683795

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 815/1261 [36:22<19:54, 2.68s/it]

The times for each task are: [0.749663, 0.912853, 0.345155, 0.567352, 1.646894] with:

Minimum: 0.345155 Maximum: 1.646894 Average: 0.8444 seconds

Number of cars 1

bbox:car_number ((925, 400), (1087, 529)) : 1

The minimum distance from car: 3 is 3.1622776601683795

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 816/1261 [36:25<19:51, 2.68s/it]

The times for each task are: [0.847844, 0.643205, 0.310639, 1.424122, 0.443817] with:

Minimum: 0.310639 Maximum: 1.424122 Average: 0.7339 seconds

Number of cars 1

bbox:car_number ((940, 400), (1077, 529)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 817/1261 [36:28<19:49, 2.68s/it]

The times for each task are: [0.670726, 0.862778, 1.39663, 0.297107, 0.434957] with:

Minimum: 0.297107 Maximum: 1.39663 Average: 0.7324 seconds

Number of cars 1

bbox:car_number ((925, 400), (1086, 535)) : 1

The minimum distance from car: 3 is 4.242640687119285

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 818/1261 [36:30<19:46, 2.68s/it]

The times for each task are: [0.876292, 0.620825, 0.410491, 1.472386, 0.340284] with:

Minimum: 0.340284 Maximum: 1.472386 Average: 0.7441 seconds

Number of cars 1

bbox:car_number ((927, 400), (1086, 529)) : 1

The minimum distance from car: 3 is 3.1622776601683795

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 819/1261 [36:33<19:43, 2.68s/it]

The times for each task are: [0.553386, 0.369148, 0.976307, 1.522579, 0.302256] with:

Minimum: 0.302256 Maximum: 1.522579 Average: 0.7447 seconds

Number of cars 1

bbox:car_number ((927, 400), (1079, 529)) : 1

The minimum distance from car: 3 is 3.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 820/1261 [36:35<19:40, 2.68s/it]

The times for each task are: [0.63249, 1.460898, 0.995929, 0.441721, 0.374334] with:

Minimum: 0.374334 Maximum: 1.460898 Average: 0.7811 seconds

Number of cars 1

bbox:car_number ((925, 400), (1079, 539)) : 1

The minimum distance from car: 3 is 5.0990195135927845

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 821/1261 [36:38<19:38, 2.68s/it]

The times for each task are: [0.871838, 1.338263, 0.727218, 0.300833, 0.459105] with:

Minimum: 0.300833 Maximum: 1.338263 Average: 0.7395 seconds

Number of cars 2

bbox:car_number ((929, 400), (1077, 518)) : 1

The minimum distance from car: 3 is 10.04987562112089

bbox:car_number ((927, 460), (927, 462)) : 2

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 822/1261 [36:41<19:35, 2.68s/it]

The times for each task are: [0.574555, 0.913489, 1.504209, 0.304294, 0.460554] with:

Minimum: 0.304294 Maximum: 1.504209 Average: 0.7514 seconds

Number of cars 1

bbox:car_number ((927, 400), (1077, 522)) : 1

The minimum distance from car: 3 is 2.23606797749979

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 823/1261 [36:43<19:32, 2.68s/it]

The times for each task are: [0.878189, 0.61477, 0.290428, 1.626165, 0.395625] with:

Minimum: 0.290428 Maximum: 1.626165 Average: 0.761 seconds

Number of cars 1

bbox:car_number ((927, 400), (1077, 529)) : 1

The minimum distance from car: 3 is 3.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 824/1261 [36:46<19:30, 2.68s/it]

The times for each task are: [1.365275, 0.856082, 0.731511, 0.298022, 0.426667] with:

Minimum: 0.298022 Maximum: 1.365275 Average: 0.7355 seconds

Number of cars 1

bbox:car_number ((913, 400), (1067, 522)) : 1

The minimum distance from car: 3 is 12.36931687685298

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

65%| 825/1261 [36:49<19:27, 2.68s/it]

The times for each task are: [0.714546, 0.908093, 1.654923, 0.432624, 0.362545] with:

Minimum: 0.362545 Maximum: 1.654923 Average: 0.8145 seconds

Number of cars 1

bbox:car_number ((913, 400), (1056, 518)) : 1

The minimum distance from car: 3 is 6.324555320336759

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

66%| 826/1261 [36:51<19:24, 2.68s/it]

The times for each task are: [0.87564, 0.632278, 1.587649, 0.295483, 0.434352] with:

Minimum: 0.295483 Maximum: 1.587649 Average: 0.7651 seconds

Number of cars 1

bbox:car_number ((927, 400), (1052, 512)) : 1

The minimum distance from car: 3 is 5.830951894845301

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

66%| 827/1261 [36:54<19:22, 2.68s/it]

The times for each task are: [0.932901, 1.434639, 0.625593, 0.387496, 0.285279] with:

Minimum: 0.285279 Maximum: 1.434639 Average: 0.7332 seconds

Number of cars 1

bbox:car_number ((895, 400), (1067, 522)) : 1

The minimum distance from car: 3 is 9.433981132056603

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

66%| 828/1261 [36:57<19:19, 2.68s/it]

The times for each task are: [1.369517, 0.646167, 0.43893, 0.884287, 0.290154] with:

Minimum: 0.290154 Maximum: 1.369517 Average: 0.7258 seconds

Number of cars 1

bbox:car_number ((913, 400), (1062, 535)) : 1

The minimum distance from car: 3 is 8.48528137423857

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

66%| 829/1261 [36:59<19:16, 2.68s/it]

The times for each task are: [0.897871, 0.578099, 1.61033, 0.447391, 0.326173] with:

Minimum: 0.326173 Maximum: 1.61033 Average: 0.772 seconds

Number of cars 1
bbox:car_number ((913, 400), (1056, 529)) : 1
The minimum distance from car: 3 is 4.242640687119285
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

66%| 830/1261 [37:02<19:13, 2.68s/it]

The times for each task are: [0.615417, 1.399055, 0.996871, 0.408139, 0.278992] with:

Minimum: 0.278992 Maximum: 1.399055 Average: 0.7397 seconds

Number of cars 1
bbox:car_number ((905, 400), (1056, 529)) : 1
The minimum distance from car: 3 is 4.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

66%| 831/1261 [37:04<19:11, 2.68s/it]

The times for each task are: [0.569935, 1.362437, 0.879931, 0.392842, 0.295848] with:

Minimum: 0.295848 Maximum: 1.362437 Average: 0.7002 seconds

Number of cars 1
bbox:car_number ((896, 400), (1052, 529)) : 1
The minimum distance from car: 3 is 6.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

66%| 832/1261 [37:07<19:08, 2.68s/it]

The times for each task are: [0.961446, 1.446803, 0.749551, 0.393619, 0.285965] with:

Minimum: 0.285965 Maximum: 1.446803 Average: 0.7675 seconds

```
Number of cars 1
bbox:car_number ((896, 419), (1025, 529)) : 1
The minimum distance from car: 3 is 17.204650534085253
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

66%| 833/1261 [37:10<19:05, 2.68s/it]

The times for each task are: [0.623091, 0.904203, 1.416647, 0.390977, 0.500553] with:

Minimum: 0.390977 Maximum: 1.416647 Average: 0.7671 seconds

```
Number of cars 1
bbox:car_number ((896, 415), (1047, 529)) : 1
The minimum distance from car: 3 is 11.180339887498949
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

66%| 834/1261 [37:13<19:03, 2.68s/it]

The times for each task are: [0.87421, 1.534328, 0.617527, 0.438517, 0.349914] with:

Minimum: 0.349914 Maximum: 1.534328 Average: 0.7629 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1025, 529)) : 1
The minimum distance from car: 3 is 13.601470508735444
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

66%| 835/1261 [37:15<19:00, 2.68s/it]

The times for each task are: [0.853564, 0.648307, 1.554964, 0.430181, 0.332669] with:

Minimum: 0.332669 Maximum: 1.554964 Average: 0.7639 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1052, 529)) : 1
The minimum distance from car: 3 is 14.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

66%| 836/1261 [37:19<18:58, 2.68s/it]

The times for each task are: [0.591694, 0.902279, 0.406009, 1.588131, 0.325793] with:
Minimum: 0.325793 Maximum: 1.588131 Average: 0.7628 seconds

```
Number of cars 1
bbox:car_number ((910, 400), (1048, 522)) : 1
The minimum distance from car: 3 is 5.830951894845301
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

66%| 837/1261 [37:21<18:55, 2.68s/it]

The times for each task are: [0.615199, 0.945616, 1.355645, 0.310716, 0.430189] with:
Minimum: 0.310716 Maximum: 1.355645 Average: 0.7315 seconds

```
Number of cars 1
bbox:car_number ((910, 419), (1047, 518)) : 1
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

66%| 838/1261 [37:24<18:52, 2.68s/it]

The times for each task are: [0.895397, 0.682741, 1.372969, 0.371127, 0.441466] with:
Minimum: 0.371127 Maximum: 1.372969 Average: 0.7527 seconds

Number of cars 1

bbox:car_number ((896, 423), (1047, 518)) : 1

The minimum distance from car: 3 is 7.280109889280518

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

67%| 839/1261 [37:26<18:50, 2.68s/it]

The times for each task are: [0.828311, 0.565649, 1.333914, 0.493093, 0.299491] with:

Minimum: 0.299491 Maximum: 1.333914 Average: 0.7041 seconds

Number of cars 1

bbox:car_number ((896, 419), (1025, 516)) : 1

The minimum distance from car: 3 is 11.40175425099138

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

67%| 840/1261 [37:29<18:47, 2.68s/it]

The times for each task are: [1.328027, 0.68613, 0.983508, 0.401683, 0.382342] with:

Minimum: 0.382342 Maximum: 1.328027 Average: 0.7563 seconds

Number of cars 1

bbox:car_number ((896, 400), (1025, 518)) : 1

The minimum distance from car: 3 is 8.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

67%| 841/1261 [37:31<18:44, 2.68s/it]

The times for each task are: [0.57008, 0.915009, 1.665044, 0.437005, 0.39714] with:

Minimum: 0.39714 Maximum: 1.665044 Average: 0.7969 seconds

Number of cars 1

bbox:car_number ((894, 400), (1025, 512)) : 1

```
The minimum distance from car: 3 is 3.1622776601683795
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

67%| 842/1261 [37:34<18:42, 2.68s/it]

The times for each task are: [0.893756, 0.654188, 0.417105, 1.621894, 0.35973] with:

Minimum: 0.35973 Maximum: 1.621894 Average: 0.7893 seconds

```
Number of cars 1
bbox:car_number ((895, 400), (1025, 512)) : 1
The minimum distance from car: 3 is 1.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

67%| 843/1261 [37:37<18:39, 2.68s/it]

The times for each task are: [0.633382, 0.90327, 1.386287, 0.402244, 0.30307] with:

Minimum: 0.30307 Maximum: 1.386287 Average: 0.7257 seconds

```
Number of cars 1
bbox:car_number ((896, 419), (1025, 512)) : 1
The minimum distance from car: 3 is 9.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

67%| 844/1261 [37:40<18:36, 2.68s/it]

The times for each task are: [0.627793, 0.913943, 1.371118, 0.315124, 0.467645] with:

Minimum: 0.315124 Maximum: 1.371118 Average: 0.7391 seconds

```
Number of cars 1
bbox:car_number ((895, 400), (1025, 512)) : 1
The minimum distance from car: 3 is 9.0
```

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 845/1261 [37:43<18:34, 2.68s/it]

The times for each task are: [0.640796, 1.084231, 1.499508, 0.404178, 0.354586] with:

Minimum: 0.354586 Maximum: 1.499508 Average: 0.7967 seconds

```
Number of cars 1
bbox:car_number ((894, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 3.1622776601683795
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 846/1261 [37:45<18:31, 2.68s/it]

The times for each task are: [1.003505, 0.746543, 1.40822, 0.336297, 0.491925] with:

Minimum: 0.336297 Maximum: 1.40822 Average: 0.7973 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1025, 498)) : 1
The minimum distance from car: 3 is 10.04987562112089
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 847/1261 [37:47<18:28, 2.68s/it]

The times for each task are: [0.833407, 1.366495, 0.669463, 0.401154, 0.298046] with:

Minimum: 0.298046 Maximum: 1.366495 Average: 0.7137 seconds

```
Number of cars 1
bbox:car_number ((896, 400), (1024, 512)) : 1
The minimum distance from car: 3 is 7.0
totalCars: 6
```

```
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 848/1261 [37:50<18:25, 2.68s/it]

The times for each task are: [0.588511, 0.872364, 1.465085, 0.308218, 0.394459] with:

Minimum: 0.308218 Maximum: 1.465085 Average: 0.7257 seconds

```
Number of cars 1
bbox:car_number ((886, 400), (1025, 512)) : 1
The minimum distance from car: 3 is 5.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 849/1261 [37:53<18:23, 2.68s/it]

The times for each task are: [0.563208, 0.877927, 1.636971, 0.389013, 0.346849] with:

Minimum: 0.346849 Maximum: 1.636971 Average: 0.7628 seconds

```
Number of cars 1
bbox:car_number ((875, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 5.830951894845301
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

67%| 850/1261 [37:55<18:20, 2.68s/it]

The times for each task are: [0.618863, 0.925835, 1.628186, 0.323594, 0.406074] with:

Minimum: 0.323594 Maximum: 1.628186 Average: 0.7805 seconds

```
Number of cars 1
bbox:car_number ((882, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
67%| 851/1261 [37:58<18:17, 2.68s/it]
```

```
The times for each task are: [0.894983, 0.683989, 1.55559, 0.485793, 0.387025] with:
```

```
Minimum: 0.387025 Maximum: 1.55559 Average: 0.8015 seconds
```

```
Number of cars 1
bbox:car_number ((882, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
68%| 852/1261 [38:01<18:15, 2.68s/it]
```

```
The times for each task are: [1.307888, 0.660383, 0.96949, 0.360072, 0.400213] with:
```

```
Minimum: 0.360072 Maximum: 1.307888 Average: 0.7396 seconds
```

```
Number of cars 1
bbox:car_number ((886, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
68%| 853/1261 [38:04<18:12, 2.68s/it]
```

```
The times for each task are: [0.607259, 0.97744, 1.36431, 0.407578, 0.367102] with:
```

```
Minimum: 0.367102 Maximum: 1.36431 Average: 0.7447 seconds
```

```
Number of cars 1
bbox:car_number ((886, 400), (1025, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
```

Number of processes used: 3

68%| 854/1261 [38:06<18:09, 2.68s/it]

The times for each task are: [0.850878, 0.654842, 1.46454, 0.456969, 0.32934] with:

Minimum: 0.32934 Maximum: 1.46454 Average: 0.7513 seconds

Number of cars 1

bbox:car_number ((880, 400), (1025, 518)) : 1

The minimum distance from car: 3 is 3.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 855/1261 [38:09<18:07, 2.68s/it]

The times for each task are: [0.592133, 1.377537, 1.061384, 0.378316, 0.284405] with:

Minimum: 0.284405 Maximum: 1.377537 Average: 0.7388 seconds

Number of cars 1

bbox:car_number ((886, 400), (1024, 518)) : 1

The minimum distance from car: 3 is 3.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 856/1261 [38:11<18:04, 2.68s/it]

The times for each task are: [0.637602, 1.012917, 0.310582, 1.443753, 0.467891] with:

Minimum: 0.310582 Maximum: 1.443753 Average: 0.7745 seconds

Number of cars 1

bbox:car_number ((875, 400), (1032, 522)) : 1

The minimum distance from car: 3 is 2.8284271247461903

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 857/1261 [38:14<18:01, 2.68s/it]

The times for each task are: [1.393673, 0.931038, 0.663127, 0.505135, 0.309477] with:

Minimum: 0.309477 Maximum: 1.393673 Average: 0.7605 seconds

Number of cars 1
bbox:car_number ((875, 400), (1047, 522)) : 1
The minimum distance from car: 3 is 8.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

68%| 858/1261 [38:17<17:58, 2.68s/it]

The times for each task are: [0.990449, 0.620341, 1.616351, 0.49944, 0.384963] with:

Minimum: 0.384963 Maximum: 1.616351 Average: 0.8223 seconds
Number of cars 1
bbox:car_number ((875, 400), (1047, 522)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

68%| 859/1261 [38:19<17:56, 2.68s/it]

The times for each task are: [1.053939, 0.629591, 1.591003, 0.325141, 0.444411] with:

Minimum: 0.325141 Maximum: 1.591003 Average: 0.8088 seconds
Number of cars 1
bbox:car_number ((875, 400), (1047, 522)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

68%| 860/1261 [38:22<17:53, 2.68s/it]

The times for each task are: [0.689901, 1.494554, 1.023447, 0.521344, 0.344888] with:

Minimum: 0.344888 Maximum: 1.494554 Average: 0.8148 seconds

Number of cars 1

bbox:car_number ((875, 400), (1048, 518)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 861/1261 [38:25<17:50, 2.68s/it]

The times for each task are: [0.592785, 0.918042, 1.526171, 0.479365, 0.289311] with:

Minimum: 0.289311 Maximum: 1.526171 Average: 0.7611 seconds

Number of cars 1

bbox:car_number ((865, 400), (1048, 512)) : 1

The minimum distance from car: 3 is 5.830951894845301

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 862/1261 [38:28<17:48, 2.68s/it]

The times for each task are: [0.60562, 1.621699, 1.057084, 0.36524, 0.440953] with:

Minimum: 0.36524 Maximum: 1.621699 Average: 0.8181 seconds

Number of cars 1

bbox:car_number ((859, 400), (1047, 518)) : 1

The minimum distance from car: 3 is 4.242640687119285

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

68%| 863/1261 [38:30<17:45, 2.68s/it]

The times for each task are: [0.997056, 1.359221, 0.633786, 0.293628, 0.435018] with:

Minimum: 0.293628 Maximum: 1.359221 Average: 0.7437 seconds

Number of cars 1

bbox:car_number ((859, 400), (1048, 518)) : 1

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

69%| 864/1261 [38:33<17:43, 2.68s/it]

The times for each task are: [1.010236, 0.645085, 1.556595, 0.297027, 0.521973] with:

Minimum: 0.297027 Maximum: 1.556595 Average: 0.8062 seconds

Number of cars 1

bbox:car_number ((859, 400), (1048, 518)) : 1

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

69%| 865/1261 [38:36<17:40, 2.68s/it]

The times for each task are: [0.971325, 0.616129, 1.541233, 0.404058, 0.291281] with:

Minimum: 0.291281 Maximum: 1.541233 Average: 0.7648 seconds

Number of cars 1

bbox:car_number ((859, 400), (1052, 518)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

69%| 866/1261 [38:38<17:37, 2.68s/it]

The times for each task are: [0.603021, 0.895332, 1.404896, 0.451321, 0.328041] with:

Minimum: 0.328041 Maximum: 1.404896 Average: 0.7365 seconds

Number of cars 1
bbox:car_number ((865, 400), (1052, 518)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

69%| 867/1261 [38:41<17:34, 2.68s/it]

The times for each task are: [0.943239, 0.65855, 0.415697, 1.420617, 0.305955] with:

Minimum: 0.305955 Maximum: 1.420617 Average: 0.7488 seconds

Number of cars 1
bbox:car_number ((865, 400), (1052, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

69%| 868/1261 [38:44<17:32, 2.68s/it]

The times for each task are: [0.86734, 0.655504, 0.4207, 1.741096, 0.415063] with:

Minimum: 0.415063 Maximum: 1.741096 Average: 0.8199 seconds

Number of cars 1
bbox:car_number ((865, 400), (1056, 518)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

69%| 869/1261 [38:46<17:29, 2.68s/it]

The times for each task are: [0.576543, 0.926051, 1.564356, 0.415124, 0.346102] with:

Minimum: 0.346102 Maximum: 1.564356 Average: 0.7656 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 518)) : 1
The minimum distance from car: 3 is 4.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

69%| 870/1261 [38:49<17:26, 2.68s/it]

The times for each task are: [0.930184, 0.688727, 1.428054, 0.419386, 0.385393] with:

Minimum: 0.385393 Maximum: 1.428054 Average: 0.7703 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1056, 518)) : 1
The minimum distance from car: 3 is 7.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

69%| 871/1261 [38:52<17:24, 2.68s/it]

The times for each task are: [1.345423, 0.546144, 0.956899, 0.394678, 0.36211] with:

Minimum: 0.36211 Maximum: 1.345423 Average: 0.7211 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1056, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

69%| 872/1261 [38:54<17:21, 2.68s/it]

The times for each task are: [0.718724, 0.98999, 1.405322, 0.39454, 0.487461] with:

Minimum: 0.39454 Maximum: 1.405322 Average: 0.7992 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1056, 518)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

69%| 873/1261 [38:57<17:18, 2.68s/it]

The times for each task are: [0.914285, 1.411212, 0.611779, 0.369821, 0.424208] with:
Minimum: 0.369821 Maximum: 1.411212 Average: 0.7463 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1067, 518)) : 1
The minimum distance from car: 3 is 6.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

69%| 874/1261 [38:59<17:16, 2.68s/it]

The times for each task are: [0.620775, 0.981022, 1.644067, 0.388228, 0.298962] with:
Minimum: 0.298962 Maximum: 1.644067 Average: 0.7866 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1067, 516)) : 1
The minimum distance from car: 3 is 2.23606797749979
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

69%| 875/1261 [39:02<17:13, 2.68s/it]

The times for each task are: [0.878637, 1.422266, 0.68868, 0.292101, 0.399242] with:
Minimum: 0.292101 Maximum: 1.422266 Average: 0.7362 seconds

Number of cars 1

```
bbox:car_number ((856, 400), (1067, 516)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

69%| 876/1261 [39:05<17:10, 2.68s/it]

The times for each task are: [0.932466, 0.622329, 1.698385, 0.398454, 0.28794] with:

Minimum: 0.28794 Maximum: 1.698385 Average: 0.7879 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

70%| 877/1261 [39:07<17:08, 2.68s/it]

The times for each task are: [0.617251, 0.909104, 1.594556, 0.353736, 0.388403] with:

Minimum: 0.353736 Maximum: 1.594556 Average: 0.7726 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

70%| 878/1261 [39:10<17:05, 2.68s/it]

The times for each task are: [0.855908, 1.425694, 0.727302, 0.323251, 0.425407] with:

Minimum: 0.323251 Maximum: 1.425694 Average: 0.7515 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 516)) : 1
```

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

70%| 879/1261 [39:13<17:02, 2.68s/it]

The times for each task are: [0.851821, 1.399021, 0.636771, 0.530846, 0.277624] with:

Minimum: 0.277624 Maximum: 1.399021 Average: 0.7392 seconds

Number of cars 1

bbox:car_number ((856, 400), (1067, 516)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

70%| 880/1261 [39:16<17:00, 2.68s/it]

The times for each task are: [0.869851, 0.705115, 1.545141, 0.399932, 0.352627] with:

Minimum: 0.352627 Maximum: 1.545141 Average: 0.7745 seconds

Number of cars 1

bbox:car_number ((859, 400), (1070, 512)) : 1

The minimum distance from car: 3 is 3.605551275463989

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

70%| 881/1261 [39:19<16:57, 2.68s/it]

The times for each task are: [0.663111, 0.879362, 1.387855, 0.392748, 0.283266] with:

Minimum: 0.283266 Maximum: 1.387855 Average: 0.7213 seconds

Number of cars 1

bbox:car_number ((856, 400), (1070, 512)) : 1

The minimum distance from car: 3 is 1.0

```
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 882/1261 [39:21<16:54, 2.68s/it]

The times for each task are: [0.809319, 0.685485, 1.351932, 0.305663, 0.496232] with:

Minimum: 0.305663 Maximum: 1.351932 Average: 0.7297 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 512)) : 1
The minimum distance from car: 3 is 1.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 883/1261 [39:24<16:52, 2.68s/it]

The times for each task are: [0.870191, 1.600209, 0.619787, 0.400303, 0.348908] with:

Minimum: 0.348908 Maximum: 1.600209 Average: 0.7679 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 518)) : 1
The minimum distance from car: 3 is 3.1622776601683795
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 884/1261 [39:27<16:49, 2.68s/it]

The times for each task are: [0.887147, 1.385959, 0.653176, 0.389773, 0.383912] with:

Minimum: 0.383912 Maximum: 1.385959 Average: 0.74 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 512)) : 1
The minimum distance from car: 3 is 3.1622776601683795
totalCars: 6
```

```
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 885/1261 [39:29<16:46, 2.68s/it]

The times for each task are: [0.648617, 0.931232, 1.422956, 0.375948, 0.30319] with:

Minimum: 0.30319 Maximum: 1.422956 Average: 0.7364 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 886/1261 [39:32<16:44, 2.68s/it]

The times for each task are: [0.680968, 1.400094, 0.904123, 0.485162, 0.353257] with:

Minimum: 0.353257 Maximum: 1.400094 Average: 0.7647 seconds

```
Number of cars 1
bbox:car_number ((856, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 1.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3
```

70%| 887/1261 [39:34<16:41, 2.68s/it]

The times for each task are: [0.871974, 1.389482, 0.741954, 0.484299, 0.337456] with:

Minimum: 0.337456 Maximum: 1.389482 Average: 0.765 seconds

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 518)) : 1
The minimum distance from car: 3 is 1.4142135623730951
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
70%| 888/1261 [39:37<16:38, 2.68s/it]
```

```
The times for each task are: [0.666683, 0.901061, 1.387803, 0.325503, 0.463971] with:
```

```
Minimum: 0.325503 Maximum: 1.387803 Average: 0.749 seconds
```

```
Number of cars 1
bbox:car_number ((856, 400), (1077, 518)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
70%| 889/1261 [39:40<16:36, 2.68s/it]
```

```
The times for each task are: [0.655241, 0.862824, 1.626807, 0.399628, 0.297778] with:
```

```
Minimum: 0.297778 Maximum: 1.626807 Average: 0.7685 seconds
```

```
Number of cars 1
bbox:car_number ((859, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 2.23606797749979
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3
```

```
71%| 890/1261 [39:43<16:33, 2.68s/it]
```

```
The times for each task are: [0.83597, 0.693789, 1.380824, 0.387217, 0.376086] with:
```

```
Minimum: 0.376086 Maximum: 1.380824 Average: 0.7348 seconds
```

```
Number of cars 1
bbox:car_number ((865, 400), (1070, 516)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
```

Number of processes used: 3

71%| 891/1261 [39:45<16:30, 2.68s/it]

The times for each task are: [0.571928, 1.395883, 1.014877, 0.366365, 0.439511] with:

Minimum: 0.366365 Maximum: 1.395883 Average: 0.7577 seconds

Number of cars 1

bbox:car_number ((859, 400), (1079, 516)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 892/1261 [39:48<16:28, 2.68s/it]

The times for each task are: [0.89943, 1.363977, 0.712443, 0.445128, 0.345785] with:

Minimum: 0.345785 Maximum: 1.363977 Average: 0.7534 seconds

Number of cars 1

bbox:car_number ((859, 400), (1077, 516)) : 1

The minimum distance from car: 3 is 1.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 893/1261 [39:51<16:25, 2.68s/it]

The times for each task are: [1.338058, 0.6775, 0.493054, 0.963098, 0.382999] with:

Minimum: 0.382999 Maximum: 1.338058 Average: 0.7709 seconds

Number of cars 1

bbox:car_number ((865, 400), (1079, 516)) : 1

The minimum distance from car: 3 is 4.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 894/1261 [39:54<16:22, 2.68s/it]

The times for each task are: [0.670663, 1.003106, 1.410458, 0.456012, 0.355664] with:

Minimum: 0.355664 Maximum: 1.410458 Average: 0.7792 seconds

Number of cars 1

bbox:car_number ((859, 400), (1086, 516)) : 1

The minimum distance from car: 3 is 0.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 895/1261 [39:56<16:19, 2.68s/it]

The times for each task are: [0.598691, 0.958556, 1.513513, 0.394681, 0.329043] with:

Minimum: 0.329043 Maximum: 1.513513 Average: 0.7589 seconds

Number of cars 1

bbox:car_number ((865, 400), (1086, 516)) : 1

The minimum distance from car: 3 is 3.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 896/1261 [39:59<16:17, 2.68s/it]

The times for each task are: [0.990283, 1.38178, 0.672566, 0.452866, 0.319274] with:

Minimum: 0.319274 Maximum: 1.38178 Average: 0.7634 seconds

Number of cars 1

bbox:car_number ((859, 400), (1087, 516)) : 1

The minimum distance from car: 3 is 2.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 897/1261 [40:02<16:14, 2.68s/it]

The times for each task are: [1.3749, 0.98998, 0.652508, 0.495538, 0.41454] with:

Minimum: 0.41454 Maximum: 1.3749 Average: 0.7855 seconds

Number of cars 1

bbox:car_number ((850, 400), (1087, 516)) : 1

The minimum distance from car: 3 is 5.0

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 898/1261 [40:04<16:12, 2.68s/it]

The times for each task are: [0.873415, 0.622086, 1.590346, 0.291671, 0.368472] with:

Minimum: 0.291671 Maximum: 1.590346 Average: 0.7492 seconds

Number of cars 1

bbox:car_number ((865, 400), (1086, 512)) : 1

The minimum distance from car: 3 is 7.280109889280518

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 899/1261 [40:07<16:09, 2.68s/it]

The times for each task are: [0.889843, 0.62019, 1.562513, 0.4248, 0.285391] with:

Minimum: 0.285391 Maximum: 1.562513 Average: 0.7565 seconds

Number of cars 1

bbox:car_number ((850, 400), (1086, 516)) : 1

The minimum distance from car: 3 is 7.280109889280518

totalCars: 6

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

71%| 900/1261 [40:09<16:06, 2.68s/it]

The times for each task are: [0.88875, 0.636773, 1.35753, 0.307869, 0.406292] with:

Minimum: 0.307869 Maximum: 1.35753 Average: 0.7194 seconds

Number of cars 1
bbox:car_number ((850, 400), (1087, 516)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3

71% | 901/1261 [40:12<16:03, 2.68s/it]

The times for each task are: [0.647525, 0.93294, 1.537113, 0.366081, 0.424955] with:

Minimum: 0.366081 Maximum: 1.537113 Average: 0.7817 seconds

Number of cars 1
bbox:car_number ((859, 400), (1087, 516)) : 1
The minimum distance from car: 3 is 5.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3

72% | 902/1261 [40:15<16:01, 2.68s/it]

The times for each task are: [1.398965, 0.857339, 0.675823, 0.293855, 0.399503] with:

Minimum: 0.293855 Maximum: 1.398965 Average: 0.7251 seconds

Number of cars 1
bbox:car_number ((834, 400), (1087, 512)) : 1
The minimum distance from car: 3 is 13.152946437965905
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3

72% | 903/1261 [40:17<15:58, 2.68s/it]

The times for each task are: [0.855237, 0.612896, 1.385946, 0.299963, 0.40697] with:

Minimum: 0.299963 Maximum: 1.385946 Average: 0.7122 seconds

Number of cars 1
bbox:car_number ((834, 400), (1093, 512)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

72%| 904/1261 [40:20<15:55, 2.68s/it]

The times for each task are: [0.903195, 0.57531, 1.405611, 0.39609, 0.298908] with:

Minimum: 0.298908 Maximum: 1.405611 Average: 0.7158 seconds

Number of cars 1
bbox:car_number ((835, 400), (1087, 512)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

72%| 905/1261 [40:22<15:53, 2.68s/it]

The times for each task are: [0.632357, 0.859618, 1.455548, 0.389281, 0.380135] with:

Minimum: 0.380135 Maximum: 1.455548 Average: 0.7434 seconds

Number of cars 1
bbox:car_number ((834, 400), (1093, 512)) : 1
The minimum distance from car: 3 is 2.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1141, 415))]
Length of task list: 5
Number of processes used: 3

72%| 906/1261 [40:25<15:50, 2.68s/it]

The times for each task are: [0.6946, 1.32614, 0.884331, 0.292421, 0.435576] with:

Minimum: 0.292421 Maximum: 1.32614 Average: 0.7266 seconds

```
Number of cars 1
bbox:car_number ((834, 400), (1093, 512)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

72%| 907/1261 [40:28<15:47, 2.68s/it]

The times for each task are: [0.568635, 0.890228, 1.673287, 0.443271, 0.297531] with:

Minimum: 0.297531 Maximum: 1.673287 Average: 0.7746 seconds

```
Number of cars 1
bbox:car_number ((834, 400), (1093, 512)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 6
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

72%| 908/1261 [40:31<15:45, 2.68s/it]

The times for each task are: [0.913621, 0.621201, 1.47148, 0.343933, 0.43044] with:

Minimum: 0.343933 Maximum: 1.47148 Average: 0.7561 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (963, 512)) : 1
bbox:car_number ((994, 400), (1093, 495)) : 2
The minimum distance from car: 1 is 8.94427190999916
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

72%| 909/1261 [40:33<15:42, 2.68s/it]

The times for each task are: [1.345103, 0.707412, 0.854511, 0.29924, 0.436223] with:

Minimum: 0.29924 Maximum: 1.345103 Average: 0.7285 seconds

Number of cars 1
bbox:car_number ((834, 400), (1106, 512)) : 1
The minimum distance from car: 3 is 7.0
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

72%| 910/1261 [40:35<15:39, 2.68s/it]

The times for each task are: [0.563164, 1.388731, 0.901025, 0.295502, 0.37606] with:

Minimum: 0.295502 Maximum: 1.388731 Average: 0.7049 seconds

Number of cars 1
bbox:car_number ((834, 400), (1107, 512)) : 1
The minimum distance from car: 3 is 0.0
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

72%| 911/1261 [40:38<15:36, 2.68s/it]

The times for each task are: [0.57894, 0.867914, 1.610021, 0.305361, 0.487084] with:

Minimum: 0.305361 Maximum: 1.610021 Average: 0.7699 seconds

Number of cars 1
bbox:car_number ((835, 400), (1106, 507)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44
Length of task list: 5
Number of processes used: 3

72%| 912/1261 [40:41<15:34, 2.68s/it]

The times for each task are: [0.624917, 0.926934, 1.628142, 0.395394, 0.284665] with:

Minimum: 0.284665 Maximum: 1.628142 Average: 0.772 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (963, 512)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((970, 400), (1106, 498)) : 2
The minimum distance from car: 1 is 5.385164807134504
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

72%| 913/1261 [40:44<15:31, 2.68s/it]

The times for each task are: [0.853359, 0.681071, 1.618434, 0.439619, 0.291707] with:
Minimum: 0.291707 Maximum: 1.618434 Average: 0.7768 seconds

```
Number of cars 1
bbox:car_number ((834, 400), (1107, 512)) : 1
The minimum distance from car: 3 is 3.0
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

72%| 914/1261 [40:46<15:28, 2.68s/it]

The times for each task are: [0.616403, 0.904028, 0.373255, 1.571014, 0.301219] with:
Minimum: 0.301219 Maximum: 1.571014 Average: 0.7532 seconds

```
Number of cars 1
bbox:car_number ((834, 400), (1105, 516)) : 1
The minimum distance from car: 3 is 2.23606797749979
totalCars: 7
Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1241, 518))]
Length of task list: 5
Number of processes used: 3
```

73%| 915/1261 [40:49<15:26, 2.68s/it]

The times for each task are: [1.066994, 0.663143, 1.354706, 0.387084, 0.302024] with:

Minimum: 0.302024 Maximum: 1.354706 Average: 0.7548 seconds

Number of cars 2

bbox:car_number ((834, 400), (963, 512)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((989, 400), (1116, 498)) : 2

The minimum distance from car: 1 is 14.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

73%| 916/1261 [40:52<15:23, 2.68s/it]

The times for each task are: [0.927674, 0.626071, 1.617321, 0.295029, 0.383383] with:

Minimum: 0.295029 Maximum: 1.617321 Average: 0.7699 seconds

Number of cars 1

bbox:car_number ((834, 400), (1116, 512)) : 1

The minimum distance from car: 3 is 6.324555320336759

totalCars: 7

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1242, 518))]

Length of task list: 5

Number of processes used: 3

73%| 917/1261 [40:55<15:20, 2.68s/it]

The times for each task are: [0.623518, 0.878667, 1.647567, 0.309053, 0.387676] with:

Minimum: 0.309053 Maximum: 1.647567 Average: 0.7693 seconds

Number of cars 1

bbox:car_number ((834, 400), (1107, 512)) : 1

The minimum distance from car: 3 is 5.0

totalCars: 7

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 446), (1242, 518))]

Length of task list: 5

Number of processes used: 3

73%| | 918/1261 [40:57<15:18, 2.68s/it]

The times for each task are: [0.880373, 1.351499, 0.681516, 0.408302, 0.276546] with:

Minimum: 0.276546 Maximum: 1.351499 Average: 0.7196 seconds

Number of cars 1

bbox:car_number ((834, 400), (1116, 512)) : 1

The minimum distance from car: 3 is 5.0

totalCars: 7

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

73%| | 919/1261 [41:00<15:15, 2.68s/it]

The times for each task are: [0.620716, 0.961508, 0.417364, 1.566369, 0.288062] with:

Minimum: 0.288062 Maximum: 1.566369 Average: 0.7708 seconds

Number of cars 1

bbox:car_number ((834, 400), (1118, 512)) : 1

The minimum distance from car: 3 is 1.0

totalCars: 7

Car Number: 3 Car Positions: [((1141, 415), (1241, 518)), ((1136, 446), (1242, 518)), ((1090, 44

Length of task list: 5

Number of processes used: 3

73%| | 920/1261 [41:03<15:12, 2.68s/it]

The times for each task are: [0.608738, 0.916063, 1.409315, 0.394686, 0.29627] with:

Minimum: 0.29627 Maximum: 1.409315 Average: 0.725 seconds

Number of cars 2

bbox:car_number ((834, 400), (963, 512)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((970, 400), (1116, 497)) : 2

The minimum distance from car: 1 is 9.055385138137417

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

73%| | 921/1261 [41:05<15:10, 2.68s/it]

The times for each task are: [0.833189, 0.582343, 1.380914, 0.401064, 0.293181] with:

Minimum: 0.293181 Maximum: 1.380914 Average: 0.6981 seconds

Number of cars 2

bbox:car_number ((834, 400), (963, 512)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((967, 400), (1118, 497)) : 2

The minimum distance from car: 1 is 1.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

73%| | 922/1261 [41:08<15:07, 2.68s/it]

The times for each task are: [0.860595, 0.659625, 1.498712, 0.433051, 0.302868] with:

Minimum: 0.302868 Maximum: 1.498712 Average: 0.751 seconds

Number of cars 2

bbox:car_number ((834, 400), (954, 512)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((970, 400), (1116, 497)) : 2

The minimum distance from car: 1 is 1.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

73%| | 923/1261 [41:10<15:04, 2.68s/it]

The times for each task are: [0.887242, 1.348001, 0.785997, 0.402545, 0.374036] with:

Minimum: 0.374036 Maximum: 1.348001 Average: 0.7596 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (963, 512)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((970, 400), (1118, 498)) : 2
The minimum distance from car: 1 is 1.4142135623730951
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

73%| | 924/1261 [41:13<15:02, 2.68s/it]

The times for each task are: [0.821412, 1.336533, 0.700316, 0.393921, 0.474942] with:

Minimum: 0.393921 Maximum: 1.336533 Average: 0.7454 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (944, 512)) : 1
The minimum distance from car: 1 is 9.0
bbox:car_number ((970, 400), (1118, 495)) : 2
The minimum distance from car: 1 is 2.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

73%| | 925/1261 [41:16<14:59, 2.68s/it]

The times for each task are: [0.954475, 1.373968, 0.68228, 0.390332, 0.346491] with:

Minimum: 0.346491 Maximum: 1.373968 Average: 0.7495 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (953, 512)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((970, 400), (1124, 498)) : 2
The minimum distance from car: 1 is 3.605551275463989
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
```

```
Length of task list: 5
Number of processes used: 3
```

```
73%| | 926/1261 [41:18<14:56, 2.68s/it]
```

```
The times for each task are: [1.057126, 0.621881, 1.405676, 0.338993, 0.443088] with:
```

```
Minimum: 0.338993 Maximum: 1.405676 Average: 0.7734 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (944, 512)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((970, 400), (1122, 507)) : 2
The minimum distance from car: 1 is 4.123105625617661
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
74%| | 927/1261 [41:21<14:54, 2.68s/it]
```

```
The times for each task are: [0.594721, 1.013605, 1.642421, 0.456817, 0.34519] with:
```

```
Minimum: 0.34519 Maximum: 1.642421 Average: 0.8106 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (944, 512)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((970, 400), (1124, 498)) : 2
The minimum distance from car: 1 is 4.123105625617661
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
74%| | 928/1261 [41:24<14:51, 2.68s/it]
```

```
The times for each task are: [0.902831, 0.703295, 1.632131, 0.400132, 0.328942] with:
```

Minimum: 0.328942 Maximum: 1.632131 Average: 0.7935 seconds

Number of cars 2
bbox:car_number ((834, 400), (944, 507)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((970, 400), (1133, 512)) : 2
The minimum distance from car: 1 is 8.06225774829855
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

74%| 929/1261 [41:27<14:48, 2.68s/it]

The times for each task are: [0.65851, 0.95564, 1.432976, 0.403904, 0.344262] with:

Minimum: 0.344262 Maximum: 1.432976 Average: 0.7591 seconds

Number of cars 2
bbox:car_number ((834, 400), (944, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((970, 400), (1133, 512)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

74%| 930/1261 [41:30<14:46, 2.68s/it]

The times for each task are: [1.083635, 0.632792, 1.670355, 0.392323, 0.501861] with:

Minimum: 0.392323 Maximum: 1.670355 Average: 0.8562 seconds

Number of cars 2
bbox:car_number ((975, 400), (1137, 507)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((834, 415), (932, 507)) : 2
The minimum distance from car: 1 is 10.0
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

```
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

74%| | 931/1261 [41:32<14:43, 2.68s/it]

The times for each task are: [0.583548, 1.048925, 1.542206, 0.488876, 0.358343] with:

Minimum: 0.358343 Maximum: 1.542206 Average: 0.8044 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (953, 512)) : 1
The minimum distance from car: 1 is 11.180339887498949
bbox:car_number ((975, 400), (1139, 507)) : 2
The minimum distance from car: 1 is 1.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| | 932/1261 [41:35<14:40, 2.68s/it]

The times for each task are: [1.010325, 1.445791, 0.576911, 0.494635, 0.395568] with:

Minimum: 0.395568 Maximum: 1.445791 Average: 0.7846 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (944, 512)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((975, 400), (1139, 507)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| | 933/1261 [41:38<14:38, 2.68s/it]

The times for each task are: [0.886716, 0.6188, 1.438158, 0.423303, 0.297642] with:

Minimum: 0.297642 Maximum: 1.438158 Average: 0.7329 seconds

Number of cars 2

bbox:car_number ((834, 400), (944, 512)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1000, 400), (1139, 507)) : 2

The minimum distance from car: 1 is 12.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (934, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1139, 507))]

Length of task list: 5

Number of processes used: 3

74%| | 934/1261 [41:40<14:35, 2.68s/it]

The times for each task are: [0.693674, 0.909205, 1.373709, 0.477608, 0.289099] with:

Minimum: 0.289099 Maximum: 1.373709 Average: 0.7487 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 512)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((989, 400), (1139, 507)) : 2

The minimum distance from car: 1 is 5.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (934, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1139, 507))]

Length of task list: 5

Number of processes used: 3

74%| | 935/1261 [41:43<14:32, 2.68s/it]

The times for each task are: [0.924621, 1.611832, 0.609294, 0.391768, 0.310832] with:

Minimum: 0.310832 Maximum: 1.611832 Average: 0.7697 seconds

Number of cars 2

bbox:car_number ((834, 400), (942, 512)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((989, 400), (1139, 512)) : 2

The minimum distance from car: 1 is 3.0

```
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| 936/1261 [41:46<14:30, 2.68s/it]

The times for each task are: [0.904184, 1.449177, 0.653112, 0.387508, 0.345657] with:
Minimum: 0.345657 Maximum: 1.449177 Average: 0.7479 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 507)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((989, 400), (1139, 512)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| 937/1261 [41:49<14:27, 2.68s/it]

The times for each task are: [0.829623, 0.605081, 1.35763, 0.311215, 0.409366] with:
Minimum: 0.311215 Maximum: 1.35763 Average: 0.7026 seconds

```
Number of cars 4
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((989, 400), (1139, 507)) : 2
The minimum distance from car: 1 is 3.0
bbox:car_number ((940, 430), (942, 462)) : 3
bbox:car_number ((985, 445), (987, 462)) : 4
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

74%| | 938/1261 [41:51<14:24, 2.68s/it]

The times for each task are: [0.854278, 0.620621, 1.373699, 0.475782, 0.293921] with:

Minimum: 0.293921 Maximum: 1.373699 Average: 0.7237 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((989, 400), (1143, 512)) : 2

The minimum distance from car: 1 is 3.605551275463989

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

74%| | 939/1261 [41:54<14:22, 2.68s/it]

The times for each task are: [0.81749, 1.374192, 0.697054, 0.284961, 0.485888] with:

Minimum: 0.284961 Maximum: 1.374192 Average: 0.7319 seconds

Number of cars 2

bbox:car_number ((834, 400), (942, 512)) : 1

The minimum distance from car: 1 is 8.06225774829855

bbox:car_number ((989, 400), (1149, 518)) : 2

The minimum distance from car: 1 is 4.242640687119285

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

75%| | 940/1261 [41:56<14:19, 2.68s/it]

The times for each task are: [1.330917, 0.957099, 0.694031, 0.433775, 0.334961] with:

Minimum: 0.334961 Maximum: 1.330917 Average: 0.7502 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((989, 400), (1149, 512)) : 2
The minimum distance from car: 1 is 3.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

75%| 941/1261 [41:59<14:16, 2.68s/it]

The times for each task are: [0.911916, 1.591809, 0.654085, 0.41193, 0.347808] with:

Minimum: 0.347808 Maximum: 1.591809 Average: 0.7835 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((989, 400), (1149, 512)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

75%| 942/1261 [42:02<14:14, 2.68s/it]

The times for each task are: [0.845409, 1.494318, 0.754962, 0.438452, 0.316715] with:

Minimum: 0.316715 Maximum: 1.494318 Average: 0.77 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((998, 400), (1160, 498)) : 2
The minimum distance from car: 1 is 12.206555615733702
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
```

```
Length of task list: 5
Number of processes used: 3
```

```
75%| | 943/1261 [42:05<14:11, 2.68s/it]
```

```
The times for each task are: [0.638393, 0.906425, 1.455119, 0.409263, 0.347119] with:
```

```
Minimum: 0.347119 Maximum: 1.455119 Average: 0.7513 seconds
```

```
Number of cars 2
bbox:car_number ((832, 400), (934, 512)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((994, 400), (1160, 512)) : 2
The minimum distance from car: 1 is 7.280109889280518
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
75%| | 944/1261 [42:07<14:08, 2.68s/it]
```

```
The times for each task are: [0.709108, 1.028494, 1.360739, 0.332001, 0.441915] with:
```

```
Minimum: 0.332001 Maximum: 1.360739 Average: 0.7745 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (934, 498)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((989, 400), (1162, 512)) : 2
The minimum distance from car: 1 is 2.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

```
75%| | 945/1261 [42:10<14:06, 2.68s/it]
```

```
The times for each task are: [0.864916, 0.687201, 0.414902, 1.679184, 0.34114] with:
```

Minimum: 0.34114 Maximum: 1.679184 Average: 0.7975 seconds

Number of cars 3
bbox:car_number ((834, 400), (934, 498)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((994, 400), (1160, 512)) : 2
The minimum distance from car: 1 is 2.0
bbox:car_number ((940, 430), (942, 477)) : 3
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

75%| 946/1261 [42:13<14:03, 2.68s/it]

The times for each task are: [0.939638, 0.667084, 1.423423, 0.347637, 0.431605] with:

Minimum: 0.347637 Maximum: 1.423423 Average: 0.7619 seconds

Number of cars 2
bbox:car_number ((832, 400), (942, 512)) : 1
The minimum distance from car: 1 is 7.615773105863909
bbox:car_number ((998, 400), (1162, 512)) : 2
The minimum distance from car: 1 is 3.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3

75%| 947/1261 [42:16<14:00, 2.68s/it]

The times for each task are: [0.804338, 1.528484, 0.724767, 0.408457, 0.321057] with:

Minimum: 0.321057 Maximum: 1.528484 Average: 0.7574 seconds

Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((989, 400), (1162, 516)) : 2
The minimum distance from car: 1 is 5.385164807134504
totalCars: 7

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))
Length of task list: 5
Number of processes used: 3
```

75%| 948/1261 [42:18<13:58, 2.68s/it]

The times for each task are: [0.927387, 1.579477, 0.662515, 0.381479, 0.353227] with:

Minimum: 0.353227 Maximum: 1.579477 Average: 0.7808 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (942, 512)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((998, 400), (1162, 512)) : 2
The minimum distance from car: 1 is 5.385164807134504
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))
Length of task list: 5
Number of processes used: 3
```

75%| 949/1261 [42:21<13:55, 2.68s/it]

The times for each task are: [0.898455, 1.397565, 0.624496, 0.410488, 0.350284] with:

Minimum: 0.350284 Maximum: 1.397565 Average: 0.7363 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 4.242640687119285
bbox:car_number ((994, 400), (1162, 498)) : 2
The minimum distance from car: 1 is 7.280109889280518
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))
Length of task list: 5
Number of processes used: 3
```

75%| 950/1261 [42:24<13:53, 2.68s/it]

The times for each task are: [0.656565, 0.984787, 1.438557, 0.409169, 0.315061] with:

Minimum: 0.315061 Maximum: 1.438557 Average: 0.7608 seconds

Number of cars 2

bbox:car_number ((834, 400), (932, 507)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((998, 400), (1160, 512)) : 2

The minimum distance from car: 1 is 7.0710678118654755

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

75%| | 951/1261 [42:27<13:50, 2.68s/it]

The times for each task are: [1.017915, 0.657438, 1.42696, 0.459609, 0.29014] with:

Minimum: 0.29014 Maximum: 1.42696 Average: 0.7704 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 4.123105625617661

bbox:car_number ((998, 400), (1162, 512)) : 2

The minimum distance from car: 1 is 1.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

75%| | 952/1261 [42:30<13:47, 2.68s/it]

The times for each task are: [0.602853, 0.929848, 1.468224, 0.409752, 0.38231] with:

Minimum: 0.38231 Maximum: 1.468224 Average: 0.7586 seconds

Number of cars 2

bbox:car_number ((832, 400), (942, 498)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((998, 400), (1162, 512)) : 2

The minimum distance from car: 1 is 0.0

```
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

76%| | 953/1261 [42:33<13:45, 2.68s/it]

The times for each task are: [0.639273, 0.900397, 1.423843, 0.328515, 0.54924] with:

Minimum: 0.328515 Maximum: 1.423843 Average: 0.7683 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1000, 400), (1162, 512)) : 2
The minimum distance from car: 1 is 1.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

76%| | 954/1261 [42:35<13:42, 2.68s/it]

The times for each task are: [0.573946, 1.113353, 1.35762, 0.383624, 0.427536] with:

Minimum: 0.383624 Maximum: 1.35762 Average: 0.7712 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (931, 498)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((994, 400), (1167, 512)) : 2
The minimum distance from car: 1 is 1.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 40
Length of task list: 5
Number of processes used: 3
```

76%| | 955/1261 [42:38<13:39, 2.68s/it]

The times for each task are: [1.351171, 0.770655, 0.982072, 0.415174, 0.399722] with:

Minimum: 0.399722 Maximum: 1.351171 Average: 0.7838 seconds

Number of cars 2

bbox:car_number ((834, 400), (932, 498)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((998, 400), (1180, 512)) : 2

The minimum distance from car: 1 is 9.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

76%| | 956/1261 [42:40<13:37, 2.68s/it]

The times for each task are: [0.886319, 0.605088, 1.418634, 0.297563, 0.525476] with:

Minimum: 0.297563 Maximum: 1.418634 Average: 0.7466 seconds

Number of cars 2

bbox:car_number ((832, 400), (931, 498)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((1000, 400), (1180, 512)) : 2

The minimum distance from car: 1 is 1.0

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]

Length of task list: 5

Number of processes used: 3

76%| | 957/1261 [42:43<13:34, 2.68s/it]

The times for each task are: [0.866872, 0.621756, 0.491395, 1.56433, 0.305846] with:

Minimum: 0.305846 Maximum: 1.56433 Average: 0.77 seconds

Number of cars 2

bbox:car_number ((998, 400), (1180, 512)) : 1

The minimum distance from car: 1 is 1.0

```
bbox:car_number ((834, 415), (934, 498)) : 2
The minimum distance from car: 1 is 7.615773105863909
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

76%| | 958/1261 [42:46<13:31, 2.68s/it]

The times for each task are: [0.588771, 0.899605, 1.579091, 0.456466, 0.298711] with:
Minimum: 0.298711 Maximum: 1.579091 Average: 0.7645 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1030, 400), (1180, 512)) : 2
The minimum distance from car: 3 is 8.54400374531753
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

76%| | 959/1261 [42:49<13:29, 2.68s/it]

The times for each task are: [0.984261, 1.371405, 0.609455, 0.288201, 0.38911] with:
Minimum: 0.288201 Maximum: 1.371405 Average: 0.7285 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 498)) : 1
The minimum distance from car: 1 is 5.656854249492381
bbox:car_number ((998, 400), (1181, 512)) : 2
The minimum distance from car: 1 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 1 Car Positions: [((1204, 400), (1273, 478)), ((1175, 400), (1273, 512)), ((1183, 400), (1273, 512))]
Length of task list: 5
Number of processes used: 3
```

76%| | 960/1261 [42:51<13:26, 2.68s/it]

The times for each task are: [1.293971, 0.905812, 0.686347, 0.400574, 0.28733] with:

Minimum: 0.28733 Maximum: 1.293971 Average: 0.7148 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((1015, 400), (1181, 498)) : 2

The minimum distance from car: 3 is 9.899494936611665

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

76%| | 961/1261 [42:54<13:23, 2.68s/it]

The times for each task are: [0.598001, 1.011237, 1.490256, 0.458965, 0.395495] with:

Minimum: 0.395495 Maximum: 1.490256 Average: 0.7908 seconds

Number of cars 2

bbox:car_number ((832, 400), (932, 512)) : 1

The minimum distance from car: 1 is 7.280109889280518

bbox:car_number ((1015, 400), (1185, 518)) : 2

The minimum distance from car: 3 is 10.198039027185569

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

76%| | 962/1261 [42:57<13:21, 2.68s/it]

The times for each task are: [1.35537, 0.968188, 0.722312, 0.384638, 0.329624] with:

Minimum: 0.329624 Maximum: 1.35537 Average: 0.752 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 507)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((1021, 400), (1185, 512)) : 2
The minimum distance from car: 3 is 4.242640687119285
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

76%| | 963/1261 [43:00<13:18, 2.68s/it]

The times for each task are: [0.836494, 1.314533, 0.701595, 0.380779, 0.31146] with:

Minimum: 0.31146 Maximum: 1.314533 Average: 0.709 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1021, 400), (1185, 512)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

76%| | 964/1261 [43:03<13:15, 2.68s/it]

The times for each task are: [0.581463, 0.976225, 1.477826, 0.41082, 0.334713] with:

Minimum: 0.334713 Maximum: 1.477826 Average: 0.7562 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1021, 400), (1185, 516)) : 2
The minimum distance from car: 3 is 2.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
```

```
Length of task list: 5
Number of processes used: 3
```

```
77%| | 965/1261 [43:06<13:13, 2.68s/it]
```

```
The times for each task are: [0.647779, 1.501464, 0.88435, 0.296268, 0.406107] with:
```

```
Minimum: 0.296268 Maximum: 1.501464 Average: 0.7472 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1021, 400), (1185, 498)) : 2
The minimum distance from car: 3 is 9.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

```
77%| | 966/1261 [43:09<13:10, 2.68s/it]
```

```
The times for each task are: [0.97938, 1.348145, 0.627577, 0.299514, 0.428596] with:
```

```
Minimum: 0.299514 Maximum: 1.348145 Average: 0.7366 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1021, 400), (1185, 507)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

```
77%| | 967/1261 [43:11<13:07, 2.68s/it]
```

```
The times for each task are: [0.6052, 0.901306, 1.604109, 0.331284, 0.424153] with:
```

Minimum: 0.331284 Maximum: 1.604109 Average: 0.7732 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1021, 400), (1187, 498)) : 2
The minimum distance from car: 3 is 4.123105625617661
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

77%| 968/1261 [43:14<13:05, 2.68s/it]

The times for each task are: [0.598834, 0.886808, 1.429519, 0.533443, 0.300734] with:

Minimum: 0.300734 Maximum: 1.429519 Average: 0.7499 seconds

```
Number of cars 2
bbox:car_number ((1021, 400), (1187, 498)) : 1
The minimum distance from car: 3 is 0.0
bbox:car_number ((834, 415), (934, 507)) : 2
The minimum distance from car: 1 is 8.94427190999916
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1181, 564))]
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

77%| 969/1261 [43:17<13:02, 2.68s/it]

The times for each task are: [0.73471, 0.917915, 1.456573, 0.320735, 0.446087] with:

Minimum: 0.320735 Maximum: 1.456573 Average: 0.7752 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (932, 498)) : 1
The minimum distance from car: 1 is 12.041594578792296
bbox:car_number ((1027, 400), (1200, 512)) : 2
The minimum distance from car: 3 is 11.40175425099138
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

77%| 970/1261 [43:19<12:59, 2.68s/it]

```
The times for each task are: [0.848724, 1.345905, 0.679573, 0.373841, 0.282113] with:
Minimum: 0.282113 Maximum: 1.345905 Average: 0.706 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1027, 400), (1208, 498)) : 2
The minimum distance from car: 3 is 8.06225774829855
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

77%| 971/1261 [43:22<12:57, 2.68s/it]

```
The times for each task are: [0.821468, 1.337674, 0.636487, 0.416459, 0.280782] with:
Minimum: 0.280782 Maximum: 1.337674 Average: 0.6986 seconds
```

```
Number of cars 2
bbox:car_number ((1027, 400), (1208, 498)) : 1
The minimum distance from car: 3 is 0.0
bbox:car_number ((835, 415), (932, 495)) : 2
The minimum distance from car: 1 is 6.0
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

77%| 972/1261 [43:25<12:54, 2.68s/it]

The times for each task are: [0.910108, 1.482487, 0.616503, 0.419474, 0.286206] with:

Minimum: 0.286206 Maximum: 1.482487 Average: 0.743 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 6.082762530298219

bbox:car_number ((1027, 400), (1211, 507)) : 2

The minimum distance from car: 3 is 4.47213595499958

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

77%| | 973/1261 [43:27<12:51, 2.68s/it]

The times for each task are: [0.633312, 1.007691, 1.445894, 0.370044, 0.419235] with:

Minimum: 0.370044 Maximum: 1.445894 Average: 0.7752 seconds

Number of cars 3

bbox:car_number ((834, 400), (934, 507)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((1021, 400), (1208, 512)) : 2

The minimum distance from car: 3 is 5.830951894845301

bbox:car_number ((1210, 430), (1211, 477)) : 3

totalCars: 7

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 7

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

77%| | 974/1261 [43:30<12:49, 2.68s/it]

The times for each task are: [1.351346, 0.930801, 0.62604, 0.306575, 0.385155] with:

Minimum: 0.306575 Maximum: 1.351346 Average: 0.72 seconds

Number of cars 2

bbox:car_number ((835, 400), (932, 497)) : 1

The minimum distance from car: 1 is 5.0990195135927845

bbox:car_number ((1027, 400), (1208, 498)) : 2

```
The minimum distance from car: 3 is 7.615773105863909
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

77%| | 975/1261 [43:32<12:46, 2.68s/it]

```
The times for each task are: [0.898287, 0.699116, 1.630689, 0.407396, 0.361021] with:
Minimum: 0.361021 Maximum: 1.630689 Average: 0.7993 seconds
```

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((1021, 400), (1211, 512)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

77%| | 976/1261 [43:35<12:43, 2.68s/it]

```
The times for each task are: [1.349588, 0.607186, 0.995388, 0.415541, 0.333954] with:
Minimum: 0.333954 Maximum: 1.349588 Average: 0.7403 seconds
```

```
Number of cars 3
bbox:car_number ((834, 400), (934, 507)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1027, 400), (1208, 512)) : 2
The minimum distance from car: 3 is 1.0
bbox:car_number ((1210, 415), (1214, 462)) : 3
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

77%| | 977/1261 [43:38<12:41, 2.68s/it]

The times for each task are: [0.631274, 0.8996, 1.453344, 0.324202, 0.406528] with:

Minimum: 0.324202 Maximum: 1.453344 Average: 0.743 seconds

Number of cars 2
bbox:car_number ((1044, 400), (1214, 512)) : 1
The minimum distance from car: 3 is 12.0
bbox:car_number ((834, 415), (932, 507)) : 2
The minimum distance from car: 1 is 8.06225774829855
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (833, 400))
Length of task list: 5
Number of processes used: 3

78%| | 978/1261 [43:40<12:38, 2.68s/it]

The times for each task are: [0.685289, 1.597517, 0.937855, 0.47172, 0.410929] with:

Minimum: 0.410929 Maximum: 1.597517 Average: 0.8207 seconds

Number of cars 2
bbox:car_number ((835, 400), (932, 498)) : 1
The minimum distance from car: 1 is 12.0
bbox:car_number ((1045, 400), (1212, 512)) : 2
The minimum distance from car: 3 is 1.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (833, 400))
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3

78%| | 979/1261 [43:43<12:35, 2.68s/it]

The times for each task are: [0.886933, 0.611597, 1.635924, 0.328802, 0.449076] with:

Minimum: 0.328802 Maximum: 1.635924 Average: 0.7825 seconds

```
Number of cars 2
bbox:car_number ((837, 400), (934, 497)) : 1
The minimum distance from car: 1 is 2.23606797749979
bbox:car_number ((1045, 400), (1214, 512)) : 2
The minimum distance from car: 3 is 1.0
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]
Length of task list: 5
Number of processes used: 3
```

78%| 980/1261 [43:46<12:33, 2.68s/it]

The times for each task are: [0.925208, 0.60041, 1.418191, 0.409482, 0.471607] with:

Minimum: 0.409482 Maximum: 1.418191 Average: 0.765 seconds

```
Number of cars 2
bbox:car_number ((837, 400), (934, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1048, 400), (1214, 518)) : 2
The minimum distance from car: 3 is 3.605551275463989
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]
Length of task list: 5
Number of processes used: 3
```

78%| 981/1261 [43:49<12:30, 2.68s/it]

The times for each task are: [0.611267, 0.868731, 1.360557, 0.391191, 0.343247] with:

Minimum: 0.343247 Maximum: 1.360557 Average: 0.715 seconds

```
Number of cars 3
bbox:car_number ((837, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1046, 400), (1214, 512)) : 2
The minimum distance from car: 3 is 3.1622776601683795
bbox:car_number ((571, 475), (612, 507)) : 3
totalCars: 7
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 7
```

```
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1060, 520), (537, 552)) : 2
Length of task list: 5
Number of processes used: 3
```

78%| 982/1261 [43:51<12:27, 2.68s/it]

The times for each task are: [0.630643, 0.831844, 1.388376, 0.295478, 0.386685] with:

Minimum: 0.295478 Maximum: 1.388376 Average: 0.7066 seconds

```
Number of cars 5
bbox:car_number ((837, 400), (934, 507)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((1060, 400), (1230, 512)) : 2
The minimum distance from car: 3 is 15.0
bbox:car_number ((550, 475), (611, 537)) : 3
bbox:car_number ((520, 520), (537, 552)) : 4
bbox:car_number ((505, 550), (507, 567)) : 5
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1060, 520), (537, 552)) : 2
totalCars: 8
Car Number: 3 Car Positions: [((550, 475), (611, 537))]
Length of task list: 5
Number of processes used: 3
```

78%| 983/1261 [43:54<12:25, 2.68s/it]

The times for each task are: [0.551351, 1.412225, 1.010155, 0.520696, 0.310969] with:

Minimum: 0.310969 Maximum: 1.412225 Average: 0.7611 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 507)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1048, 400), (1230, 512)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1060, 520), (537, 552)) : 2
Length of task list: 5
Number of processes used: 3
```

78%| 984/1261 [43:57<12:22, 2.68s/it]

The times for each task are: [0.610566, 0.884663, 1.415672, 0.411815, 0.405348] with:

Minimum: 0.405348 Maximum: 1.415672 Average: 0.7456 seconds

Number of cars 3
bbox:car_number ((836, 400), (934, 497)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1048, 400), (1241, 512)) : 2
The minimum distance from car: 3 is 5.0
bbox:car_number ((533, 469), (612, 552)) : 3
The minimum distance from car: 3 is 8.94427190999916
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 490), (1178, 552))]
totalCars: 8
Car Number: 3 Car Positions: [((550, 475), (611, 537)), ((533, 469), (612, 552))]
Length of task list: 5
Number of processes used: 3

78%| 985/1261 [44:00<12:19, 2.68s/it]

The times for each task are: [0.716765, 0.901041, 1.576074, 0.366474, 0.469462] with:

Minimum: 0.366474 Maximum: 1.576074 Average: 0.806 seconds

Number of cars 3
bbox:car_number ((837, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1046, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 7.0710678118654755
bbox:car_number ((535, 490), (592, 554)) : 3
The minimum distance from car: 3 is 15.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 490), (1178, 552))]
totalCars: 8
Car Number: 3 Car Positions: [((550, 475), (611, 537)), ((533, 469), (612, 552)), ((535, 490), (611, 552))]
Length of task list: 5
Number of processes used: 3

78%| | 986/1261 [44:02<12:17, 2.68s/it]

The times for each task are: [0.871273, 0.68977, 1.478256, 0.389729, 0.286228] with:

Minimum: 0.286228 Maximum: 1.478256 Average: 0.7431 seconds

Number of cars 4

bbox:car_number ((837, 400), (932, 497)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1048, 400), (1230, 498)) : 2

The minimum distance from car: 3 is 4.0

bbox:car_number ((514, 492), (586, 564)) : 3

The minimum distance from car: 3 is 14.317821063276353

bbox:car_number ((493, 550), (507, 582)) : 4

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 490), (1144, 505))]

totalCars: 8

Car Number: 3 Car Positions: [((550, 475), (611, 537)), ((533, 469), (612, 552)), ((535, 490), (550, 475))]

Length of task list: 5

Number of processes used: 3

78%| | 987/1261 [44:05<12:14, 2.68s/it]

The times for each task are: [0.585842, 0.937567, 1.663095, 0.380924, 0.434703] with:

Minimum: 0.380924 Maximum: 1.663095 Average: 0.8004 seconds

Number of cars 2

bbox:car_number ((837, 400), (934, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1048, 400), (1241, 518)) : 2

The minimum distance from car: 3 is 11.180339887498949

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 490), (1144, 505))]

Length of task list: 5

Number of processes used: 3

78%| | 988/1261 [44:07<12:11, 2.68s/it]

The times for each task are: [0.963019, 1.439172, 0.6322, 0.350055, 0.464801] with:

Minimum: 0.350055 Maximum: 1.439172 Average: 0.7698 seconds

Number of cars 2

bbox:car_number ((837, 400), (931, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1051, 400), (1238, 512)) : 2

The minimum distance from car: 3 is 3.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

78%| | 989/1261 [44:10<12:08, 2.68s/it]

The times for each task are: [0.613012, 0.869326, 1.370963, 0.390746, 0.357404] with:

Minimum: 0.357404 Maximum: 1.370963 Average: 0.7203 seconds

Number of cars 2

bbox:car_number ((856, 400), (934, 498)) : 1

The minimum distance from car: 1 is 11.045361017187261

bbox:car_number ((1046, 400), (1241, 518)) : 2

The minimum distance from car: 3 is 3.1622776601683795

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

79%| | 990/1261 [44:12<12:06, 2.68s/it]

The times for each task are: [0.627551, 0.884707, 1.410228, 0.415247, 0.354957] with:

Minimum: 0.354957 Maximum: 1.410228 Average: 0.7385 seconds

Number of cars 2

bbox:car_number ((1048, 400), (1241, 518)) : 1

The minimum distance from car: 3 is 1.0

bbox:car_number ((836, 415), (934, 498)) : 2

The minimum distance from car: 1 is 12.206555615733702

```
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

79%| 991/1261 [44:15<12:03, 2.68s/it]

The times for each task are: [0.851964, 0.58031, 1.43178, 0.530237, 0.297484] with:

Minimum: 0.297484 Maximum: 1.43178 Average: 0.7384 seconds

```
Number of cars 2
bbox:car_number ((859, 400), (934, 497)) : 1
The minimum distance from car: 1 is 13.601470508735444
bbox:car_number ((1048, 400), (1241, 518)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

79%| 992/1261 [44:18<12:00, 2.68s/it]

The times for each task are: [0.619219, 0.889379, 0.406288, 0.392498, 1.3539] with:

Minimum: 0.392498 Maximum: 1.3539 Average: 0.7323 seconds

```
Number of cars 2
bbox:car_number ((1048, 400), (1242, 522)) : 1
The minimum distance from car: 3 is 2.23606797749979
bbox:car_number ((856, 415), (942, 498)) : 2
The minimum distance from car: 1 is 8.54400374531753
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

79%| | 993/1261 [44:21<11:58, 2.68s/it]

The times for each task are: [0.645935, 1.036753, 1.435087, 0.4513, 0.369414] with:

Minimum: 0.369414 Maximum: 1.435087 Average: 0.7877 seconds

Number of cars 2

bbox:car_number ((856, 400), (934, 497)) : 1

The minimum distance from car: 1 is 8.94427190999916

bbox:car_number ((1051, 400), (1242, 518)) : 2

The minimum distance from car: 3 is 2.23606797749979

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

79%| | 994/1261 [44:24<11:55, 2.68s/it]

The times for each task are: [0.647863, 1.066183, 1.399561, 0.382808, 0.334226] with:

Minimum: 0.334226 Maximum: 1.399561 Average: 0.7661 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 11.045361017187261

bbox:car_number ((1065, 400), (1257, 518)) : 2

The minimum distance from car: 3 is 15.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

79%| | 995/1261 [44:26<11:52, 2.68s/it]

The times for each task are: [0.607732, 0.899621, 1.630486, 0.302294, 0.382051] with:

Minimum: 0.302294 Maximum: 1.630486 Average: 0.7644 seconds

Number of cars 2

bbox:car_number ((1051, 400), (1253, 518)) : 1

The minimum distance from car: 3 is 9.0

```
bbox:car_number ((859, 415), (932, 478)) : 2
The minimum distance from car: 1 is 11.40175425099138
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

79%| 996/1261 [44:29<11:50, 2.68s/it]

The times for each task are: [0.619056, 0.93751, 1.394554, 0.299026, 0.436091] with:

Minimum: 0.299026 Maximum: 1.394554 Average: 0.7372 seconds

```
Number of cars 2
bbox:car_number ((859, 400), (932, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1051, 400), (1253, 522)) : 2
The minimum distance from car: 3 is 2.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

79%| 997/1261 [44:31<11:47, 2.68s/it]

The times for each task are: [0.582871, 1.416793, 0.886211, 0.39044, 0.30199] with:

Minimum: 0.30199 Maximum: 1.416793 Average: 0.7157 seconds

```
Number of cars 2
bbox:car_number ((1051, 400), (1257, 516)) : 1
The minimum distance from car: 3 is 3.605551275463989
bbox:car_number ((859, 415), (927, 485)) : 2
The minimum distance from car: 1 is 3.605551275463989
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

79%| | 998/1261 [44:34<11:44, 2.68s/it]

The times for each task are: [0.887003, 0.601894, 1.52261, 0.474115, 0.290461] with:

Minimum: 0.290461 Maximum: 1.52261 Average: 0.7552 seconds

Number of cars 2

bbox:car_number ((1065, 400), (1253, 512)) : 1

The minimum distance from car: 3 is 5.385164807134504

bbox:car_number ((835, 415), (932, 498)) : 2

The minimum distance from car: 1 is 11.661903789690601

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (

Length of task list: 5

Number of processes used: 3

79%| | 999/1261 [44:36<11:42, 2.68s/it]

The times for each task are: [0.865745, 1.316627, 0.729992, 0.434131, 0.338183] with:

Minimum: 0.338183 Maximum: 1.316627 Average: 0.7369 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 7.0710678118654755

bbox:car_number ((1065, 400), (1242, 512)) : 2

The minimum distance from car: 3 is 6.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

Length of task list: 5

Number of processes used: 3

79%| | 1000/1261 [44:39<11:39, 2.68s/it]

The times for each task are: [0.900198, 1.441552, 0.652548, 0.411821, 0.287164] with:

Minimum: 0.287164 Maximum: 1.441552 Average: 0.7387 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1065, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

79%| 1001/1261 [44:42<11:36, 2.68s/it]

The times for each task are: [1.399181, 0.62245, 1.039388, 0.435062, 0.304674] with:

Minimum: 0.304674 Maximum: 1.399181 Average: 0.7602 seconds

```
Number of cars 2
bbox:car_number ((1129, 423), (1139, 459)) : 1
bbox:car_number ((1165, 430), (1212, 477)) : 2
Length of task list: 5
Number of processes used: 3
```

79%| 1002/1261 [44:44<11:33, 2.68s/it]

The times for each task are: [0.695644, 0.897406, 1.397234, 0.370673, 0.466814] with:

Minimum: 0.370673 Maximum: 1.397234 Average: 0.7656 seconds

```
Number of cars 2
bbox:car_number ((837, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1090, 400), (1214, 485)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

80%| 1003/1261 [44:47<11:31, 2.68s/it]

The times for each task are: [0.636618, 0.908462, 1.417418, 0.288833, 0.37995] with:

Minimum: 0.288833 Maximum: 1.417418 Average: 0.7263 seconds

Number of cars 2

bbox:car_number ((859, 400), (932, 485)) : 1

The minimum distance from car: 1 is 12.529964086141668

bbox:car_number ((1084, 400), (1238, 485)) : 2

The minimum distance from car: 3 is 9.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

80%| 1004/1261 [44:49<11:28, 2.68s/it]

The times for each task are: [0.831712, 0.678808, 1.322123, 0.316383, 0.406009] with:

Minimum: 0.316383 Maximum: 1.322123 Average: 0.711 seconds

Number of cars 2

bbox:car_number ((835, 400), (908, 485)) : 1

The minimum distance from car: 1 is 24.0

bbox:car_number ((1084, 400), (1238, 485)) : 2

The minimum distance from car: 3 is 0.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

80%| 1005/1261 [44:52<11:25, 2.68s/it]

The times for each task are: [0.619316, 0.854139, 1.365546, 0.444046, 0.30298] with:

Minimum: 0.30298 Maximum: 1.365546 Average: 0.7172 seconds

Number of cars 3

bbox:car_number ((859, 400), (932, 495)) : 1

The minimum distance from car: 1 is 24.515301344262525

bbox:car_number ((1156, 400), (1214, 472)) : 2

The minimum distance from car: 3 is 24.73863375370596

```
bbox:car_number ((1129, 427), (1139, 485)) : 3
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

80%| 1006/1261 [44:55<11:23, 2.68s/it]

The times for each task are: [0.961311, 1.359491, 0.718892, 0.390765, 0.397301] with:

Minimum: 0.390765 Maximum: 1.359491 Average: 0.7656 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (944, 498)) : 1
The minimum distance from car: 1 is 6.324555320336759
bbox:car_number ((1105, 400), (1219, 485)) : 2
The minimum distance from car: 3 is 23.769728648009426
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

80%| 1007/1261 [44:57<11:20, 2.68s/it]

The times for each task are: [0.843872, 0.611706, 0.4098, 1.639596, 0.305133] with:

Minimum: 0.305133 Maximum: 1.639596 Average: 0.762 seconds

```
Number of cars 2
bbox:car_number ((1075, 400), (1219, 498)) : 1
The minimum distance from car: 3 is 16.55294535724685
bbox:car_number ((859, 423), (932, 495)) : 2
The minimum distance from car: 1 is 11.661903789690601
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

80%| 1008/1261 [45:00<11:17, 2.68s/it]

The times for each task are: [0.839991, 1.514863, 0.660055, 0.397938, 0.346037] with:

Minimum: 0.346037 Maximum: 1.514863 Average: 0.7518 seconds

Number of cars 2

bbox:car_number ((1075, 400), (1227, 497)) : 1

The minimum distance from car: 3 is 4.123105625617661

bbox:car_number ((836, 415), (932, 498)) : 2

The minimum distance from car: 1 is 11.40175425099138

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

80%| 1009/1261 [45:03<11:15, 2.68s/it]

The times for each task are: [0.62781, 1.420733, 0.995983, 0.349851, 0.404722] with:

Minimum: 0.349851 Maximum: 1.420733 Average: 0.7598 seconds

Number of cars 2

bbox:car_number ((835, 400), (934, 498)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1090, 400), (1219, 485)) : 2

The minimum distance from car: 3 is 6.708203932499369

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49

Length of task list: 5

Number of processes used: 3

80%| 1010/1261 [45:06<11:12, 2.68s/it]

The times for each task are: [0.571011, 0.954011, 0.396307, 1.557369, 0.424203] with:

Minimum: 0.396307 Maximum: 1.557369 Average: 0.7806 seconds

Number of cars 2

bbox:car_number ((835, 400), (934, 498)) : 1

The minimum distance from car: 1 is 0.0

```
bbox:car_number ((1090, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 13.038404810405298
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

80%| 1011/1261 [45:08<11:09, 2.68s/it]

The times for each task are: [0.919905, 0.621117, 1.555491, 0.355045, 0.38182] with:

Minimum: 0.355045 Maximum: 1.555491 Average: 0.7667 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1082, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

80%| 1012/1261 [45:11<11:07, 2.68s/it]

The times for each task are: [1.34547, 0.652052, 0.978274, 0.377981, 0.293242] with:

Minimum: 0.293242 Maximum: 1.34547 Average: 0.7294 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((1090, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

80%| 1013/1261 [45:14<11:04, 2.68s/it]

The times for each task are: [0.615033, 0.992307, 1.358066, 0.360234, 0.397747] with:

Minimum: 0.360234 Maximum: 1.358066 Average: 0.7447 seconds

Number of cars 2

bbox:car_number ((835, 400), (942, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1102, 400), (1253, 498)) : 2

The minimum distance from car: 3 is 12.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

80%| 1014/1261 [45:17<11:01, 2.68s/it]

The times for each task are: [0.569931, 0.868961, 1.517381, 0.39048, 0.285121] with:

Minimum: 0.285121 Maximum: 1.517381 Average: 0.7264 seconds

Number of cars 2

bbox:car_number ((835, 400), (942, 497)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1090, 400), (1253, 498)) : 2

The minimum distance from car: 3 is 6.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

80%| 1015/1261 [45:19<10:59, 2.68s/it]

The times for each task are: [0.880946, 0.611508, 1.413502, 0.343658, 0.401577] with:

Minimum: 0.343658 Maximum: 1.413502 Average: 0.7302 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1103, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

81%| 1016/1261 [45:22<10:56, 2.68s/it]

The times for each task are: [0.834748, 1.377019, 0.705714, 0.435116, 0.280621] with:

Minimum: 0.280621 Maximum: 1.377019 Average: 0.7266 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (942, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1102, 400), (1257, 512)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

81%| 1017/1261 [45:25<10:53, 2.68s/it]

The times for each task are: [1.3107, 0.980251, 0.582742, 0.413903, 0.297272] with:

Minimum: 0.297272 Maximum: 1.3107 Average: 0.717 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (942, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1075, 400), (1242, 512)) : 2
The minimum distance from car: 3 is 21.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
81%| 1018/1261 [45:27<10:51, 2.68s/it]
```

```
The times for each task are: [0.874084, 0.745771, 1.391891, 0.406886, 0.282699] with:
```

```
Minimum: 0.282699 Maximum: 1.391891 Average: 0.7403 seconds
```

```
Number of cars 4
bbox:car_number ((834, 400), (942, 498)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1113, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 25.96150997149434
bbox:car_number ((1075, 427), (1077, 459)) : 3
bbox:car_number ((1082, 427), (1086, 459)) : 4
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

```
81%| 1019/1261 [45:30<10:48, 2.68s/it]
```

```
The times for each task are: [0.608509, 1.412604, 0.882955, 0.39546, 0.427221] with:
```

```
Minimum: 0.39546 Maximum: 1.412604 Average: 0.7453 seconds
```

```
Number of cars 2
bbox:car_number ((835, 400), (942, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1046, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 40.0
totalCars: 8
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 8
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

```
81%| 1020/1261 [45:33<10:45, 2.68s/it]
```

The times for each task are: [1.013764, 1.400281, 0.693289, 0.30302, 0.490032] with:

Minimum: 0.30302 Maximum: 1.400281 Average: 0.7801 seconds

Number of cars 3

bbox:car_number ((834, 400), (944, 498)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((1090, 400), (1253, 498)) : 2

The minimum distance from car: 3 is 28.0

bbox:car_number ((1082, 427), (1086, 459)) : 3

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

81%| 1021/1261 [45:35<10:43, 2.68s/it]

The times for each task are: [1.37758, 0.645772, 1.055197, 0.300907, 0.387696] with:

Minimum: 0.300907 Maximum: 1.37758 Average: 0.7534 seconds

Number of cars 2

bbox:car_number ((835, 400), (942, 498)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1082, 400), (1257, 498)) : 2

The minimum distance from car: 3 is 2.0

totalCars: 8

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 8

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]

Length of task list: 5

Number of processes used: 3

81%| 1022/1261 [45:38<10:40, 2.68s/it]

The times for each task are: [0.637347, 0.916509, 0.319468, 1.393688, 0.445907] with:

Minimum: 0.319468 Maximum: 1.393688 Average: 0.7426 seconds

Number of cars 3

bbox:car_number ((835, 400), (942, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1102, 400), (1187, 485)) : 2

```
The minimum distance from car: 3 is 25.96150997149434
bbox:car_number ((1195, 400), (1268, 477)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1175, 400))]
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

81% | 1023/1261 [45:41<10:37, 2.68s/it]

The times for each task are: [0.60181, 1.039449, 1.472833, 0.296347, 0.404021] with:
Minimum: 0.296347 Maximum: 1.472833 Average: 0.7629 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (934, 498)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((1102, 400), (1268, 498)) : 2
The minimum distance from car: 3 is 41.593268686170845
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1175, 400))]
Length of task list: 5
Number of processes used: 3
```

81% | 1024/1261 [45:44<10:35, 2.68s/it]

The times for each task are: [0.867646, 1.394557, 0.617795, 0.393432, 0.288728] with:
Minimum: 0.288728 Maximum: 1.394557 Average: 0.7124 seconds

```
Number of cars 3
bbox:car_number ((835, 400), (942, 498)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1175, 400), (1268, 497)) : 2
The minimum distance from car: 3 is 14.142135623730951
bbox:car_number ((1122, 427), (1160, 459)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497))]
```

```
Length of task list: 5
Number of processes used: 3
```

```
81%| | 1025/1261 [45:47<10:32, 2.68s/it]
```

```
The times for each task are: [0.564792, 0.936364, 1.619091, 0.388006, 0.425783] with:
```

```
Minimum: 0.388006 Maximum: 1.619091 Average: 0.7868 seconds
```

```
Number of cars 2
bbox:car_number ((836, 400), (942, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1122, 400), (1273, 485)) : 2
The minimum distance from car: 3 is 13.892443989449804
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

```
81%| | 1026/1261 [45:49<10:29, 2.68s/it]
```

```
The times for each task are: [1.401313, 0.876846, 0.679835, 0.405463, 0.348737] with:
```

```
Minimum: 0.348737 Maximum: 1.401313 Average: 0.7424 seconds
```

```
Number of cars 2
bbox:car_number ((835, 400), (931, 492)) : 1
The minimum distance from car: 1 is 6.708203932499369
bbox:car_number ((1102, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 12.206555615733702
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

```
81%| | 1027/1261 [45:52<10:27, 2.68s/it]
```

```
The times for each task are: [0.875643, 1.574073, 0.470254, 0.69213, 0.338025] with:
```

Minimum: 0.338025 Maximum: 1.574073 Average: 0.79 seconds

Number of cars 2
bbox:car_number ((835, 400), (932, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1102, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3

82%| 1028/1261 [45:54<10:24, 2.68s/it]

The times for each task are: [0.582882, 0.893321, 1.542708, 0.40063, 0.300468] with:

Minimum: 0.300468 Maximum: 1.542708 Average: 0.744 seconds

Number of cars 2
bbox:car_number ((835, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1102, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3

82%| 1029/1261 [45:57<10:21, 2.68s/it]

The times for each task are: [0.706254, 1.074261, 1.612349, 0.339605, 0.508245] with:

Minimum: 0.339605 Maximum: 1.612349 Average: 0.8481 seconds

Number of cars 2
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1273, 512)) : 2
The minimum distance from car: 3 is 9.219544457292887
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

```
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

82%| | 1030/1261 [46:00<10:19, 2.68s/it]

```
The times for each task are: [0.588197, 0.983831, 1.637475, 0.389466, 0.304258] with:
Minimum: 0.304258 Maximum: 1.637475 Average: 0.7806 seconds
```

```
Number of cars 2
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1113, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

82%| | 1031/1261 [46:03<10:16, 2.68s/it]

```
The times for each task are: [0.983552, 0.679818, 1.399748, 0.394213, 0.459714] with:
Minimum: 0.394213 Maximum: 1.399748 Average: 0.7834 seconds
```

```
Number of cars 3
bbox:car_number ((834, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 0.0
bbox:car_number ((535, 475), (567, 507)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 498), (1144, 505))]
Length of task list: 5
Number of processes used: 3
```

82%| | 1032/1261 [46:06<10:13, 2.68s/it]

The times for each task are: [0.897478, 0.645902, 1.579841, 0.350766, 0.395283] with:

Minimum: 0.350766 Maximum: 1.579841 Average: 0.7739 seconds

Number of cars 3

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((1113, 400), (1187, 498)) : 2

The minimum distance from car: 3 is 43.0

bbox:car_number ((1195, 400), (1273, 495)) : 3

The minimum distance from car: 3 is 13.038404810405298

totalCars: 9

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 9

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1175, 400))]

totalCars: 9

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

82% | 1033/1261 [46:09<10:11, 2.68s/it]

The times for each task are: [0.836419, 1.578857, 0.611753, 0.40273, 0.337653] with:

Minimum: 0.337653 Maximum: 1.578857 Average: 0.7535 seconds

Number of cars 2

bbox:car_number ((834, 400), (932, 497)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((1135, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 30.066592756745816

totalCars: 9

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 9

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

82% | 1034/1261 [46:11<10:08, 2.68s/it]

The times for each task are: [0.600538, 0.959988, 1.521031, 0.48974, 0.395221] with:

Minimum: 0.395221 Maximum: 1.521031 Average: 0.7933 seconds

Number of cars 3

```
bbox:car_number ((832, 400), (934, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1135, 400), (1272, 498)) : 2
The minimum distance from car: 3 is 1.0
bbox:car_number ((514, 475), (567, 522)) : 3
The minimum distance from car: 3 is 31.622776601683793
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
totalCars: 9
Car Number: 3 Car Positions: [((550, 475), (611, 537)), ((533, 469), (612, 552)), ((535, 490), (611, 537))]
Length of task list: 5
Number of processes used: 3
```

82%| | 1035/1261 [46:14<10:05, 2.68s/it]

The times for each task are: [0.886098, 0.618083, 1.526901, 0.409589, 0.343757] with:

Minimum: 0.343757 Maximum: 1.526901 Average: 0.7569 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (934, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1268, 498)) : 2
The minimum distance from car: 3 is 13.0
bbox:car_number ((493, 505), (522, 507)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

82%| | 1036/1261 [46:17<10:03, 2.68s/it]

The times for each task are: [0.574579, 0.924025, 1.465887, 0.396696, 0.290859] with:

Minimum: 0.290859 Maximum: 1.465887 Average: 0.7304 seconds

```
Number of cars 3
bbox:car_number ((832, 400), (942, 498)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1113, 400), (1272, 498)) : 2
The minimum distance from car: 3 is 2.0
```

```
bbox:car_number ((505, 505), (522, 537)) : 3
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

82%| 1037/1261 [46:20<10:00, 2.68s/it]

The times for each task are: [0.85834, 0.673296, 1.658982, 0.466105, 0.37498] with:

Minimum: 0.37498 Maximum: 1.658982 Average: 0.8063 seconds

```
Number of cars 5
bbox:car_number ((834, 400), (942, 512)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((1136, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 12.0
bbox:car_number ((1027, 415), (1029, 447)) : 3
bbox:car_number ((490, 505), (492, 522)) : 4
bbox:car_number ((495, 505), (497, 522)) : 5
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

82%| 1038/1261 [46:22<09:57, 2.68s/it]

The times for each task are: [0.876665, 0.584709, 1.501177, 0.512517, 0.308547] with:

Minimum: 0.308547 Maximum: 1.501177 Average: 0.7567 seconds

```
Number of cars 4
bbox:car_number ((835, 400), (942, 498)) : 1
The minimum distance from car: 1 is 7.0
bbox:car_number ((1113, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 11.0
bbox:car_number ((1102, 427), (1106, 462)) : 3
bbox:car_number ((490, 520), (507, 522)) : 4
totalCars: 9
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 9
```

```
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

82%| | 1039/1261 [46:25<09:55, 2.68s/it]

The times for each task are: [0.885307, 0.634938, 1.440361, 0.294811, 0.390841] with:

Minimum: 0.294811 Maximum: 1.440361 Average: 0.7293 seconds

```
Number of cars 4  
bbox:car_number ((834, 400), (932, 498)) : 1  
The minimum distance from car: 1 is 5.0  
bbox:car_number ((1113, 400), (1276, 495)) : 2  
The minimum distance from car: 3 is 2.23606797749979  
bbox:car_number ((1030, 415), (1047, 459)) : 3  
bbox:car_number ((460, 514), (516, 560)) : 4  
totalCars: 10  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
totalCars: 10  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
totalCars: 10  
Car Number: 4 Car Positions: [((460, 514), (516, 560))]  
Length of task list: 5  
Number of processes used: 3
```

82%| | 1040/1261 [46:27<09:52, 2.68s/it]

The times for each task are: [1.028271, 0.614252, 1.512724, 0.480129, 0.298789] with:

Minimum: 0.298789 Maximum: 1.512724 Average: 0.7868 seconds

```
Number of cars 2  
bbox:car_number ((834, 400), (932, 498)) : 1  
The minimum distance from car: 1 is 0.0  
bbox:car_number ((1113, 400), (1273, 498)) : 2  
The minimum distance from car: 3 is 2.23606797749979  
totalCars: 10  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
totalCars: 10  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

83%| | 1041/1261 [46:30<09:49, 2.68s/it]

The times for each task are: [0.627361, 0.889984, 1.544662, 0.292187, 0.443007] with:

Minimum: 0.292187 Maximum: 1.544662 Average: 0.7594 seconds

Number of cars 4

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1129, 400), (1214, 485)) : 2

The minimum distance from car: 3 is 22.135943621178654

bbox:car_number ((1217, 415), (1272, 472)) : 3

bbox:car_number ((1075, 535), (1107, 582)) : 4

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1175, 535))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

83%| | 1042/1261 [46:33<09:47, 2.68s/it]

The times for each task are: [0.809534, 0.676534, 1.512337, 0.480458, 0.356415] with:

Minimum: 0.356415 Maximum: 1.512337 Average: 0.7671 seconds

Number of cars 2

bbox:car_number ((1113, 400), (1273, 498)) : 1

The minimum distance from car: 3 is 0.0

bbox:car_number ((834, 415), (934, 512)) : 2

The minimum distance from car: 1 is 14.0

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 497))]

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

83%| | 1043/1261 [46:36<09:44, 2.68s/it]

The times for each task are: [0.947975, 0.627738, 1.677169, 0.42847, 0.283201] with:

Minimum: 0.283201 Maximum: 1.677169 Average: 0.7929 seconds

```
Number of cars 4
bbox:car_number ((834, 400), (942, 497)) : 1
The minimum distance from car: 1 is 15.524174696260024
bbox:car_number ((1129, 400), (1200, 485)) : 2
The minimum distance from car: 3 is 7.0
bbox:car_number ((1204, 400), (1214, 472)) : 3
bbox:car_number ((1217, 400), (1276, 477)) : 4
The minimum distance from car: 3 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (934, 497))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477))]
Length of task list: 5
Number of processes used: 3
```

83% | 1044/1261 [46:38<09:41, 2.68s/it]

The times for each task are: [0.654004, 0.913331, 0.500898, 0.380789, 1.45329] with:

Minimum: 0.380789 Maximum: 1.45329 Average: 0.7805 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (934, 497)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1136, 400), (1214, 485)) : 2
The minimum distance from car: 3 is 11.0
bbox:car_number ((1217, 419), (1272, 472)) : 3
The minimum distance from car: 3 is 7.280109889280518
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (934, 497))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1144, 505))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 400), (1276, 477))]
Length of task list: 5
Number of processes used: 3
```

83% | 1045/1261 [46:41<09:39, 2.68s/it]

The times for each task are: [0.912297, 0.618505, 0.295126, 1.643798, 0.474952] with:

Minimum: 0.295126 Maximum: 1.643798 Average: 0.7889 seconds

```
Number of cars 4
bbox:car_number ((835, 400), (934, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1200, 485)) : 2
The minimum distance from car: 3 is 19.0
bbox:car_number ((1204, 400), (1214, 485)) : 3
bbox:car_number ((1217, 415), (1276, 477)) : 4
The minimum distance from car: 3 is 2.23606797749979
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1217, 415))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 400), (1276, 477))]
Length of task list: 5
Number of processes used: 3
```

83%| 1046/1261 [46:44<09:36, 2.68s/it]

The times for each task are: [0.806435, 0.596493, 1.559932, 0.287926, 0.39455] with:

Minimum: 0.287926 Maximum: 1.559932 Average: 0.7291 seconds

```
Number of cars 4
bbox:car_number ((834, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1129, 400), (1212, 485)) : 2
The minimum distance from car: 3 is 14.0
bbox:car_number ((1090, 427), (1092, 447)) : 3
bbox:car_number ((1225, 430), (1257, 447)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1217, 415))]
Length of task list: 5
Number of processes used: 3
```

83%| 1047/1261 [46:46<09:33, 2.68s/it]

The times for each task are: [0.974709, 0.617423, 1.361401, 0.300349, 0.440167] with:

Minimum: 0.300349 Maximum: 1.361401 Average: 0.7388 seconds

Number of cars 3

```
bbox:car_number ((834, 400), (934, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1136, 400), (1211, 485)) : 2
The minimum distance from car: 3 is 3.0
bbox:car_number ((1217, 419), (1276, 459)) : 3
The minimum distance from car: 3 is 7.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1217, 415))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 400), (1276, 477))]
Length of task list: 5
Number of processes used: 3
```

83%| | 1048/1261 [46:49<09:30, 2.68s/it]

The times for each task are: [0.613971, 1.342919, 0.851344, 0.55248, 0.287616] with:

Minimum: 0.287616 Maximum: 1.342919 Average: 0.7297 seconds

```
Number of cars 3
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1136, 400), (1211, 472)) : 2
The minimum distance from car: 3 is 6.0
bbox:car_number ((1225, 430), (1272, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1217, 415))]
Length of task list: 5
Number of processes used: 3
```

83%| | 1049/1261 [46:51<09:28, 2.68s/it]

The times for each task are: [0.836702, 1.326564, 0.739922, 0.440104, 0.297889] with:

Minimum: 0.297889 Maximum: 1.326564 Average: 0.7282 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (934, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1136, 400), (1211, 485)) : 2
The minimum distance from car: 3 is 6.0
```

```
bbox:car_number ((1225, 430), (1242, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

83%| | 1050/1261 [46:54<09:25, 2.68s/it]

The times for each task are: [1.326108, 0.861785, 0.687211, 0.374687, 0.293309] with:

Minimum: 0.293309 Maximum: 1.326108 Average: 0.7086 seconds

```
Number of cars 2
bbox:car_number ((835, 400), (932, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1136, 400), (1212, 485)) : 2
The minimum distance from car: 3 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

83%| | 1051/1261 [46:57<09:22, 2.68s/it]

The times for each task are: [0.643509, 1.468625, 0.959338, 0.382015, 0.293015] with:

Minimum: 0.293015 Maximum: 1.468625 Average: 0.7493 seconds

```
Number of cars 4
bbox:car_number ((835, 400), (934, 497)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1129, 400), (1211, 472)) : 2
The minimum distance from car: 3 is 7.211102550927978
bbox:car_number ((1113, 431), (1124, 459)) : 3
bbox:car_number ((1210, 460), (1211, 485)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

83%| | 1052/1261 [46:59<09:20, 2.68s/it]

The times for each task are: [0.613298, 0.843969, 1.67148, 0.393092, 0.328479] with:

Minimum: 0.328479 Maximum: 1.67148 Average: 0.7701 seconds

Number of cars 2

bbox:car_number ((834, 400), (932, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1113, 400), (1214, 485)) : 2

The minimum distance from car: 3 is 9.219544457292887

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

84%| | 1053/1261 [47:02<09:17, 2.68s/it]

The times for each task are: [0.898424, 0.679474, 0.392413, 1.554894, 0.379972] with:

Minimum: 0.379972 Maximum: 1.554894 Average: 0.781 seconds

Number of cars 2

bbox:car_number ((832, 400), (932, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1136, 400), (1214, 498)) : 2

The minimum distance from car: 3 is 13.892443989449804

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1178, 564))]

Length of task list: 5

Number of processes used: 3

84%| | 1054/1261 [47:05<09:14, 2.68s/it]

The times for each task are: [0.871902, 0.626413, 1.404039, 0.411826, 0.289573] with:

Minimum: 0.289573 Maximum: 1.404039 Average: 0.7208 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (934, 497)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1136, 400), (1214, 498)) : 2
The minimum distance from car: 3 is 0.0
bbox:car_number ((1113, 431), (1122, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

84%| | 1055/1261 [47:08<09:12, 2.68s/it]

The times for each task are: [0.552846, 1.001091, 1.511794, 0.385569, 0.331841] with:
Minimum: 0.331841 Maximum: 1.511794 Average: 0.7566 seconds

```
Number of cars 4
bbox:car_number ((834, 400), (932, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1129, 400), (1219, 498)) : 2
The minimum distance from car: 3 is 1.0
bbox:car_number ((1225, 400), (1242, 447)) : 3
bbox:car_number ((1113, 431), (1122, 447)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 495), (1181, 564))]
Length of task list: 5
Number of processes used: 3
```

84%| | 1056/1261 [47:10<09:09, 2.68s/it]

The times for each task are: [0.852583, 0.620008, 1.534752, 0.352623, 0.391345] with:
Minimum: 0.352623 Maximum: 1.534752 Average: 0.7503 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (932, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1129, 400), (1219, 498)) : 2
The minimum distance from car: 3 is 0.0
bbox:car_number ((1225, 400), (1257, 447)) : 3
```

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

84%| 1057/1261 [47:13<09:06, 2.68s/it]

The times for each task are: [0.884819, 0.609594, 1.373742, 0.3223, 0.437656] with:

Minimum: 0.3223 Maximum: 1.373742 Average: 0.7256 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1060, 400), (1212, 498)) : 2
The minimum distance from car: 3 is 38.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1144, 505), (1200, 573)), ((1120, 495), (1181, 564)), ((1113, 49
Length of task list: 5
Number of processes used: 3
```

84%| 1058/1261 [47:16<09:04, 2.68s/it]

The times for each task are: [0.851969, 0.687446, 1.498362, 0.390362, 0.369928] with:

Minimum: 0.369928 Maximum: 1.498362 Average: 0.7596 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1120, 400), (1214, 498)) : 2
The minimum distance from car: 3 is 26.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 40
Length of task list: 5
Number of processes used: 3
```

84%| | 1059/1261 [47:18<09:01, 2.68s/it]

The times for each task are: [0.640334, 0.857504, 1.369833, 0.30291, 0.401171] with:

Minimum: 0.30291 Maximum: 1.369833 Average: 0.7144 seconds

Number of cars 2

bbox:car_number ((834, 400), (934, 498)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1113, 400), (1212, 497)) : 2

The minimum distance from car: 3 is 5.0990195135927845

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

84%| | 1060/1261 [47:21<08:58, 2.68s/it]

The times for each task are: [1.024815, 0.702257, 1.382182, 0.467531, 0.330913] with:

Minimum: 0.330913 Maximum: 1.382182 Average: 0.7815 seconds

Number of cars 3

bbox:car_number ((832, 400), (932, 498)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((1129, 400), (1214, 498)) : 2

The minimum distance from car: 3 is 9.055385138137417

bbox:car_number ((1113, 415), (1124, 462)) : 3

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

84%| | 1061/1261 [47:24<08:56, 2.68s/it]

The times for each task are: [0.552752, 1.453788, 0.975316, 0.400102, 0.30321] with:

Minimum: 0.30321 Maximum: 1.453788 Average: 0.737 seconds

Number of cars 4

bbox:car_number ((834, 400), (932, 497)) : 1

```
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1150, 415), (1214, 478)) : 2
The minimum distance from car: 3 is 11.40175425099138
bbox:car_number ((1217, 419), (1219, 459)) : 3
bbox:car_number ((1129, 438), (1139, 472)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

84%| | 1062/1261 [47:27<08:53, 2.68s/it]

The times for each task are: [0.850393, 0.657411, 1.612153, 0.438986, 0.334161] with:

Minimum: 0.334161 Maximum: 1.612153 Average: 0.7786 seconds

```
Number of cars 3
bbox:car_number ((834, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1144, 400), (1219, 485)) : 2
The minimum distance from car: 3 is 4.123105625617661
bbox:car_number ((1129, 431), (1137, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

84%| | 1063/1261 [47:29<08:50, 2.68s/it]

The times for each task are: [0.97193, 1.393383, 0.655464, 0.453574, 0.306232] with:

Minimum: 0.306232 Maximum: 1.393383 Average: 0.7561 seconds

```
Number of cars 3
bbox:car_number ((820, 400), (932, 497)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((1129, 400), (1227, 498)) : 2
The minimum distance from car: 3 is 7.615773105863909
bbox:car_number ((1225, 460), (1230, 472)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

84%| | 1064/1261 [47:32<08:48, 2.68s/it]

The times for each task are: [0.698169, 0.937323, 1.55892, 0.326987, 0.483565] with:

Minimum: 0.326987 Maximum: 1.55892 Average: 0.801 seconds

```
Number of cars 3
bbox:car_number ((820, 400), (932, 498)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1113, 400), (1214, 498)) : 2
The minimum distance from car: 3 is 15.0
bbox:car_number ((1102, 427), (1105, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

84%| | 1065/1261 [47:34<08:45, 2.68s/it]

The times for each task are: [0.882506, 0.657351, 0.429504, 1.585046, 0.320166] with:

Minimum: 0.320166 Maximum: 1.585046 Average: 0.7749 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (927, 495)) : 1
The minimum distance from car: 1 is 3.605551275463989
bbox:car_number ((1141, 400), (1227, 485)) : 2
The minimum distance from car: 3 is 22.135943621178654
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

85%| | 1066/1261 [47:37<08:42, 2.68s/it]

The times for each task are: [0.696588, 0.87246, 1.378554, 0.301119, 0.437593] with:

Minimum: 0.301119 Maximum: 1.378554 Average: 0.7373 seconds

Number of cars 2

bbox:car_number ((813, 400), (931, 495)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1129, 400), (1219, 498)) : 2

The minimum distance from car: 3 is 12.206555615733702

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1067/1261 [47:39<08:39, 2.68s/it]

The times for each task are: [0.919863, 0.6506, 1.517024, 0.323772, 0.398977] with:

Minimum: 0.323772 Maximum: 1.517024 Average: 0.762 seconds

Number of cars 3

bbox:car_number ((832, 400), (927, 495)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1136, 400), (1230, 495)) : 2

The minimum distance from car: 3 is 9.219544457292887

bbox:car_number ((1225, 460), (1230, 472)) : 3

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1068/1261 [47:42<08:37, 2.68s/it]

The times for each task are: [0.631773, 0.911598, 1.390168, 0.428851, 0.300609] with:

Minimum: 0.300609 Maximum: 1.390168 Average: 0.7326 seconds

Number of cars 2

bbox:car_number ((820, 400), (931, 495)) : 1

The minimum distance from car: 1 is 4.0

bbox:car_number ((1129, 400), (1227, 498)) : 2

```
The minimum distance from car: 3 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

85%| | 1069/1261 [47:45<08:34, 2.68s/it]

The times for each task are: [0.866442, 0.64508, 1.44462, 0.281535, 0.380247] with:

Minimum: 0.281535 Maximum: 1.44462 Average: 0.7236 seconds

```
Number of cars 2
bbox:car_number ((820, 400), (927, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1129, 400), (1227, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

85%| | 1070/1261 [47:48<08:31, 2.68s/it]

The times for each task are: [0.619395, 0.903615, 1.36834, 0.367325, 0.348902] with:

Minimum: 0.348902 Maximum: 1.36834 Average: 0.7215 seconds

```
Number of cars 2
bbox:car_number ((820, 400), (931, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1129, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

85%| | 1071/1261 [47:50<08:29, 2.68s/it]

The times for each task are: [0.846017, 0.584574, 0.405247, 1.482657, 0.4194] with:

Minimum: 0.405247 Maximum: 1.482657 Average: 0.7476 seconds

Number of cars 2

bbox:car_number ((820, 400), (927, 495)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((1156, 400), (1230, 485)) : 2

The minimum distance from car: 3 is 10.63014581273465

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1072/1261 [47:53<08:26, 2.68s/it]

The times for each task are: [0.862131, 1.375914, 0.700743, 0.319571, 0.425616] with:

Minimum: 0.319571 Maximum: 1.375914 Average: 0.7368 seconds

Number of cars 3

bbox:car_number ((820, 400), (927, 495)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1156, 400), (1238, 485)) : 2

The minimum distance from car: 3 is 4.0

bbox:car_number ((1144, 427), (1152, 447)) : 3

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1073/1261 [47:56<08:23, 2.68s/it]

The times for each task are: [0.855134, 0.616337, 1.595622, 0.409359, 0.333567] with:

Minimum: 0.333567 Maximum: 1.595622 Average: 0.762 seconds

Number of cars 2

bbox:car_number ((834, 400), (927, 495)) : 1

```
The minimum distance from car: 1 is 7.0
bbox:car_number ((1158, 400), (1230, 472)) : 2
The minimum distance from car: 3 is 6.708203932499369
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 40
Length of task list: 5
Number of processes used: 3
```

85%| | 1074/1261 [47:59<08:21, 2.68s/it]

The times for each task are: [0.615701, 0.976424, 1.446254, 0.425986, 0.295734] with:

Minimum: 0.295734 Maximum: 1.446254 Average: 0.752 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (917, 495)) : 1
The minimum distance from car: 1 is 6.0
bbox:car_number ((1144, 400), (1241, 498)) : 2
The minimum distance from car: 3 is 13.152946437965905
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 40
Length of task list: 5
Number of processes used: 3
```

85%| | 1075/1261 [48:01<08:18, 2.68s/it]

The times for each task are: [1.303114, 0.583241, 0.946795, 0.369485, 0.284972] with:

Minimum: 0.284972 Maximum: 1.303114 Average: 0.6975 seconds

```
Number of cars 3
bbox:car_number ((832, 400), (927, 495)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1158, 400), (1238, 472)) : 2
The minimum distance from car: 3 is 14.317821063276353
bbox:car_number ((1144, 431), (1152, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 40
Length of task list: 5
```

Number of processes used: 3

85%| | 1076/1261 [48:04<08:15, 2.68s/it]

The times for each task are: [1.360991, 0.975299, 0.729659, 0.35947, 0.426778] with:

Minimum: 0.35947 Maximum: 1.360991 Average: 0.7704 seconds

Number of cars 2

bbox:car_number ((832, 400), (927, 495)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1144, 400), (1238, 498)) : 2

The minimum distance from car: 3 is 14.7648230602334

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1077/1261 [48:07<08:13, 2.68s/it]

The times for each task are: [0.858443, 1.424839, 0.602265, 0.318567, 0.497676] with:

Minimum: 0.318567 Maximum: 1.424839 Average: 0.7404 seconds

Number of cars 2

bbox:car_number ((832, 400), (927, 495)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1156, 400), (1241, 485)) : 2

The minimum distance from car: 3 is 9.899494936611665

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

85%| | 1078/1261 [48:10<08:10, 2.68s/it]

The times for each task are: [0.862599, 1.417573, 0.608168, 0.388468, 0.291862] with:

Minimum: 0.291862 Maximum: 1.417573 Average: 0.7137 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (927, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1144, 400), (1241, 512)) : 2
The minimum distance from car: 3 is 15.231546211727817
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

86%| | 1079/1261 [48:12<08:07, 2.68s/it]

The times for each task are: [0.586585, 1.365687, 1.008191, 0.4394, 0.28787] with:
Minimum: 0.28787 Maximum: 1.365687 Average: 0.7375 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (917, 495)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1156, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 9.899494936611665
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

86%| | 1080/1261 [48:15<08:05, 2.68s/it]

The times for each task are: [0.620261, 0.915852, 0.408773, 1.643324, 0.426555] with:
Minimum: 0.408773 Maximum: 1.643324 Average: 0.803 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (917, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1144, 400), (1241, 507)) : 2
The minimum distance from car: 3 is 8.06225774829855
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
```

```
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

86%| | 1081/1261 [48:18<08:02, 2.68s/it]

The times for each task are: [0.955503, 1.366332, 0.597221, 0.290748, 0.382696] with:

Minimum: 0.290748 Maximum: 1.366332 Average: 0.7185 seconds

```
Number of cars 2  
bbox:car_number ((818, 400), (927, 495)) : 1  
The minimum distance from car: 1 is 2.0  
bbox:car_number ((1156, 400), (1241, 477)) : 2  
The minimum distance from car: 3 is 16.15549442140351  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

86%| | 1082/1261 [48:20<07:59, 2.68s/it]

The times for each task are: [0.955519, 0.590895, 1.412809, 0.423372, 0.292945] with:

Minimum: 0.292945 Maximum: 1.412809 Average: 0.7351 seconds

```
Number of cars 2  
bbox:car_number ((832, 400), (931, 495)) : 1  
The minimum distance from car: 1 is 9.0  
bbox:car_number ((1156, 400), (1241, 485)) : 2  
The minimum distance from car: 3 is 4.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]  
Length of task list: 5  
Number of processes used: 3
```

86%| | 1083/1261 [48:23<07:57, 2.68s/it]

The times for each task are: [0.830081, 1.372829, 0.608801, 0.377946, 0.33854] with:

Minimum: 0.33854 Maximum: 1.372829 Average: 0.7056 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (931, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1144, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 8.602325267042627
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1084/1261 [48:26<07:54, 2.68s/it]

The times for each task are: [0.86642, 0.709041, 1.597284, 0.405775, 0.283699] with:

Minimum: 0.283699 Maximum: 1.597284 Average: 0.7724 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (917, 495)) : 1
The minimum distance from car: 1 is 7.0
bbox:car_number ((1144, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1085/1261 [48:28<07:51, 2.68s/it]

The times for each task are: [0.876775, 0.62815, 1.566666, 0.301637, 0.399351] with:

Minimum: 0.301637 Maximum: 1.566666 Average: 0.7545 seconds

```
Number of cars 2
bbox:car_number ((834, 400), (917, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1156, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1086/1261 [48:31<07:49, 2.68s/it]

The times for each task are: [1.350831, 0.856571, 0.71377, 0.438745, 0.391843] with:

Minimum: 0.391843 Maximum: 1.350831 Average: 0.7504 seconds

```
Number of cars 2
bbox:car_number ((832, 400), (927, 485)) : 1
The minimum distance from car: 1 is 6.4031242374328485
bbox:car_number ((1144, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1087/1261 [48:33<07:46, 2.68s/it]

The times for each task are: [0.655552, 0.926042, 0.34193, 1.414716, 0.505072] with:

Minimum: 0.34193 Maximum: 1.414716 Average: 0.7687 seconds

```
Number of cars 2
bbox:car_number ((836, 400), (915, 495)) : 1
The minimum distance from car: 1 is 6.4031242374328485
bbox:car_number ((1156, 400), (1242, 498)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

86%| | 1088/1261 [48:36<07:43, 2.68s/it]

The times for each task are: [0.653967, 0.996034, 1.431091, 0.469961, 0.359553] with:

Minimum: 0.359553 Maximum: 1.431091 Average: 0.7821 seconds

Number of cars 2

bbox:car_number ((832, 400), (927, 497)) : 1

The minimum distance from car: 1 is 4.123105625617661

bbox:car_number ((1156, 400), (1242, 498)) : 2

The minimum distance from car: 3 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

86%| | 1089/1261 [48:39<07:41, 2.68s/it]

The times for each task are: [1.369794, 0.889652, 0.671219, 0.345734, 0.40998] with:

Minimum: 0.345734 Maximum: 1.369794 Average: 0.7373 seconds

Number of cars 2

bbox:car_number ((820, 400), (915, 495)) : 1

The minimum distance from car: 1 is 12.041594578792296

bbox:car_number ((1144, 400), (1242, 497)) : 2

The minimum distance from car: 3 is 6.082762530298219

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

86%| | 1090/1261 [48:41<07:38, 2.68s/it]

The times for each task are: [0.562093, 1.346581, 1.008537, 0.337528, 0.420106] with:

Minimum: 0.337528 Maximum: 1.346581 Average: 0.735 seconds

Number of cars 2

bbox:car_number ((820, 400), (917, 497)) : 1

The minimum distance from car: 1 is 1.4142135623730951

bbox:car_number ((1150, 400), (1242, 497)) : 2

The minimum distance from car: 3 is 3.0

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1091/1261 [48:44<07:35, 2.68s/it]

The times for each task are: [1.395975, 0.897201, 0.677775, 0.424209, 0.318337] with:
Minimum: 0.318337 Maximum: 1.395975 Average: 0.7427 seconds

```
Number of cars 2
bbox:car_number ((820, 400), (915, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1156, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 8.06225774829855
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1092/1261 [48:46<07:32, 2.68s/it]

The times for each task are: [0.884914, 0.642686, 1.448833, 0.439947, 0.378366] with:
Minimum: 0.378366 Maximum: 1.448833 Average: 0.7589 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 485)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1144, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 6.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1093/1261 [48:49<07:30, 2.68s/it]

The times for each task are: [0.651403, 0.905512, 1.439713, 0.389897, 0.291048] with:

Minimum: 0.291048 Maximum: 1.439713 Average: 0.7355 seconds

Number of cars 2

bbox:car_number ((818, 400), (915, 495)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1156, 400), (1242, 498)) : 2

The minimum distance from car: 3 is 1.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 497))]

Length of task list: 5

Number of processes used: 3

87%| | 1094/1261 [48:52<07:27, 2.68s/it]

The times for each task are: [1.345325, 0.840441, 0.710435, 0.390603, 0.338089] with:

Minimum: 0.338089 Maximum: 1.345325 Average: 0.725 seconds

Number of cars 2

bbox:car_number ((813, 400), (915, 495)) : 1

The minimum distance from car: 1 is 2.0

bbox:car_number ((1156, 400), (1253, 498)) : 2

The minimum distance from car: 3 is 5.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 497))]

Length of task list: 5

Number of processes used: 3

87%| | 1095/1261 [48:54<07:24, 2.68s/it]

The times for each task are: [0.838205, 0.629306, 1.397701, 0.293066, 0.407792] with:

Minimum: 0.293066 Maximum: 1.397701 Average: 0.7132 seconds

Number of cars 2

bbox:car_number ((813, 400), (915, 495)) : 1

The minimum distance from car: 1 is 0.0

```
bbox:car_number ((1156, 400), (1253, 498)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

87%| | 1096/1261 [48:57<07:22, 2.68s/it]

The times for each task are: [0.855515, 1.445975, 0.690099, 0.282975, 0.382439] with:

Minimum: 0.282975 Maximum: 1.445975 Average: 0.7314 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (915, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1144, 400), (1257, 498)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

87%| | 1097/1261 [49:00<07:19, 2.68s/it]

The times for each task are: [0.892505, 0.678927, 1.570713, 0.397229, 0.38744] with:

Minimum: 0.38744 Maximum: 1.570713 Average: 0.7854 seconds

```
Number of cars 2
bbox:car_number ((820, 400), (915, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1156, 400), (1253, 497)) : 2
The minimum distance from car: 3 is 4.123105625617661
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

87%| | 1098/1261 [49:03<07:16, 2.68s/it]

The times for each task are: [0.988633, 1.37733, 0.568122, 0.523581, 0.367948] with:

Minimum: 0.367948 Maximum: 1.37733 Average: 0.7651 seconds

Number of cars 2

bbox:car_number ((818, 400), (915, 495)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1156, 400), (1257, 498)) : 2

The minimum distance from car: 3 is 2.23606797749979

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

87%| | 1099/1261 [49:05<07:14, 2.68s/it]

The times for each task are: [0.875612, 0.618839, 1.44836, 0.485269, 0.297505] with:

Minimum: 0.297505 Maximum: 1.44836 Average: 0.7451 seconds

Number of cars 2

bbox:car_number ((818, 400), (915, 497)) : 1

The minimum distance from car: 1 is 1.0

bbox:car_number ((1156, 400), (1257, 498)) : 2

The minimum distance from car: 3 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

87%| | 1100/1261 [49:08<07:11, 2.68s/it]

The times for each task are: [0.959876, 0.631663, 1.419747, 0.311124, 0.383677] with:

Minimum: 0.311124 Maximum: 1.419747 Average: 0.7412 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 497)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1156, 400), (1257, 485)) : 2
The minimum distance from car: 3 is 7.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1101/1261 [49:11<07:08, 2.68s/it]

The times for each task are: [0.614873, 0.888722, 0.409385, 1.380985, 0.290738] with:

Minimum: 0.290738 Maximum: 1.380985 Average: 0.7169 seconds

```
Number of cars 2
bbox:car_number ((820, 400), (915, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1160, 400), (1257, 498)) : 2
The minimum distance from car: 3 is 7.280109889280518
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

87%| | 1102/1261 [49:13<07:06, 2.68s/it]

The times for each task are: [1.014302, 1.467774, 0.688864, 0.41884, 0.308677] with:

Minimum: 0.308677 Maximum: 1.467774 Average: 0.7797 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (912, 495)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1156, 400), (1257, 498)) : 2
The minimum distance from car: 3 is 2.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
```

```
Length of task list: 5
Number of processes used: 3
```

```
87%| | 1103/1261 [49:16<07:03, 2.68s/it]
```

```
The times for each task are: [0.656945, 0.893156, 1.49273, 0.394841, 0.301942] with:
```

```
Minimum: 0.301942 Maximum: 1.49273 Average: 0.7479 seconds
```

```
Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1158, 400), (1257, 485)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

```
88%| | 1104/1261 [49:18<07:00, 2.68s/it]
```

```
The times for each task are: [0.97469, 0.588847, 1.472462, 0.527234, 0.33187] with:
```

```
Minimum: 0.33187 Maximum: 1.472462 Average: 0.779 seconds
```

```
Number of cars 2
bbox:car_number ((818, 400), (912, 485)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1144, 400), (1257, 498)) : 2
The minimum distance from car: 3 is 9.899494936611665
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

```
88%| | 1105/1261 [49:21<06:58, 2.68s/it]
```

```
The times for each task are: [0.989604, 1.415812, 0.737333, 0.296461, 0.429416] with:
```

Minimum: 0.296461 Maximum: 1.415812 Average: 0.7737 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 5.0990195135927845
bbox:car_number ((1165, 423), (1242, 495)) : 2
The minimum distance from car: 3 is 10.44030650891055
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88%| 1106/1261 [49:24<06:55, 2.68s/it]

The times for each task are: [0.631817, 0.893255, 1.6208, 0.399227, 0.29] with:

Minimum: 0.29 Maximum: 1.6208 Average: 0.767 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 497)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1156, 400), (1257, 507)) : 2
The minimum distance from car: 3 is 6.708203932499369
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88%| 1107/1261 [49:27<06:52, 2.68s/it]

The times for each task are: [0.902779, 0.602986, 1.662312, 0.478264, 0.362726] with:

Minimum: 0.362726 Maximum: 1.662312 Average: 0.8018 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 1.0
bbox:car_number ((1175, 400), (1257, 485)) : 2
The minimum distance from car: 3 is 14.866068747318506
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88% | 1108/1261 [49:30<06:50, 2.68s/it]

The times for each task are: [0.575822, 0.99406, 1.375653, 0.389299, 0.351596] with:

Minimum: 0.351596 Maximum: 1.375653 Average: 0.7373 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (908, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1175, 400), (1268, 498)) : 2
The minimum distance from car: 3 is 8.602325267042627
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88% | 1109/1261 [49:33<06:47, 2.68s/it]

The times for each task are: [0.894807, 1.457174, 0.687318, 0.41501, 0.389434] with:

Minimum: 0.389434 Maximum: 1.457174 Average: 0.7687 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1183, 400), (1257, 485)) : 2
The minimum distance from car: 3 is 7.0710678118654755
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

88% | 1110/1261 [49:35<06:44, 2.68s/it]

The times for each task are: [0.896278, 1.420949, 0.636464, 0.384788, 0.348365] with:

Minimum: 0.348365 Maximum: 1.420949 Average: 0.7374 seconds

Number of cars 2
bbox:car_number ((818, 400), (915, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1175, 400), (1272, 485)) : 2
The minimum distance from car: 3 is 3.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

88%| | 1111/1261 [49:38<06:42, 2.68s/it]

The times for each task are: [0.87356, 0.648588, 1.416403, 0.339842, 0.38126] with:

Minimum: 0.339842 Maximum: 1.416403 Average: 0.7319 seconds

Number of cars 2
bbox:car_number ((813, 400), (915, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1198, 400), (1268, 472)) : 2
The minimum distance from car: 3 is 11.661903789690601
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

88%| | 1112/1261 [49:40<06:39, 2.68s/it]

The times for each task are: [0.572441, 1.051727, 1.459453, 0.432208, 0.309302] with:

Minimum: 0.309302 Maximum: 1.459453 Average: 0.765 seconds

Number of cars 2
bbox:car_number ((820, 400), (915, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1144, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 28.178005607210743

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

88%| 1113/1261 [49:43<06:36, 2.68s/it]

The times for each task are: [0.848219, 0.55581, 0.282965, 0.379105, 1.80022] with:

Minimum: 0.282965 Maximum: 1.80022 Average: 0.7733 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (915, 495)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1175, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 16.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

88%| 1114/1261 [49:45<06:34, 2.68s/it]

The times for each task are: [1.351454, 0.574033, 1.033145, 0.40384, 0.33508] with:

Minimum: 0.33508 Maximum: 1.351454 Average: 0.7395 seconds

```
Number of cars 4
bbox:car_number ((813, 400), (912, 495)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1103, 400), (1105, 459)) : 2
bbox:car_number ((1175, 400), (1273, 485)) : 3
The minimum distance from car: 3 is 7.0
bbox:car_number ((1156, 427), (1162, 459)) : 4
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

88%| | 1115/1261 [49:48<06:31, 2.68s/it]

The times for each task are: [1.001087, 0.610437, 1.384789, 0.287089, 0.427939] with:

Minimum: 0.287089 Maximum: 1.384789 Average: 0.7423 seconds

Number of cars 2

bbox:car_number ((813, 400), (912, 495)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1156, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 12.206555615733702

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

89%| | 1116/1261 [49:51<06:28, 2.68s/it]

The times for each task are: [1.336966, 0.835352, 0.715052, 0.454023, 0.314613] with:

Minimum: 0.314613 Maximum: 1.336966 Average: 0.7312 seconds

Number of cars 3

bbox:car_number ((832, 400), (908, 495)) : 1

The minimum distance from car: 1 is 8.0

bbox:car_number ((1103, 400), (1107, 459)) : 2

bbox:car_number ((1120, 400), (1273, 512)) : 3

The minimum distance from car: 3 is 19.313207915827967

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

89%| | 1117/1261 [49:54<06:26, 2.68s/it]

The times for each task are: [0.626933, 0.853618, 1.494595, 0.383934, 0.352489] with:

Minimum: 0.352489 Maximum: 1.494595 Average: 0.7423 seconds

```
Number of cars 2
bbox:car_number ((818, 400), (908, 497)) : 1
The minimum distance from car: 1 is 7.0710678118654755
bbox:car_number ((1156, 400), (1273, 512)) : 2
The minimum distance from car: 3 is 18.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| 1118/1261 [49:57<06:23, 2.68s/it]

The times for each task are: [0.899598, 1.404633, 0.708143, 0.427125, 0.354676] with:

Minimum: 0.354676 Maximum: 1.404633 Average: 0.7588 seconds

```
Number of cars 3
bbox:car_number ((820, 400), (908, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
bbox:car_number ((1175, 400), (1273, 497)) : 2
The minimum distance from car: 3 is 12.806248474865697
bbox:car_number ((1160, 438), (1167, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| 1119/1261 [49:59<06:20, 2.68s/it]

The times for each task are: [0.887525, 1.358926, 0.605121, 0.397484, 0.364423] with:

Minimum: 0.364423 Maximum: 1.358926 Average: 0.7227 seconds

```
Number of cars 3
bbox:car_number ((820, 400), (908, 495)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1175, 400), (1272, 478)) : 2
The minimum distance from car: 3 is 9.055385138137417
bbox:car_number ((1122, 415), (1122, 447)) : 3
totalCars: 11
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))  
Length of task list: 5  
Number of processes used: 3
```

89%| | 1120/1261 [50:02<06:18, 2.68s/it]

The times for each task are: [0.856642, 0.60818, 1.358038, 0.40371, 0.350749] with:

Minimum: 0.350749 Maximum: 1.358038 Average: 0.7155 seconds

```
Number of cars 2  
bbox:car_number ((820, 400), (912, 495)) : 1  
The minimum distance from car: 1 is 2.0  
bbox:car_number ((1175, 400), (1273, 478)) : 2  
The minimum distance from car: 3 is 1.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))  
Length of task list: 5  
Number of processes used: 3
```

89%| | 1121/1261 [50:05<06:15, 2.68s/it]

The times for each task are: [0.850029, 1.37863, 0.702993, 0.301546, 0.397985] with:

Minimum: 0.301546 Maximum: 1.37863 Average: 0.7262 seconds

```
Number of cars 2  
bbox:car_number ((813, 400), (908, 495)) : 1  
The minimum distance from car: 1 is 6.0  
bbox:car_number ((1183, 400), (1273, 485)) : 2  
The minimum distance from car: 3 is 5.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 11  
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))  
Length of task list: 5  
Number of processes used: 3
```

89%| | 1122/1261 [50:08<06:12, 2.68s/it]

The times for each task are: [0.849197, 0.616148, 0.40259, 1.350424, 0.29079] with:

Minimum: 0.29079 Maximum: 1.350424 Average: 0.7018 seconds

Number of cars 2

bbox:car_number ((813, 400), (908, 485)) : 1

The minimum distance from car: 1 is 5.0

bbox:car_number ((1160, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 13.892443989449804

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

89% | 1123/1261 [50:11<06:10, 2.68s/it]

The times for each task are: [0.896493, 1.415528, 0.631056, 0.421984, 0.418015] with:

Minimum: 0.418015 Maximum: 1.415528 Average: 0.7566 seconds

Number of cars 2

bbox:car_number ((832, 400), (901, 485)) : 1

The minimum distance from car: 1 is 6.0

bbox:car_number ((1180, 400), (1273, 477)) : 2

The minimum distance from car: 3 is 14.866068747318506

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

89% | 1124/1261 [50:14<06:07, 2.68s/it]

The times for each task are: [0.623588, 0.926962, 1.443126, 0.392426, 0.287254] with:

Minimum: 0.287254 Maximum: 1.443126 Average: 0.7347 seconds

Number of cars 3

bbox:car_number ((813, 400), (901, 485)) : 1

The minimum distance from car: 1 is 9.0

bbox:car_number ((1156, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 16.278820596099706

```
bbox:car_number ((1144, 431), (1152, 447)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| 1125/1261 [50:16<06:04, 2.68s/it]

The times for each task are: [0.608989, 1.409907, 0.980561, 0.441891, 0.296743] with:

Minimum: 0.296743 Maximum: 1.409907 Average: 0.7476 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (901, 485)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1180, 400), (1272, 477)) : 2
The minimum distance from car: 3 is 16.278820596099706
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| 1126/1261 [50:19<06:01, 2.68s/it]

The times for each task are: [1.344785, 1.034976, 0.738756, 0.344557, 0.442203] with:

Minimum: 0.344557 Maximum: 1.344785 Average: 0.7811 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (901, 485)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1144, 400), (1273, 507)) : 2
The minimum distance from car: 3 is 23.430749027719962
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

89%| | 1127/1261 [50:22<05:59, 2.68s/it]

The times for each task are: [0.905586, 0.660251, 1.436438, 0.400215, 0.356551] with:

Minimum: 0.356551 Maximum: 1.436438 Average: 0.7518 seconds

Number of cars 2

bbox:car_number ((813, 400), (901, 485)) : 1

The minimum distance from car: 1 is 0.0

bbox:car_number ((1179, 400), (1273, 485)) : 2

The minimum distance from car: 3 is 20.223748416156685

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

89%| | 1128/1261 [50:24<05:56, 2.68s/it]

The times for each task are: [1.40446, 0.663639, 0.949084, 0.401764, 0.423863] with:

Minimum: 0.401764 Maximum: 1.40446 Average: 0.7686 seconds

Number of cars 2

bbox:car_number ((820, 400), (901, 485)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((1183, 400), (1273, 472)) : 2

The minimum distance from car: 3 is 6.324555320336759

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

90%| | 1129/1261 [50:27<05:53, 2.68s/it]

The times for each task are: [1.445081, 0.625756, 0.972754, 0.397266, 0.292759] with:

Minimum: 0.292759 Maximum: 1.445081 Average: 0.7467 seconds

Number of cars 2

bbox:car_number ((813, 400), (901, 485)) : 1

The minimum distance from car: 1 is 3.0

```
bbox:car_number ((1175, 400), (1273, 478)) : 2
The minimum distance from car: 3 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1130/1261 [50:30<05:51, 2.68s/it]

The times for each task are: [0.619409, 0.857225, 0.40864, 0.29643, 1.65712] with:

Minimum: 0.29643 Maximum: 1.65712 Average: 0.7678 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (901, 485)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1179, 400), (1273, 498)) : 2
The minimum distance from car: 3 is 10.198039027185569
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1131/1261 [50:32<05:48, 2.68s/it]

The times for each task are: [0.87047, 0.605032, 1.574865, 0.304698, 0.397339] with:

Minimum: 0.304698 Maximum: 1.574865 Average: 0.7505 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (908, 495)) : 1
The minimum distance from car: 1 is 5.830951894845301
bbox:car_number ((1183, 400), (1273, 497)) : 2
The minimum distance from car: 3 is 2.23606797749979
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1132/1261 [50:35<05:45, 2.68s/it]

The times for each task are: [0.971004, 0.623871, 1.525801, 0.44315, 0.360738] with:

Minimum: 0.360738 Maximum: 1.525801 Average: 0.7849 seconds

Number of cars 2

bbox:car_number ((805, 400), (901, 495)) : 1

The minimum distance from car: 1 is 7.0

bbox:car_number ((1175, 400), (1273, 498)) : 2

The minimum distance from car: 3 is 4.123105625617661

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

90%| | 1133/1261 [50:38<05:43, 2.68s/it]

The times for each task are: [1.358435, 0.867764, 0.629152, 0.394285, 0.307395] with:

Minimum: 0.307395 Maximum: 1.358435 Average: 0.7114 seconds

Number of cars 2

bbox:car_number ((820, 400), (901, 485)) : 1

The minimum distance from car: 1 is 8.602325267042627

bbox:car_number ((1217, 430), (1268, 459)) : 2

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

90%| | 1134/1261 [50:40<05:40, 2.68s/it]

The times for each task are: [0.649518, 0.91202, 1.362516, 0.310397, 0.387811] with:

Minimum: 0.310397 Maximum: 1.362516 Average: 0.7245 seconds

Number of cars 2

bbox:car_number ((805, 400), (897, 495)) : 1

The minimum distance from car: 1 is 10.295630140987

```
bbox:car_number ((1179, 430), (1241, 497)) : 2
The minimum distance from car: 3 is 10.198039027185569
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1135/1261 [50:43<05:37, 2.68s/it]

The times for each task are: [1.32852, 0.689522, 0.897521, 0.361799, 0.481881] with:

Minimum: 0.361799 Maximum: 1.32852 Average: 0.7518 seconds

```
Number of cars 2
bbox:car_number ((813, 400), (901, 485)) : 1
The minimum distance from car: 1 is 7.810249675906654
bbox:car_number ((1204, 419), (1273, 478)) : 2
The minimum distance from car: 3 is 14.035668847618199
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1136/1261 [50:46<05:35, 2.68s/it]

The times for each task are: [0.973643, 0.600587, 1.363857, 0.4488, 0.291809] with:

Minimum: 0.291809 Maximum: 1.363857 Average: 0.7357 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (897, 485)) : 1
The minimum distance from car: 1 is 6.0
bbox:car_number ((1183, 427), (1268, 485)) : 2
The minimum distance from car: 3 is 15.264337522473747
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1137/1261 [50:49<05:32, 2.68s/it]

The times for each task are: [0.820833, 0.679043, 1.340283, 0.417665, 0.430862] with:

Minimum: 0.417665 Maximum: 1.340283 Average: 0.7377 seconds

Number of cars 2

bbox:car_number ((805, 400), (901, 495)) : 1

The minimum distance from car: 1 is 5.385164807134504

bbox:car_number ((1204, 431), (1268, 472)) : 2

The minimum distance from car: 3 is 12.083045973594572

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

90%| | 1138/1261 [50:51<05:29, 2.68s/it]

The times for each task are: [0.893097, 0.74313, 0.34883, 1.520523, 0.476086] with:

Minimum: 0.34883 Maximum: 1.520523 Average: 0.7963 seconds

Number of cars 2

bbox:car_number ((805, 400), (897, 478)) : 1

The minimum distance from car: 1 is 8.246211251235321

bbox:car_number ((1175, 431), (1268, 497)) : 2

The minimum distance from car: 3 is 11.045361017187261

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

90%| | 1139/1261 [50:54<05:27, 2.68s/it]

The times for each task are: [0.864503, 0.637316, 1.421185, 0.435448, 0.276603] with:

Minimum: 0.276603 Maximum: 1.421185 Average: 0.727 seconds

```
Number of cars 3
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 5.0
bbox:car_number ((1204, 430), (1273, 498)) : 2
The minimum distance from car: 3 is 13.152946437965905
bbox:car_number ((1179, 438), (1185, 472)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1140/1261 [50:57<05:24, 2.68s/it]

The times for each task are: [0.85978, 0.63333, 1.537724, 0.409024, 0.300248] with:

Minimum: 0.300248 Maximum: 1.537724 Average: 0.748 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1175, 427), (1273, 516)) : 2
The minimum distance from car: 3 is 7.615773105863909
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

90%| | 1141/1261 [50:59<05:21, 2.68s/it]

The times for each task are: [0.620371, 0.888674, 1.419116, 0.406188, 0.314311] with:

Minimum: 0.314311 Maximum: 1.419116 Average: 0.7297 seconds

```
Number of cars 3
bbox:car_number ((803, 400), (897, 485)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1175, 423), (1268, 498)) : 2
The minimum distance from car: 3 is 11.40175425099138
bbox:car_number ((1156, 431), (1167, 462)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91%| | 1142/1261 [51:02<05:19, 2.68s/it]

The times for each task are: [0.869503, 0.592506, 1.438745, 0.489337, 0.292739] with:

Minimum: 0.292739 Maximum: 1.438745 Average: 0.7366 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (897, 485)) : 1
```

The minimum distance from car: 1 is 1.0

```
bbox:car_number ((1175, 427), (1272, 512)) : 2
```

The minimum distance from car: 3 is 9.219544457292887

```
totalCars: 11
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
```

```
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
```

Length of task list: 5

Number of processes used: 3

91%| | 1143/1261 [51:05<05:16, 2.68s/it]

The times for each task are: [0.639456, 0.936826, 1.457986, 0.388586, 0.2899] with:

Minimum: 0.2899 Maximum: 1.457986 Average: 0.7426 seconds

```
Number of cars 2
```

```
bbox:car_number ((805, 400), (897, 485)) : 1
```

The minimum distance from car: 1 is 0.0

```
bbox:car_number ((1183, 430), (1272, 498)) : 2
```

The minimum distance from car: 3 is 6.4031242374328485

```
totalCars: 11
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
```

```
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
```

Length of task list: 5

Number of processes used: 3

91%| | 1144/1261 [51:07<05:13, 2.68s/it]

The times for each task are: [0.864322, 1.337935, 0.677225, 0.409865, 0.296336] with:

Minimum: 0.296336 Maximum: 1.337935 Average: 0.7171 seconds

Number of cars 2

bbox:car_number ((799, 400), (897, 495)) : 1

The minimum distance from car: 1 is 5.830951894845301

bbox:car_number ((1183, 460), (1227, 507)) : 2

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

91% | 1145/1261 [51:10<05:11, 2.68s/it]

The times for each task are: [0.575579, 0.861563, 1.604486, 0.421506, 0.333331] with:

Minimum: 0.333331 Maximum: 1.604486 Average: 0.7593 seconds

Number of cars 2

bbox:car_number ((805, 400), (896, 478)) : 1

The minimum distance from car: 1 is 8.246211251235321

bbox:car_number ((1204, 460), (1227, 472)) : 2

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

91% | 1146/1261 [51:12<05:08, 2.68s/it]

The times for each task are: [1.436881, 0.735371, 1.058815, 0.38048, 0.304161] with:

Minimum: 0.304161 Maximum: 1.436881 Average: 0.7831 seconds

Number of cars 2

bbox:car_number ((805, 400), (897, 485)) : 1

The minimum distance from car: 1 is 3.1622776601683795

bbox:car_number ((1204, 430), (1268, 477)) : 2

The minimum distance from car: 3 is 11.180339887498949

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

91%| 1147/1261 [51:15<05:05, 2.68s/it]

The times for each task are: [0.588801, 0.9676, 1.551208, 0.361841, 0.443276] with:

Minimum: 0.361841 Maximum: 1.551208 Average: 0.7825 seconds

Number of cars 2

bbox:car_number ((803, 400), (897, 495)) : 1

The minimum distance from car: 1 is 5.0990195135927845

bbox:car_number ((1183, 427), (1268, 485)) : 2

The minimum distance from car: 3 is 8.246211251235321

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]

Length of task list: 5

Number of processes used: 3

91%| 1148/1261 [51:18<05:02, 2.68s/it]

The times for each task are: [0.862211, 0.612218, 1.552711, 0.400967, 0.27899] with:

Minimum: 0.27899 Maximum: 1.552711 Average: 0.7414 seconds

Number of cars 4

bbox:car_number ((805, 400), (885, 485)) : 1

The minimum distance from car: 1 is 7.0710678118654755

bbox:car_number ((1195, 427), (1268, 478)) : 2

The minimum distance from car: 3 is 5.0990195135927845

bbox:car_number ((595, 460), (627, 492)) : 3

bbox:car_number ((640, 460), (647, 497)) : 4

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 11

Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

Length of task list: 5

Number of processes used: 3

91%| 1149/1261 [51:20<05:00, 2.68s/it]

The times for each task are: [0.611612, 0.934244, 1.495232, 0.394893, 0.401391] with:

Minimum: 0.394893 Maximum: 1.495232 Average: 0.7675 seconds

```
Number of cars 2
bbox:car_number ((803, 400), (885, 478)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((1195, 427), (1268, 478)) : 2
The minimum distance from car: 3 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

91%| 1150/1261 [51:23<04:57, 2.68s/it]

The times for each task are: [1.378672, 1.003651, 0.686642, 0.413341, 0.338043] with:

Minimum: 0.338043 Maximum: 1.378672 Average: 0.7641 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (890, 478)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1183, 427), (1272, 497)) : 2
The minimum distance from car: 3 is 6.324555320336759
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91%|| 1151/1261 [51:26<04:54, 2.68s/it]

The times for each task are: [0.550267, 1.368019, 0.933463, 0.483448, 0.288018] with:

Minimum: 0.288018 Maximum: 1.368019 Average: 0.7246 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1179, 427), (1268, 497)) : 2
The minimum distance from car: 3 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91%|| 1152/1261 [51:29<04:52, 2.68s/it]

The times for each task are: [0.824301, 0.691292, 1.513828, 0.35617, 0.449355] with:

Minimum: 0.35617 Maximum: 1.513828 Average: 0.767 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (896, 485)) : 1
The minimum distance from car: 1 is 3.0
bbox:car_number ((1181, 427), (1268, 507)) : 2
The minimum distance from car: 3 is 5.0990195135927845
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3
```

91%|| 1153/1261 [51:31<04:49, 2.68s/it]

The times for each task are: [0.923526, 0.614244, 1.437989, 0.511761, 0.408676] with:

Minimum: 0.408676 Maximum: 1.437989 Average: 0.7792 seconds

```
Number of cars 1
bbox:car_number ((803, 400), (896, 495)) : 1
The minimum distance from car: 1 is 5.0990195135927845
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1154/1261 [51:34<04:46, 2.68s/it]

The times for each task are: [0.673877, 0.92628, 1.488367, 0.460783, 0.298992] with:

Minimum: 0.298992 Maximum: 1.488367 Average: 0.7697 seconds

```
Number of cars 1
bbox:car_number ((803, 400), (885, 478)) : 1
The minimum distance from car: 1 is 9.433981132056603
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1155/1261 [51:37<04:44, 2.68s/it]

The times for each task are: [0.904001, 1.412144, 0.706547, 0.451823, 0.319025] with:
Minimum: 0.319025 Maximum: 1.412144 Average: 0.7587 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 3.1622776601683795
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1156/1261 [51:40<04:41, 2.68s/it]

The times for each task are: [0.908543, 0.645852, 1.583576, 0.396857, 0.294617] with:
Minimum: 0.294617 Maximum: 1.583576 Average: 0.7659 seconds

```
Number of cars 1
bbox:car_number ((803, 400), (885, 485)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1157/1261 [51:42<04:38, 2.68s/it]

The times for each task are: [0.84452, 1.371163, 0.700496, 0.29416, 0.387362] with:
Minimum: 0.29416 Maximum: 1.371163 Average: 0.7195 seconds

Number of cars 1

```
bbox:car_number ((805, 400), (890, 478)) : 1
The minimum distance from car: 1 is 4.242640687119285
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1158/1261 [51:45<04:36, 2.68s/it]

The times for each task are: [0.61351, 0.855896, 1.404114, 0.410323, 0.367115] with:

Minimum: 0.367115 Maximum: 1.404114 Average: 0.7302 seconds

```
Number of cars 1
bbox:car_number ((799, 400), (890, 485)) : 1
The minimum distance from car: 1 is 4.242640687119285
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1159/1261 [51:47<04:33, 2.68s/it]

The times for each task are: [0.862828, 0.677442, 1.648193, 0.385722, 0.33871] with:

Minimum: 0.33871 Maximum: 1.648193 Average: 0.7826 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 6.4031242374328485
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1160/1261 [51:50<04:30, 2.68s/it]

The times for each task are: [0.88094, 0.752961, 1.512511, 0.327567, 0.384979] with:

Minimum: 0.327567 Maximum: 1.512511 Average: 0.7718 seconds

```
Number of cars 1
bbox:car_number ((820, 415), (882, 477)) : 1
```

```
The minimum distance from car: 1 is 8.246211251235321
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1161/1261 [51:53<04:28, 2.68s/it]

The times for each task are: [0.925208, 0.626984, 1.460858, 0.317265, 0.470735] with:

Minimum: 0.317265 Maximum: 1.460858 Average: 0.7602 seconds

```
Number of cars 3
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 8.246211251235321
bbox:car_number ((1183, 454), (1185, 495)) : 2
bbox:car_number ((1210, 454), (1257, 477)) : 3
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1162/1261 [51:55<04:25, 2.68s/it]

The times for each task are: [1.388382, 0.686415, 1.051972, 0.416057, 0.33368] with:

Minimum: 0.33368 Maximum: 1.388382 Average: 0.7753 seconds

```
Number of cars 2
bbox:car_number ((805, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.0
bbox:car_number ((1210, 430), (1272, 507)) : 2
The minimum distance from car: 3 is 17.029386365926403
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 11
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 512))]
Length of task list: 5
Number of processes used: 3
```

92%|| 1163/1261 [51:58<04:22, 2.68s/it]

The times for each task are: [0.95748, 0.592546, 1.628314, 0.321074, 0.475491] with:

Minimum: 0.321074 Maximum: 1.628314 Average: 0.795 seconds

Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

92%|| 1164/1261 [52:01<04:20, 2.68s/it]

The times for each task are: [0.629568, 1.411662, 1.069288, 0.403501, 0.286014] with:

Minimum: 0.286014 Maximum: 1.411662 Average: 0.76 seconds

Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

92%|| 1165/1261 [52:03<04:17, 2.68s/it]

The times for each task are: [0.590113, 0.863824, 1.560539, 0.480298, 0.298142] with:

Minimum: 0.298142 Maximum: 1.560539 Average: 0.7586 seconds

Number of cars 1
bbox:car_number ((805, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

92%|| 1166/1261 [52:06<04:14, 2.68s/it]

The times for each task are: [0.846086, 1.410455, 0.699084, 0.406075, 0.287477] with:

Minimum: 0.287477 Maximum: 1.410455 Average: 0.7298 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

93%|| 1167/1261 [52:09<04:12, 2.68s/it]

The times for each task are: [0.635815, 0.930931, 0.481605, 0.29727, 1.696238] with:

Minimum: 0.29727 Maximum: 1.696238 Average: 0.8084 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

93%|| 1168/1261 [52:12<04:09, 2.68s/it]

The times for each task are: [0.856418, 0.645999, 1.558073, 0.382393, 0.287534] with:

Minimum: 0.287534 Maximum: 1.558073 Average: 0.7461 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 472)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

93%|| 1169/1261 [52:15<04:06, 2.68s/it]

The times for each task are: [0.616147, 1.04224, 1.586411, 0.392207, 0.338955] with:

Minimum: 0.338955 Maximum: 1.586411 Average: 0.7952 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 11.180339887498949
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1170/1261 [52:17<04:04, 2.68s/it]

```
The times for each task are: [0.633713, 0.891965, 0.410227, 1.610061, 0.315642] with:
Minimum: 0.315642 Maximum: 1.610061 Average: 0.7723 seconds
```

```
Number of cars 2
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 5.385164807134504
bbox:car_number ((1225, 460), (1257, 477)) : 2
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1171/1261 [52:20<04:01, 2.68s/it]

```
The times for each task are: [0.616906, 1.395657, 0.972777, 0.418598, 0.326898] with:
Minimum: 0.326898 Maximum: 1.395657 Average: 0.7462 seconds
```

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1172/1261 [52:23<03:58, 2.68s/it]

```
The times for each task are: [0.862831, 0.666855, 1.380213, 0.394285, 0.376636] with:
Minimum: 0.376636 Maximum: 1.380213 Average: 0.7362 seconds
```

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1173/1261 [52:25<03:55, 2.68s/it]

The times for each task are: [1.047238, 0.581835, 1.398652, 0.285159, 0.500296] with:
Minimum: 0.285159 Maximum: 1.398652 Average: 0.7626 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 485)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1174/1261 [52:28<03:53, 2.68s/it]

The times for each task are: [1.362017, 1.011279, 0.666975, 0.448339, 0.345546] with:
Minimum: 0.345546 Maximum: 1.362017 Average: 0.7668 seconds

```
Number of cars 1
bbox:car_number ((803, 400), (890, 497)) : 1
The minimum distance from car: 1 is 6.708203932499369
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1175/1261 [52:31<03:50, 2.68s/it]

The times for each task are: [0.706933, 0.841851, 1.373378, 0.306707, 0.441568] with:
Minimum: 0.306707 Maximum: 1.373378 Average: 0.7341 seconds

Number of cars 1

```
bbox:car_number ((805, 400), (885, 495)) : 1
The minimum distance from car: 1 is 1.4142135623730951
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1176/1261 [52:33<03:47, 2.68s/it]

The times for each task are: [0.971047, 1.282559, 0.691706, 0.428705, 0.326283] with:
Minimum: 0.326283 Maximum: 1.282559 Average: 0.7401 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 478)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1177/1261 [52:36<03:45, 2.68s/it]

The times for each task are: [0.830047, 0.661758, 1.564404, 0.404557, 0.335295] with:
Minimum: 0.335295 Maximum: 1.564404 Average: 0.7592 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 4.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

93%|| 1178/1261 [52:38<03:42, 2.68s/it]

The times for each task are: [0.540553, 1.329675, 1.02673, 0.379063, 0.291636] with:
Minimum: 0.291636 Maximum: 1.329675 Average: 0.7135 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
```

The minimum distance from car: 1 is 1.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

93%|| 1179/1261 [52:41<03:39, 2.68s/it]

The times for each task are: [0.896324, 1.387536, 0.712578, 0.435181, 0.355068] with:

Minimum: 0.355068 Maximum: 1.387536 Average: 0.7573 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 478)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1180/1261 [52:44<03:37, 2.68s/it]

The times for each task are: [0.805666, 1.369838, 0.677408, 0.305273, 0.391935] with:

Minimum: 0.305273 Maximum: 1.369838 Average: 0.71 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 477)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1181/1261 [52:46<03:34, 2.68s/it]

The times for each task are: [0.605645, 0.964162, 1.495534, 0.376838, 0.33245] with:

Minimum: 0.33245 Maximum: 1.495534 Average: 0.7549 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 477)) : 1

The minimum distance from car: 1 is 0.0

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

94%|| 1182/1261 [52:49<03:31, 2.68s/it]

The times for each task are: [0.850951, 1.535436, 0.684818, 0.394629, 0.333549] with:

Minimum: 0.333549 Maximum: 1.535436 Average: 0.7599 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

94%|| 1183/1261 [52:52<03:29, 2.68s/it]

The times for each task are: [1.431851, 0.673807, 0.975717, 0.419487, 0.370208] with:

Minimum: 0.370208 Maximum: 1.431851 Average: 0.7742 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

94%|| 1184/1261 [52:55<03:26, 2.68s/it]

The times for each task are: [0.636547, 0.887591, 1.41785, 0.409078, 0.364639] with:

Minimum: 0.364639 Maximum: 1.41785 Average: 0.7431 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (890, 492)) : 1
The minimum distance from car: 1 is 8.246211251235321
totalCars: 11
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
Length of task list: 5  
Number of processes used: 3
```

94%|| 1185/1261 [52:57<03:23, 2.68s/it]

The times for each task are: [0.635213, 0.863531, 1.427351, 0.381719, 0.383429] with:

Minimum: 0.381719 Maximum: 1.427351 Average: 0.7382 seconds

```
Number of cars 2  
bbox:car_number ((805, 415), (890, 485)) : 1  
The minimum distance from car: 1 is 5.656854249492381  
bbox:car_number ((1165, 415), (1227, 462)) : 2  
The minimum distance from car: 3 is 37.69615364994153  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
totalCars: 11  
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))  
Length of task list: 5  
Number of processes used: 3
```

94%|| 1186/1261 [53:00<03:21, 2.68s/it]

The times for each task are: [0.579345, 1.528067, 0.895966, 0.302479, 0.458888] with:

Minimum: 0.302479 Maximum: 1.528067 Average: 0.7529 seconds

```
Number of cars 1  
bbox:car_number ((805, 400), (890, 492)) : 1  
The minimum distance from car: 1 is 4.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
Length of task list: 5  
Number of processes used: 3
```

94%|| 1187/1261 [53:02<03:18, 2.68s/it]

The times for each task are: [1.292979, 0.802931, 0.663149, 0.336564, 0.38209] with:

Minimum: 0.336564 Maximum: 1.292979 Average: 0.6955 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 492)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1188/1261 [53:05<03:15, 2.68s/it]

The times for each task are: [0.906693, 1.694878, 0.602164, 0.376573, 0.427998] with:

Minimum: 0.376573 Maximum: 1.694878 Average: 0.8017 seconds

Number of cars 1

bbox:car_number ((813, 415), (885, 477)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1189/1261 [53:08<03:13, 2.68s/it]

The times for each task are: [1.005102, 0.586634, 1.435349, 0.45963, 0.35129] with:

Minimum: 0.35129 Maximum: 1.435349 Average: 0.7676 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 2.23606797749979

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

94%|| 1190/1261 [53:11<03:10, 2.68s/it]

The times for each task are: [1.38587, 0.96876, 0.684375, 0.430655, 0.352697] with:

Minimum: 0.352697 Maximum: 1.38587 Average: 0.7645 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 485)) : 1

```
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

94%|| 1191/1261 [53:14<03:07, 2.68s/it]

The times for each task are: [0.869546, 0.631183, 1.399398, 0.309103, 0.430398] with:

Minimum: 0.309103 Maximum: 1.399398 Average: 0.7279 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1192/1261 [53:16<03:05, 2.68s/it]

The times for each task are: [0.840634, 1.411988, 0.604611, 0.401905, 0.309559] with:

Minimum: 0.309559 Maximum: 1.411988 Average: 0.7137 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (890, 485)) : 1
The minimum distance from car: 1 is 6.4031242374328485
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1193/1261 [53:19<03:02, 2.68s/it]

The times for each task are: [0.613614, 1.367166, 0.904533, 0.287333, 0.37043] with:

Minimum: 0.287333 Maximum: 1.367166 Average: 0.7086 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (890, 485)) : 1
The minimum distance from car: 1 is 0.0
```

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1194/1261 [53:22<02:59, 2.68s/it]

The times for each task are: [0.598563, 0.968149, 1.321086, 0.406544, 0.364532] with:

Minimum: 0.364532 Maximum: 1.321086 Average: 0.7318 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 477)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1195/1261 [53:24<02:57, 2.68s/it]

The times for each task are: [0.883161, 0.6987, 1.569173, 0.41297, 0.338417] with:

Minimum: 0.338417 Maximum: 1.569173 Average: 0.7805 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 4.47213595499958
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

95%|| 1196/1261 [53:27<02:54, 2.68s/it]

The times for each task are: [0.837176, 1.490669, 0.73216, 0.400101, 0.349376] with:

Minimum: 0.349376 Maximum: 1.490669 Average: 0.7619 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 498)) : 1
The minimum distance from car: 1 is 7.0
totalCars: 11
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
Length of task list: 5  
Number of processes used: 3
```

95%|| 1197/1261 [53:29<02:51, 2.68s/it]

The times for each task are: [0.826156, 1.584487, 0.689187, 0.386081, 0.335043] with:

Minimum: 0.335043 Maximum: 1.584487 Average: 0.7642 seconds

```
Number of cars 1  
bbox:car_number ((813, 400), (885, 485)) : 1  
The minimum distance from car: 1 is 7.280109889280518  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
Length of task list: 5  
Number of processes used: 3
```

95%|| 1198/1261 [53:32<02:48, 2.68s/it]

The times for each task are: [0.660259, 0.917664, 1.629226, 0.308713, 0.536402] with:

Minimum: 0.308713 Maximum: 1.629226 Average: 0.8105 seconds

```
Number of cars 1  
bbox:car_number ((805, 400), (885, 485)) : 1  
The minimum distance from car: 1 is 4.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))  
Length of task list: 5  
Number of processes used: 3
```

95%|| 1199/1261 [53:35<02:46, 2.68s/it]

The times for each task are: [0.618029, 0.932646, 1.659, 0.335598, 0.446948] with:

Minimum: 0.335598 Maximum: 1.659 Average: 0.7984 seconds

```
Number of cars 1  
bbox:car_number ((813, 400), (885, 478)) : 1  
The minimum distance from car: 1 is 5.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))
```

```
Length of task list: 5
Number of processes used: 3
```

```
95%|| 1200/1261 [53:38<02:43, 2.68s/it]
```

```
The times for each task are: [0.892444, 1.341886, 0.40563, 0.738488, 0.299458] with:
```

```
Minimum: 0.299458 Maximum: 1.341886 Average: 0.7356 seconds
```

```
Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
95%|| 1201/1261 [53:41<02:40, 2.68s/it]
```

```
The times for each task are: [0.893039, 0.601287, 1.500701, 0.469743, 0.287804] with:
```

```
Minimum: 0.287804 Maximum: 1.500701 Average: 0.7505 seconds
```

```
Number of cars 1
bbox:car_number ((805, 400), (890, 485)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
95%|| 1202/1261 [53:43<02:38, 2.68s/it]
```

```
The times for each task are: [0.847183, 0.576434, 1.566581, 0.333165, 0.396144] with:
```

```
Minimum: 0.333165 Maximum: 1.566581 Average: 0.7439 seconds
```

```
Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
```

Number of processes used: 3

95%|| 1203/1261 [53:46<02:35, 2.68s/it]

The times for each task are: [0.852984, 1.445572, 0.724274, 0.415336, 0.301846] with:

Minimum: 0.301846 Maximum: 1.445572 Average: 0.748 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

95%|| 1204/1261 [53:49<02:32, 2.68s/it]

The times for each task are: [0.868049, 0.633984, 0.287974, 1.801483, 0.410155] with:

Minimum: 0.287974 Maximum: 1.801483 Average: 0.8003 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 485)) : 1

The minimum distance from car: 1 is 5.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1205/1261 [53:51<02:30, 2.68s/it]

The times for each task are: [0.83085, 0.635015, 1.411997, 0.49877, 0.431141] with:

Minimum: 0.431141 Maximum: 1.411997 Average: 0.7616 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 5.385164807134504

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1206/1261 [53:55<02:27, 2.68s/it]

The times for each task are: [0.657198, 0.995843, 1.592983, 0.419496, 0.343757] with:

Minimum: 0.343757 Maximum: 1.592983 Average: 0.8019 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1207/1261 [53:57<02:24, 2.68s/it]

The times for each task are: [0.62671, 1.437516, 0.922083, 0.297855, 0.432882] with:

Minimum: 0.297855 Maximum: 1.437516 Average: 0.7434 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 485)) : 1

The minimum distance from car: 1 is 5.385164807134504

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1208/1261 [54:00<02:22, 2.68s/it]

The times for each task are: [0.57287, 0.909892, 1.358466, 0.29524, 0.43232] with:

Minimum: 0.29524 Maximum: 1.358466 Average: 0.7138 seconds

Number of cars 1

bbox:car_number ((805, 400), (896, 495)) : 1

The minimum distance from car: 1 is 5.0990195135927845

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1209/1261 [54:02<02:19, 2.68s/it]

The times for each task are: [0.845958, 0.59897, 1.635064, 0.348443, 0.403924] with:

Minimum: 0.348443 Maximum: 1.635064 Average: 0.7665 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 485)) : 1

The minimum distance from car: 1 is 7.0710678118654755

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1210/1261 [54:05<02:16, 2.68s/it]

The times for each task are: [0.70914, 0.864369, 1.426922, 0.456919, 0.312985] with:

Minimum: 0.312985 Maximum: 1.426922 Average: 0.7541 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 5.385164807134504

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1211/1261 [54:08<02:14, 2.68s/it]

The times for each task are: [0.941255, 0.576763, 1.622296, 0.295305, 0.463373] with:

Minimum: 0.295305 Maximum: 1.622296 Average: 0.7798 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 495)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1212/1261 [54:10<02:11, 2.68s/it]

The times for each task are: [0.902685, 0.62106, 0.300815, 1.718305, 0.394243] with:

Minimum: 0.300815 Maximum: 1.718305 Average: 0.7874 seconds

Number of cars 1

bbox:car_number ((805, 400), (885, 485)) : 1

The minimum distance from car: 1 is 5.385164807134504

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1213/1261 [54:13<02:08, 2.68s/it]

The times for each task are: [0.84347, 0.627541, 1.4514, 0.289899, 0.390232] with:

Minimum: 0.289899 Maximum: 1.4514 Average: 0.7205 seconds

Number of cars 1

bbox:car_number ((813, 400), (885, 485)) : 1

The minimum distance from car: 1 is 4.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1214/1261 [54:15<02:06, 2.68s/it]

The times for each task are: [0.896612, 0.608305, 1.417792, 0.308277, 0.41324] with:

Minimum: 0.308277 Maximum: 1.417792 Average: 0.7288 seconds

Number of cars 1

bbox:car_number ((805, 400), (890, 485)) : 1

The minimum distance from car: 1 is 2.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

96%|| 1215/1261 [54:18<02:03, 2.68s/it]

The times for each task are: [0.664936, 1.319182, 0.960607, 0.327544, 0.403158] with:

Minimum: 0.327544 Maximum: 1.319182 Average: 0.7351 seconds

Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 2.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

96%|| 1216/1261 [54:21<02:00, 2.68s/it]

The times for each task are: [0.620049, 0.897243, 1.59628, 0.385819, 0.301455] with:

Minimum: 0.301455 Maximum: 1.59628 Average: 0.7602 seconds

Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

97%|| 1217/1261 [54:23<01:57, 2.68s/it]

The times for each task are: [0.931635, 1.362442, 0.734484, 0.39319, 0.350969] with:

Minimum: 0.350969 Maximum: 1.362442 Average: 0.7545 seconds

Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

97%|| 1218/1261 [54:25<01:55, 2.68s/it]

The times for each task are: [0.877912, 1.369053, 0.322804, 0.414114, 0.744957] with:

Minimum: 0.322804 Maximum: 1.369053 Average: 0.7458 seconds

```
Number of cars 1
bbox:car_number ((813, 400), (885, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

97%|| 1219/1261 [54:28<01:52, 2.68s/it]

The times for each task are: [0.659024, 0.902872, 1.595104, 0.39858, 0.308906] with:

Minimum: 0.308906 Maximum: 1.595104 Average: 0.7729 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

97%|| 1220/1261 [54:30<01:49, 2.68s/it]

The times for each task are: [0.893263, 1.421784, 0.668674, 0.405068, 0.314131] with:

Minimum: 0.314131 Maximum: 1.421784 Average: 0.7406 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

97%|| 1221/1261 [54:33<01:47, 2.68s/it]

The times for each task are: [0.847861, 1.408969, 0.689303, 0.306587, 0.39383] with:

Minimum: 0.306587 Maximum: 1.408969 Average: 0.7293 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1222/1261 [54:36<01:44, 2.68s/it]

The times for each task are: [0.88641, 1.357156, 0.70737, 0.389301, 0.290046] with:

Minimum: 0.290046 Maximum: 1.357156 Average: 0.7261 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1223/1261 [54:38<01:41, 2.68s/it]

The times for each task are: [0.843319, 1.655338, 0.422607, 0.723746, 0.318466] with:

Minimum: 0.318466 Maximum: 1.655338 Average: 0.7927 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (890, 495)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1224/1261 [54:41<01:39, 2.68s/it]

The times for each task are: [0.897567, 0.621675, 1.366382, 0.316565, 0.542437] with:

Minimum: 0.316565 Maximum: 1.366382 Average: 0.7489 seconds

Number of cars 1

```
bbox:car_number ((813, 419), (885, 485)) : 1
The minimum distance from car: 1 is 5.385164807134504
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1225/1261 [54:43<01:36, 2.68s/it]

The times for each task are: [0.606728, 1.3673, 0.864383, 0.290092, 0.403378] with:

Minimum: 0.290092 Maximum: 1.3673 Average: 0.7064 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (885, 485)) : 1
The minimum distance from car: 1 is 10.770329614269007
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1226/1261 [54:46<01:33, 2.68s/it]

The times for each task are: [0.826553, 0.697731, 1.604409, 0.40808, 0.340932] with:

Minimum: 0.340932 Maximum: 1.604409 Average: 0.7755 seconds

```
Number of cars 1
bbox:car_number ((799, 400), (885, 485)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

97%|| 1227/1261 [54:48<01:31, 2.68s/it]

The times for each task are: [0.937437, 0.617771, 1.579298, 0.297332, 0.423671] with:

Minimum: 0.297332 Maximum: 1.579298 Average: 0.7711 seconds

```
Number of cars 1
bbox:car_number ((799, 400), (885, 485)) : 1
```

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

97%|| 1228/1261 [54:51<01:28, 2.68s/it]

The times for each task are: [0.664612, 0.91532, 1.424166, 0.289199, 0.484428] with:

Minimum: 0.289199 Maximum: 1.424166 Average: 0.7555 seconds

Number of cars 1

bbox:car_number ((799, 400), (885, 485)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

97%|| 1229/1261 [54:54<01:25, 2.68s/it]

The times for each task are: [0.980012, 0.640127, 1.369858, 0.465078, 0.341367] with:

Minimum: 0.341367 Maximum: 1.369858 Average: 0.7593 seconds

Number of cars 1

bbox:car_number ((803, 400), (885, 478)) : 1

The minimum distance from car: 1 is 3.605551275463989

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

98%|| 1230/1261 [54:56<01:23, 2.68s/it]

The times for each task are: [0.964017, 0.708365, 1.602908, 0.404245, 0.360381] with:

Minimum: 0.360381 Maximum: 1.602908 Average: 0.808 seconds

Number of cars 1

bbox:car_number ((805, 400), (877, 477)) : 1

The minimum distance from car: 1 is 3.1622776601683795

```
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

98%|| 1231/1261 [54:59<01:20, 2.68s/it]

The times for each task are: [0.647022, 1.446236, 0.927583, 0.316441, 0.408697] with:

Minimum: 0.316441 Maximum: 1.446236 Average: 0.7492 seconds

```
Number of cars 1
bbox:car_number ((805, 400), (877, 478)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

98%|| 1232/1261 [55:01<01:17, 2.68s/it]

The times for each task are: [0.931147, 0.614965, 1.532549, 0.447557, 0.311295] with:

Minimum: 0.311295 Maximum: 1.532549 Average: 0.7675 seconds

```
Number of cars 1
bbox:car_number ((799, 400), (867, 478)) : 1
The minimum distance from car: 1 is 8.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

98%|| 1233/1261 [55:04<01:15, 2.68s/it]

The times for each task are: [0.880503, 1.36334, 0.698627, 0.282756, 0.388069] with:

Minimum: 0.282756 Maximum: 1.36334 Average: 0.7227 seconds

```
Number of cars 1
bbox:car_number ((790, 400), (867, 478)) : 1
The minimum distance from car: 1 is 5.0
totalCars: 11
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
Length of task list: 5  
Number of processes used: 3
```

98%|| 1234/1261 [55:07<01:12, 2.68s/it]

The times for each task are: [1.433427, 1.027786, 0.645267, 0.399323, 0.337576] with:

Minimum: 0.337576 Maximum: 1.433427 Average: 0.7687 seconds

```
Number of cars 1  
bbox:car_number ((790, 400), (867, 485)) : 1  
The minimum distance from car: 1 is 3.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
Length of task list: 5  
Number of processes used: 3
```

98%|| 1235/1261 [55:09<01:09, 2.68s/it]

The times for each task are: [0.992461, 1.41789, 0.618254, 0.290631, 0.44543] with:

Minimum: 0.290631 Maximum: 1.41789 Average: 0.7529 seconds

```
Number of cars 1  
bbox:car_number ((790, 400), (867, 485)) : 1  
The minimum distance from car: 1 is 0.0  
totalCars: 11  
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]  
Length of task list: 5  
Number of processes used: 3
```

98%|| 1236/1261 [55:12<01:07, 2.68s/it]

The times for each task are: [0.677247, 1.030346, 0.476981, 1.508231, 0.366409] with:

Minimum: 0.366409 Maximum: 1.508231 Average: 0.8118 seconds

```
Number of cars 2  
bbox:car_number ((790, 400), (867, 485)) : 1  
The minimum distance from car: 1 is 0.0  
bbox:car_number ((580, 490), (597, 507)) : 2  
totalCars: 11
```

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

98%|| 1237/1261 [55:15<01:04, 2.68s/it]

The times for each task are: [0.721311, 1.365319, 0.342808, 0.937005, 0.451417] with:

Minimum: 0.342808 Maximum: 1.365319 Average: 0.7636 seconds

Number of cars 1
bbox:car_number ((790, 400), (862, 478)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

98%|| 1238/1261 [55:18<01:01, 2.68s/it]

The times for each task are: [0.632365, 1.57779, 0.97634, 0.374807, 0.434627] with:

Minimum: 0.374807 Maximum: 1.57779 Average: 0.7992 seconds

Number of cars 1
bbox:car_number ((790, 400), (867, 485)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3

98%|| 1239/1261 [55:20<00:58, 2.68s/it]

The times for each task are: [0.630776, 0.900262, 1.545125, 0.30074, 0.436574] with:

Minimum: 0.30074 Maximum: 1.545125 Average: 0.7627 seconds

Number of cars 1
bbox:car_number ((790, 400), (863, 478)) : 1
The minimum distance from car: 1 is 3.605551275463989
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

```
Length of task list: 5
Number of processes used: 3
```

```
98%|| 1240/1261 [55:23<00:56, 2.68s/it]
```

```
The times for each task are: [0.839979, 1.356759, 0.655463, 0.353226, 0.414784] with:
```

```
Minimum: 0.353226 Maximum: 1.356759 Average: 0.724 seconds
```

```
Number of cars 1
bbox:car_number ((790, 400), (863, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
98%|| 1241/1261 [55:26<00:53, 2.68s/it]
```

```
The times for each task are: [0.551936, 1.540841, 1.017617, 0.396436, 0.397736] with:
```

```
Minimum: 0.396436 Maximum: 1.540841 Average: 0.7809 seconds
```

```
Number of cars 1
bbox:car_number ((790, 400), (862, 478)) : 1
The minimum distance from car: 1 is 0.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
98%|| 1242/1261 [55:29<00:50, 2.68s/it]
```

```
The times for each task are: [0.987559, 0.665298, 1.432817, 0.506456, 0.339665] with:
```

```
Minimum: 0.339665 Maximum: 1.432817 Average: 0.7864 seconds
```

```
Number of cars 1
bbox:car_number ((790, 400), (863, 485)) : 1
The minimum distance from car: 1 is 3.0
totalCars: 11
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
```

Number of processes used: 3

99%|| 1243/1261 [55:32<00:48, 2.68s/it]

The times for each task are: [0.866007, 0.671861, 1.579152, 0.343634, 0.459097] with:

Minimum: 0.343634 Maximum: 1.579152 Average: 0.784 seconds

Number of cars 1

bbox:car_number ((780, 400), (863, 478)) : 1

The minimum distance from car: 1 is 5.830951894845301

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

99%|| 1244/1261 [55:34<00:45, 2.68s/it]

The times for each task are: [0.836724, 0.654963, 1.707637, 0.527276, 0.390313] with:

Minimum: 0.390313 Maximum: 1.707637 Average: 0.8234 seconds

Number of cars 1

bbox:car_number ((780, 400), (863, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

99%|| 1245/1261 [55:37<00:42, 2.68s/it]

The times for each task are: [0.57212, 1.43871, 0.89776, 0.300285, 0.511154] with:

Minimum: 0.300285 Maximum: 1.43871 Average: 0.744 seconds

Number of cars 1

bbox:car_number ((780, 400), (862, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

99%|| 1246/1261 [55:39<00:40, 2.68s/it]

The times for each task are: [0.635942, 0.857703, 0.320502, 1.488574, 0.475771] with:

Minimum: 0.320502 Maximum: 1.488574 Average: 0.7557 seconds

Number of cars 1

bbox:car_number ((780, 400), (862, 478)) : 1

The minimum distance from car: 1 is 0.0

totalCars: 11

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

Length of task list: 5

Number of processes used: 3

99%|| 1247/1261 [55:42<00:37, 2.68s/it]

The times for each task are: [0.603627, 0.983346, 1.560914, 0.359824, 0.446462] with:

Minimum: 0.359824 Maximum: 1.560914 Average: 0.7908 seconds

Number of cars 2

bbox:car_number ((780, 400), (862, 485)) : 1

The minimum distance from car: 1 is 3.0

bbox:car_number ((571, 475), (630, 522)) : 2

totalCars: 12

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

totalCars: 12

Car Number: 2 Car Positions: [((571, 475), (630, 522))]

Length of task list: 5

Number of processes used: 3

99%|| 1248/1261 [55:45<00:34, 2.68s/it]

The times for each task are: [0.925637, 0.716578, 1.386176, 0.338293, 0.445195] with:

Minimum: 0.338293 Maximum: 1.386176 Average: 0.7624 seconds

Number of cars 1

bbox:car_number ((778, 400), (862, 485)) : 1

The minimum distance from car: 1 is 1.0

totalCars: 12

Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]

```
Length of task list: 5
Number of processes used: 3
```

```
99%|| 1249/1261 [55:47<00:32, 2.68s/it]
```

```
The times for each task are: [0.922205, 1.39727, 0.679084, 0.410139, 0.340271] with:
```

```
Minimum: 0.340271 Maximum: 1.39727 Average: 0.7498 seconds
```

```
Number of cars 3
bbox:car_number ((780, 400), (862, 478)) : 1
The minimum distance from car: 1 is 3.1622776601683795
bbox:car_number ((1165, 445), (1197, 462)) : 2
bbox:car_number ((1204, 445), (1212, 462)) : 3
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
99%|| 1250/1261 [55:50<00:29, 2.68s/it]
```

```
The times for each task are: [0.937424, 0.733903, 1.564135, 0.400855, 0.336598] with:
```

```
Minimum: 0.336598 Maximum: 1.564135 Average: 0.7946 seconds
```

```
Number of cars 1
bbox:car_number ((778, 400), (862, 478)) : 1
The minimum distance from car: 1 is 1.0
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

```
99%|| 1251/1261 [55:52<00:26, 2.68s/it]
```

```
The times for each task are: [0.600077, 1.467705, 0.949667, 0.489677, 0.301619] with:
```

```
Minimum: 0.301619 Maximum: 1.467705 Average: 0.7617 seconds
```

```
Number of cars 2
bbox:car_number ((799, 400), (858, 472)) : 1
The minimum distance from car: 1 is 8.54400374531753
bbox:car_number ((1217, 430), (1272, 477)) : 2
```

```
The minimum distance from car: 3 is 15.297058540778355
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 12
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

99%|| 1252/1261 [55:55<00:24, 2.68s/it]

```
The times for each task are: [0.952425, 1.324381, 0.684849, 0.299251, 0.448098] with:
Minimum: 0.299251 Maximum: 1.324381 Average: 0.7418 seconds
```

```
Number of cars 1
bbox:car_number ((778, 415), (852, 478)) : 1
The minimum distance from car: 1 is 16.401219466856727
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
Length of task list: 5
Number of processes used: 3
```

99%|| 1253/1261 [55:58<00:21, 2.68s/it]

```
The times for each task are: [0.661409, 0.985005, 1.366776, 0.443883, 0.331998] with:
Minimum: 0.331998 Maximum: 1.366776 Average: 0.7578 seconds
```

```
Number of cars 2
bbox:car_number ((775, 415), (852, 477)) : 1
The minimum distance from car: 1 is 2.0
bbox:car_number ((1183, 419), (1273, 497)) : 2
The minimum distance from car: 3 is 16.76305461424021
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400),
totalCars: 12
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400),
Length of task list: 5
Number of processes used: 3
```

99%|| 1254/1261 [56:00<00:18, 2.68s/it]

```
The times for each task are: [0.620564, 1.413799, 1.012666, 0.418963, 0.391648] with:
```

Minimum: 0.391648 Maximum: 1.413799 Average: 0.7715 seconds

Number of cars 2
bbox:car_number ((775, 415), (852, 478)) : 1
The minimum distance from car: 1 is 0.0
bbox:car_number ((1181, 423), (1268, 498)) : 2
The minimum distance from car: 3 is 4.47213595499958
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 12
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

100%|| 1255/1261 [56:04<00:16, 2.68s/it]

The times for each task are: [0.622636, 1.05251, 0.428304, 1.472544, 0.325762] with:

Minimum: 0.325762 Maximum: 1.472544 Average: 0.7804 seconds

Number of cars 2
bbox:car_number ((775, 400), (852, 477)) : 1
The minimum distance from car: 1 is 8.0
bbox:car_number ((1180, 419), (1253, 492)) : 2
The minimum distance from car: 3 is 9.433981132056603
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 12
Car Number: 3 Car Positions: [((1195, 400), (1268, 477)), ((1175, 400), (1268, 497)), ((1195, 400), (1268, 477))]
Length of task list: 5
Number of processes used: 3

100%|| 1256/1261 [56:06<00:13, 2.68s/it]

The times for each task are: [0.605051, 1.31573, 0.996871, 0.383565, 0.304321] with:

Minimum: 0.304321 Maximum: 1.31573 Average: 0.7211 seconds

Number of cars 2
bbox:car_number ((1158, 400), (1238, 495)) : 1
The minimum distance from car: 3 is 9.219544457292887
bbox:car_number ((772, 415), (852, 478)) : 2
The minimum distance from car: 1 is 8.06225774829855
totalCars: 12
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]

```
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
Length of task list: 5
Number of processes used: 3
```

100%|| 1257/1261 [56:09<00:10, 2.68s/it]

The times for each task are: [0.686009, 0.844169, 0.386138, 1.363957, 0.393652] with:

Minimum: 0.386138 Maximum: 1.363957 Average: 0.7348 seconds

```
Number of cars 2
bbox:car_number ((772, 400), (840, 478)) : 1
The minimum distance from car: 1 is 9.219544457292887
bbox:car_number ((1158, 419), (1230, 485)) : 2
The minimum distance from car: 3 is 6.4031242374328485
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 12
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

100%|| 1258/1261 [56:12<00:08, 2.68s/it]

The times for each task are: [1.017055, 1.320518, 0.680669, 0.40165, 0.40326] with:

Minimum: 0.40165 Maximum: 1.320518 Average: 0.7646 seconds

```
Number of cars 2
bbox:car_number ((780, 400), (840, 477)) : 1
The minimum distance from car: 1 is 4.123105625617661
bbox:car_number ((1158, 415), (1241, 495)) : 2
The minimum distance from car: 3 is 5.830951894845301
totalCars: 12
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
totalCars: 12
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
Length of task list: 5
Number of processes used: 3
```

100%|| 1259/1261 [56:14<00:05, 2.68s/it]

```
The times for each task are: [0.843212, 1.336831, 0.694759, 0.463643, 0.281089] with:
```

```
Minimum: 0.281089 Maximum: 1.336831 Average: 0.7239 seconds
```

```
Number of cars 2
```

```
bbox:car_number ((780, 400), (839, 472)) : 1
```

```
The minimum distance from car: 1 is 2.23606797749979
```

```
bbox:car_number ((1156, 400), (1230, 485)) : 2
```

```
The minimum distance from car: 3 is 14.317821063276353
```

```
totalCars: 12
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 12
```

```
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
```

```
Length of task list: 5
```

```
Number of processes used: 3
```

```
100%|| 1260/1261 [56:17<00:02, 2.68s/it]
```

```
The times for each task are: [0.867814, 1.404609, 0.665583, 0.401478, 0.282405] with:
```

```
Minimum: 0.282405 Maximum: 1.404609 Average: 0.7244 seconds
```

```
Number of cars 3
```

```
bbox:car_number ((778, 400), (839, 478)) : 1
```

```
The minimum distance from car: 1 is 3.1622776601683795
```

```
bbox:car_number ((1136, 400), (1208, 485)) : 2
```

```
The minimum distance from car: 3 is 21.0
```

```
bbox:car_number ((1210, 415), (1230, 472)) : 3
```

```
totalCars: 12
```

```
Car Number: 1 Car Positions: [((835, 400), (963, 512)), ((836, 400), (963, 512)), ((834, 400), (963, 512))]
```

```
totalCars: 12
```

```
Car Number: 3 Car Positions: [((1217, 415), (1272, 472)), ((1217, 400), (1276, 477)), ((1217, 415), (1272, 472))]
```

```
[MoviePy] Done.
```

```
[MoviePy] >>> Video ready: project_videos_output/project_video_after_process_with_vicitniy9.mp4
```

```
CPU times: user 1min 42s, sys: 21min 34s, total: 23min 16s
```

```
Wall time: 56min 18s
```

```
In [170]: project_video_output_no_vicinity = 'project_videos_output/project_video_after_process_no_vicinity'
```

```
In [171]: HTML("""
```

```
<video width="640" height="360" controls>
```

```
<source src="{0}">
</video>
""".format(project_video_output_no_vicinity))
```

```
Out[171]: <IPython.core.display.HTML object>
```

```
In [172]: HTML("""
<video width="640" height="360" controls>
<source src="{0}">
</video>
""".format(project_video_output))
```

```
Out[172]: <IPython.core.display.HTML object>
```