

Vehicle Detection and Tracking Project

Project Details

For this project, a labeled dataset was provided and my job is to decide what features to extract, then train a classifier and ultimately track vehicles in a video stream. Here are links to the labeled data for [vehicle](https://s3.amazonaws.com/udacity-sdc/Vehicle_Tracking/vehicles.zip) (https://s3.amazonaws.com/udacity-sdc/Vehicle_Tracking/vehicles.zip) and [non-vehicle](https://s3.amazonaws.com/udacity-sdc/Vehicle_Tracking/non-vehicles.zip) (https://s3.amazonaws.com/udacity-sdc/Vehicle_Tracking/non-vehicles.zip) examples to train your classifier. These example images come from a combination of the [GTI vehicle image database](http://www.gti.ssr.upm.es/data/Vehicle_database.html) (http://www.gti.ssr.upm.es/data/Vehicle_database.html), the [KITTI vision benchmark suite](http://www.cvlibs.net/datasets/kitti/) (<http://www.cvlibs.net/datasets/kitti/>), and examples extracted from the project video itself.

Udacity recently released a labeled dataset which can be used to take advantage of to augment the training data. The Udacity data can be found [here](https://github.com/udacity/self-driving-car/tree/master/annotations) (<https://github.com/udacity/self-driving-car/tree/master/annotations>). In each of the folders containing images there's a csv file containing all the labels and bounding boxes. To add vehicle images to the training data, we can use the csv files to extract the bounding box regions and scale them to the same size as the rest of the training images.

The project video will be the same one as for the Advanced Lane Finding Project.

The goals / steps of this project are the following:

Pipeline (single images)

- [Step 1](#): Perform a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set of images. Optionally, apply a color transform and append binned color features, as well as histograms of color, to HOG feature vector.
- [Step 2](#): Normalize the features and randomize a selection for training and testing.
- [Step 3](#): Train a classifier using Linear SVM classifier with extracted features.
- [Step 4](#): Implement a sliding-window technique and use trained classifier to search for vehicles in images.
- [Step 5](#): Multiple Detections & False Positives
- [Step 6](#): Estimate a bounding box for vehicles detected.

Video Implementation

Run the pipeline on a video stream and create a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles.

Pipeline (single images)

Import necessary libraries and packages

```
In [1]: import numpy as np
import cv2
import glob
import os
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline
from PIL import Image

from skimage.feature import hog
from scipy.ndimage.measurements import label

import time
from sklearn.svm import LinearSVC, SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

Step 1 Histogram of Oriented Gradients (HOG)

For vehicle detection and tracking, I needed to extract features from training dataset which were used for training process later. I started by reading in all the vehicle and non-vehicle images, exploring the dataset, and displaying an example of one of each of the vehicle and non-vehicle classes as below:

```
In [2]: # Read in car and non-car images
cars = []
images = glob.glob('../..../P5_Data/vehicles/**/*.png', recursive=True)
for image in images:
    cars.append(image)

notcars = []
images = glob.glob('../..../P5_Data/non-vehicles/**/*.png', recursive=True)
for image in images:
    notcars.append(image)

# 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
    example_img = mpimg.imread(car_list[0])
    # Define a key "image_shape" and store the test image shape 3-tuple
    data_dict["image_shape"] = example_img.shape
    # Define a key "data_type" and store the data type of the test image.
    data_dict["data_type"] = example_img.dtype
    # Return data_dict
    return data_dict

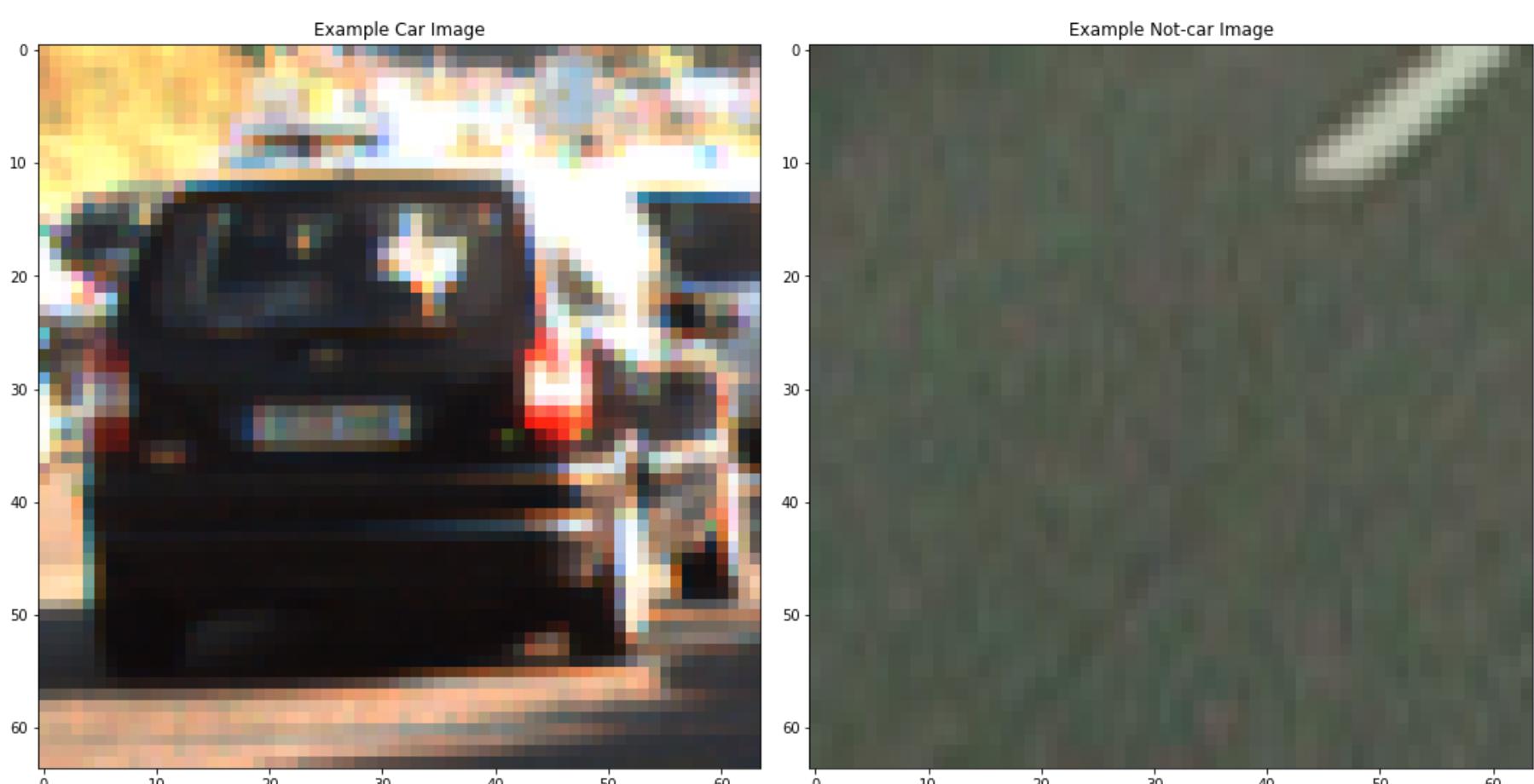
data_info = data_look(cars, notcars)

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"])
# choose random car / not-car indices and plot example images
car_ind = np.random.randint(0, len(cars))
notcar_ind = np.random.randint(0, len(notcars))

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

# Plot the examples
fig = plt.figure(figsize=(15,15))
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')
fig.tight_layout()
```

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



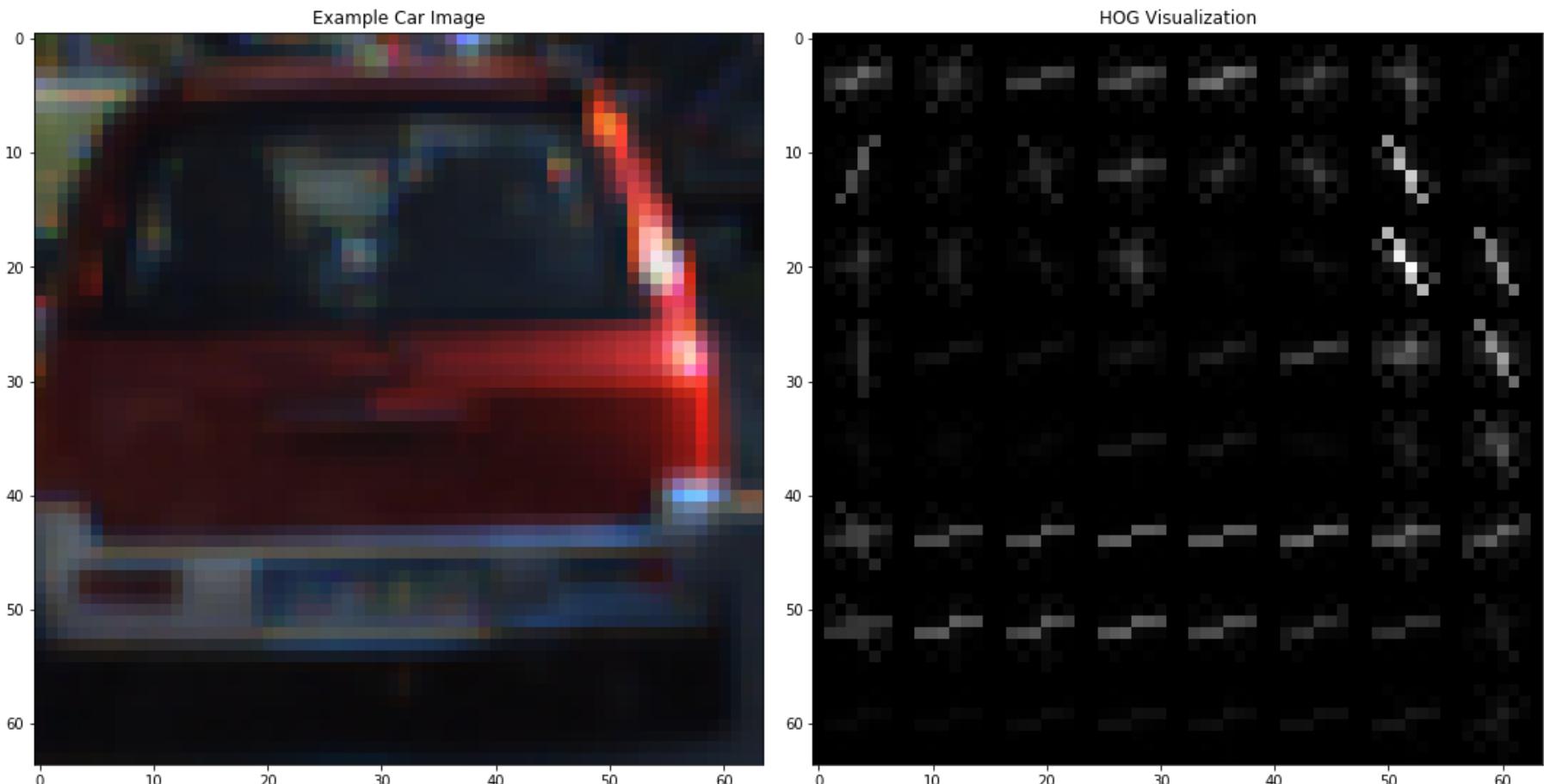
I then explored different color spaces and different `skimage.hog()` (<http://scikit-image.org/docs/stable/api/skimage.feature.html?highlight=hog#skimage.feature.hog>) parameters (orientations, pixels_per_cell, and cells_per_block). I grabbed random images from each of the two classes and displayed them to get a feel for what the `skimage.hog()` output looks like.

```
In [3]: # 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):
    if vis == True:
        features, hog_image = hog(img, orientations=orient, pixels_per_cell=(pix_per_cell, pix_per_cell),
                                  cells_per_block=(cell_per_block, cell_per_block), transform_sqrt=False,
                                  visualise=True, feature_vector=False)
    else:
        features = hog(img, orientations=orient, pixels_per_cell=(pix_per_cell, pix_per_cell),
                      cells_per_block=(cell_per_block, cell_per_block), transform_sqrt=False,
                      visualise=False, feature_vector=feature_vec, block_norm='L2-Hys')
    return features

# Generate a random index to look at a car image
ind = np.random.randint(0, len(cars))
# Read in the image
image = mpimg.imread(cars[ind])
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
# Define HOG parameters
orient = 9
pix_per_cell = 8
cell_per_block = 2
# Call our function with vis=True to see an image output
features, hog_image = get_hog_features(gray, orient, pix_per_cell, cell_per_block, vis=True,
                                         feature_vec=True)

# Plot the examples
fig = plt.figure(figsize=(15,15))
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')
fig.tight_layout()
```

/Users/phuongpham/anaconda3/anaconda/envs/carnd-term1/lib/python3.5/site-packages/skimage/feature/_hog.py:11
9: skimage_deprecation: Default value of `block_norm`=='L1` is deprecated and will be changed to `L2-Hys` in
v0.15
'be changed to `L2-Hys` in v0.15', skimage_deprecation)

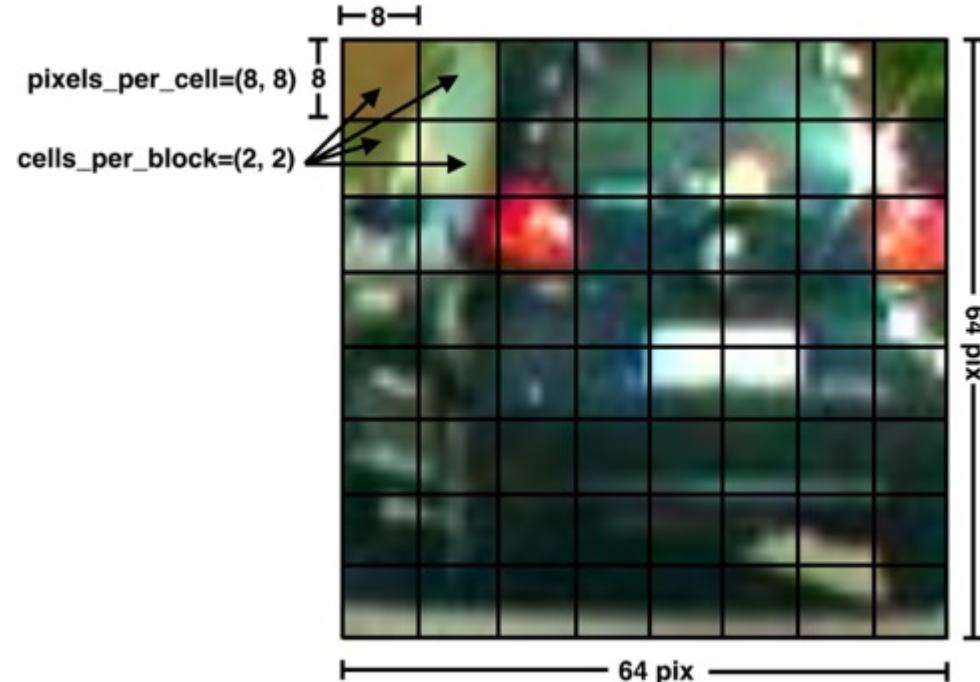


The `skimage.hog()` function takes in a single color channel or grayscaled image as input, as well as various parameters. These parameters include orientations, `pixels_per_cell` and `cells_per_block`.

The value of `orientations` is the number of orientation bins that the gradient information will be split up into in the histogram. Typical values are between 6 and 12 bins.

The `pixels_per_cell` parameter specifies the cell size over which each gradient histogram is computed. This parameter is commonly selected as a 2-tuple so that the cells would be square.

The `cells_per_block` parameter is also passed as a 2-tuple, and specifies the local area over which the histogram counts in a given cell will be normalized. Block normalization is not necessarily required, but generally leads to a more robust feature set.



Source: From Udacity lecture

There is another optional power law or "gamma" normalization scheme set by the flag `transform_sqrt`. This type of normalization may help reduce the effects of shadows or other illumination variation, but will cause an error if the image contains negative values (because it's taking the square root of image values).

The HOG visualization is a representation that shows the dominant gradient direction within each cell with brightness corresponding to the strength of gradients in that cell.

Here is an example using the YCrCb color space and HOG parameters of `orientations=9`, `pixels_per_cell=(8, 8)` and `cells_per_block=(2, 2)`

```
In [4]: # Generate a random index to look at a car image
ind = np.random.randint(0, len(cars))
# Read in the image
image = mpimg.imread(cars[ind])
YCrCb = cv2.cvtColor(image, cv2.COLOR_RGB2YCrCb)
Y_channel = YCrCb[:, :, 0]
Cr_channel = YCrCb[:, :, 1]
Cb_channel = YCrCb[:, :, 2]

# Define HOG parameters
orient = 9
pix_per_cell = 8
cell_per_block = 2
# Call our function with vis=True to see an image output
features_1, hog_image_1 = get_hog_features(Y_channel, orient, pix_per_cell, cell_per_block, vis=True, feature_vec=True)
features_2, hog_image_2 = get_hog_features(Y_channel, orient, pix_per_cell, cell_per_block, vis=True, feature_vec=True)
features_3, hog_image_3 = get_hog_features(Y_channel, orient, pix_per_cell, cell_per_block, vis=True, feature_vec=True)

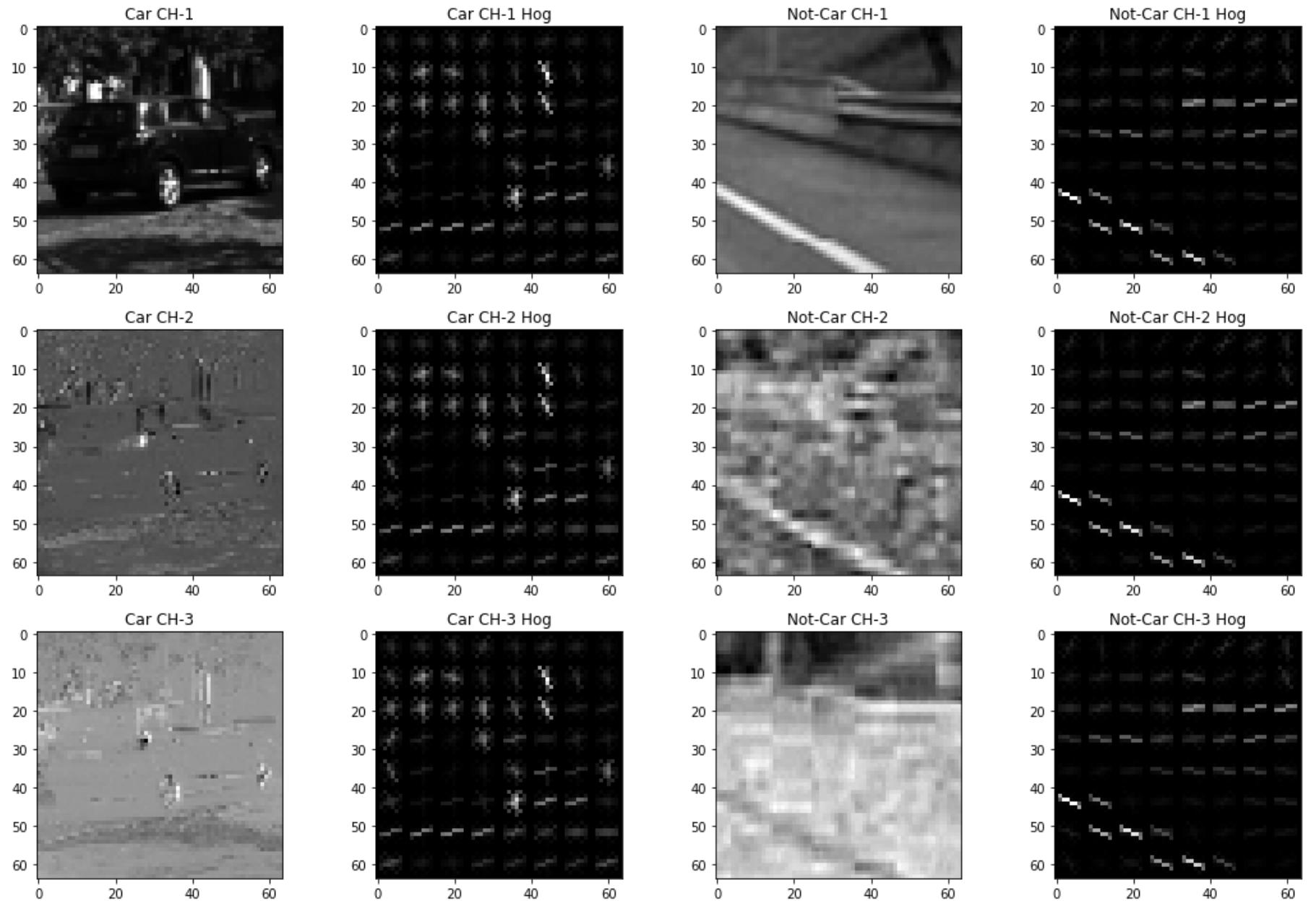
# Plot the examples
fig = plt.figure(figsize=(15,10))
plt.subplot(3,4,1)
plt.imshow(Y_channel, cmap='gray')
plt.title('Car CH-1')
plt.subplot(3,4,2)
plt.imshow(hog_image_1, cmap='gray')
plt.title('Car CH-1 Hog')
plt.subplot(3,4,5)
plt.imshow(Cr_channel, cmap='gray')
plt.title('Car CH-2')
plt.subplot(3,4,6)
plt.imshow(hog_image_2, cmap='gray')
plt.title('Car CH-2 Hog')
plt.subplot(3,4,9)
plt.imshow(Cb_channel, cmap='gray')
plt.title('Car CH-3')
plt.subplot(3,4,10)
plt.imshow(hog_image_3, cmap='gray')
plt.title('Car CH-3 Hog')

image = mpimg.imread(notcars[ind])
YCrCb = cv2.cvtColor(image, cv2.COLOR_RGB2YCrCb)
Y_channel = YCrCb[:, :, 0]
Cr_channel = YCrCb[:, :, 1]
Cb_channel = YCrCb[:, :, 2]

# Call our function with vis=True to see an image output
features_1, hog_image_1 = get_hog_features(Y_channel, orient, pix_per_cell, cell_per_block, vis=True, feature_vec=True)
features_2, hog_image_2 = get_hog_features(Y_channel, orient, pix_per_cell, cell_per_block, vis=True, feature_vec=True)
features_3, hog_image_3 = get_hog_features(Y_channel, orient, pix_per_cell, cell_per_block, vis=True, feature_vec=True)

plt.subplot(3,4,3)
plt.imshow(Y_channel, cmap='gray')
plt.title('Not-Car CH-1')
plt.subplot(3,4,4)
plt.imshow(hog_image_1, cmap='gray')
plt.title('Not-Car CH-1 Hog')
plt.subplot(3,4,7)
plt.imshow(Cr_channel, cmap='gray')
plt.title('Not-Car CH-2')
plt.subplot(3,4,8)
plt.imshow(hog_image_2, cmap='gray')
plt.title('Not-Car CH-2 Hog')
plt.subplot(3,4,11)
plt.imshow(Cb_channel, cmap='gray')
plt.title('Not-Car CH-3')
plt.subplot(3,4,12)
plt.imshow(hog_image_3, cmap='gray')
plt.title('Not-Car CH-3 Hog')
fig.tight_layout()
```

```
/Users/phuongpham/anaconda3/anaconda/envs/carnd-term1/lib/python3.5/site-packages/skimage/feature/_hog.py:11  
9: skimage_deprecation: Default value of `block_norm`=='L1` is deprecated and will be changed to `L2-Hys` in  
v0.15  
'be changed to `L2-Hys` in v0.15', skimage_deprecation)
```



Apply a color transform and append binned color features, as well as histograms of color, to HOG feature vector.

More features such as spatial binned color and histograms of color were extracted and combined to HOG features.

```
In [12]: def convert_color(img, conv='RGB2YCrCb'):
    if conv == 'RGB2YCrCb':
        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)

# 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

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

I tried various combinations of parameters and extracted features for further classification. Below is my final choice of parameters which gave the good results in vehicle detection steps later on.

```
In [6]: ### 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 = (16, 16) # Spatial binning dimensions
hist_bins = 16 # Number of histogram bins
spatial_feat = True # Spatial features on or off
hist_feat = True # Histogram features on or off
hog_feat = True # HOG features on or off
y_start_stop = [None, None] # Min and max in y to search in slide_window()

# Extract features
t=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)
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(round(t2-t, 2), 'Seconds to extract features...')

230.18 Seconds to extract features...
```

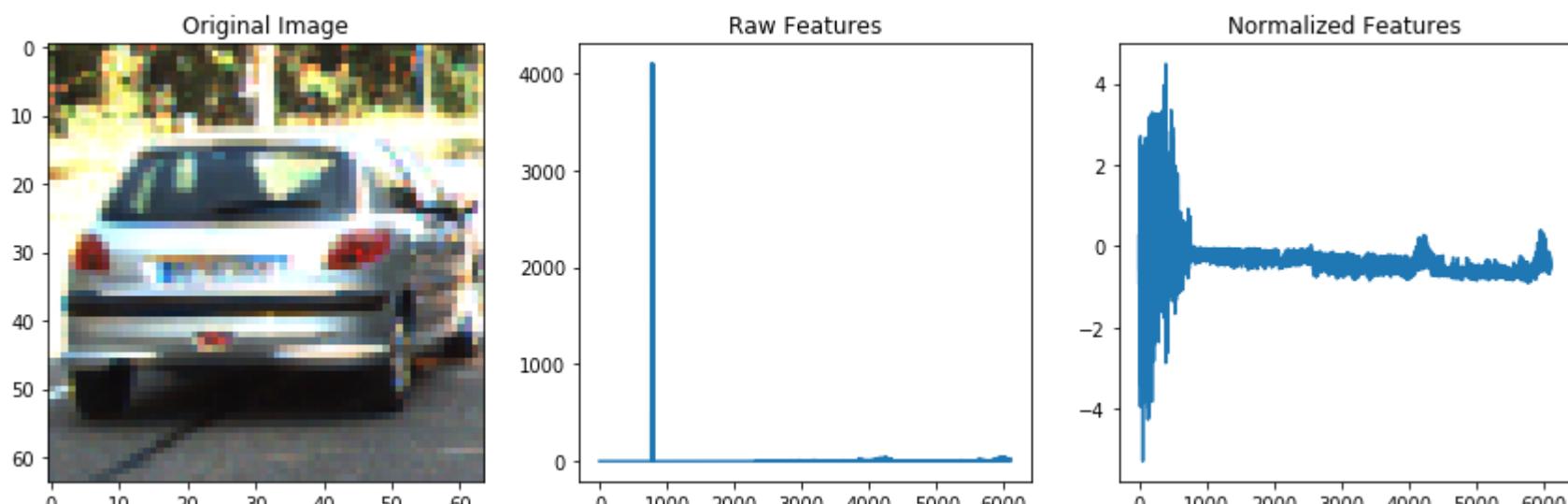
Step 2 Normalize the features and randomize a selection for training and testing.

After gaining features from previous steps, it was necessary to normalize the data before applying any machine learning algorithms. I used `StandardScaler()` (<http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>) method from `sklearn` (<http://scikit-learn.org/stable/index.html>) for normalization.

```
In [7]: 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)

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

    # Plot an example of raw and scaled features
    car_ind = np.random.randint(0, len(cars))
    fig = plt.figure(figsize=(12,4))
    plt.subplot(131)
    plt.imshow(mpimg.imread(cars[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...')
```



The above plots showed the raw features and normalized features extracted from a random car image.

Then the normalized data was randomly splitted into training and test sets as below.

```
In [8]: # 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(
    scaled_X, y, test_size=0.2, random_state=rand_state)

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]))
```

Using: 9 orientations 8 pixels per cell and 2 cells per block
Feature vector length: 6108

Step 3 Train a classifier using Linear SVM classifier with extracted features.

I trained a linear SVM using selected HOG features and color features.

Define a Performance Metric

For the performance_metric function in the code cell below, I used r2_score from `sklearn.metrics` to perform a performance calculation between `y_true` and `y_predict`. Assign the performance score to the `score` variable.

```
In [9]: # TODO: Import 'r2_score'
from sklearn.metrics import r2_score

def performance_metric(y_true, y_predict):
    """ Calculates and returns the performance score between
    true and predicted values based on the metric chosen. """

    # TODO: Calculate the performance score between 'y_true' and 'y_predict'
    score = r2_score(y_true, y_predict)

    # Return the score
    return score
```

Evaluating Model Performance Using Grid Search and Cross-Validation

Hyper-parameters are parameters that are not directly learnt within estimators. In scikit-learn they are passed as arguments to the constructor of the estimator classes. Typical examples include C, kernel and gamma for Support Vector Classifier. It is possible and recommended to search the hyper-parameter space for the best performance score of the learning algorithm. Generally, a search consists of:

- an estimator (regressor or classifier such as `sklearn.svm.SVC()`);
- a parameter space;
- a method for searching or sampling candidates;
- a cross-validation scheme; and
- a score function.

A generic approach to sampling search candidates (provided in scikit-learn) is grid search, considers all parameter combinations. In scikit-learn, the grid search provided by `GridSearchCV` exhaustively generates candidates from a grid of parameter values specified with the `param_grid` parameter. The `GridSearchCV` instance then are input to construct the models. The fitting models of all the possible combinations of parameter values are evaluated and the best combination is retained.

```
In [11]: from sklearn.metrics import make_scorer
from sklearn.model_selection import GridSearchCV
from sklearn.cross_validation import ShuffleSplit

def fit_model(X, y):
    """ Performs grid search over the 'C' parameter for a
    SVM trained on the input data [X, y]. """

    # Create cross-validation sets from the training data
    # sklearn version 0.18: ShuffleSplit(n_splits=10, test_size=None, random_state=None)
    # sklearn version 0.17: ShuffleSplit(n, n_iter=10, test_size=0.1, train_size=None, random_state=None)
    random_state = np.random.randint(0, 100)
    cv_sets = ShuffleSplit(X.shape[0], n_iter=5, test_size=0.20, random_state=random_state)

    # TODO: Create a decision tree regressor object
    svc = SVC(random_state=random_state)

    # TODO: Create a dictionary for the parameter 'max_depth' with a range from 1 to 10
    params = {'C': [1, 10], 'kernel': ['rbf', 'linear']}

    # TODO: Transform 'performance_metric' into a scoring function using 'make_scorer'
    scoring_fnc = make_scorer(performance_metric)

    # TODO: Create the grid search object
    grid = GridSearchCV(svc, params, scoring=scoring_fnc, cv=cv_sets)

    # Fit the grid search object to the data to compute the optimal model
    grid = grid.fit(X, y)

    # Return the optimal model after fitting the data
    return grid.best_estimator_

# Fit the training data to the model using grid search
result = fit_model(X_train, y_train)
print(result)
```

SVC(C=10, cache_size=200, class_weight=None, coef0=0.0,
 decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
 max_iter=-1, probability=False, random_state=25, shrinking=True,
 tol=0.001, verbose=False)

The best result after grid search(C=10, kernel='rbf') were selected for training and testing.

```
In [14]: # Use a SVC
random_state = np.random.randint(0, 100)
svc = SVC(C=10, kernel='rbf', random_state=random_state)
# 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')

161.2 Seconds to train SVC...
Test Accuracy of SVC =  0.9938
My SVC predicts: [ 0.  0.  0.  0.  0.  1.  1.  0.  1.]
For these 10 labels: [ 0.  0.  0.  0.  0.  1.  1.  0.  1.]
0.15954 Seconds to predict 10 labels with SVC
```

Step 4 Sliding Window Search

I started with sliding window search which searched over all the windows defined by `slide_windows()` with window size was (96, 96) pixels and overlap 50% of the windows, extracted features at each window position, and predicted with the classifier on each set of features.

```
In [17]: # Define a function to extract features from a single image window
# This function is very similar to extract_features()
# just for a single image rather than list of images
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
```

```



```

```

#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]], (64, 64))
    #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

# 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 [18]: images = glob.glob('./test_images/*.jpg')

for fname in images:
    image = mpimg.imread(fname)

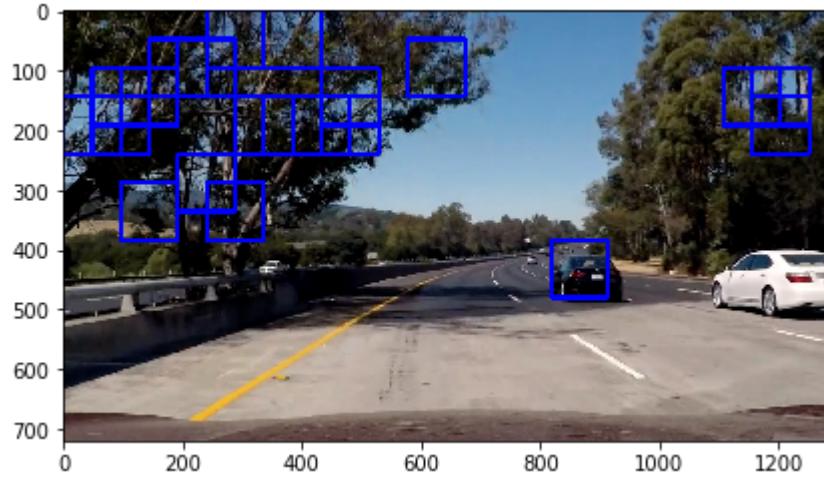
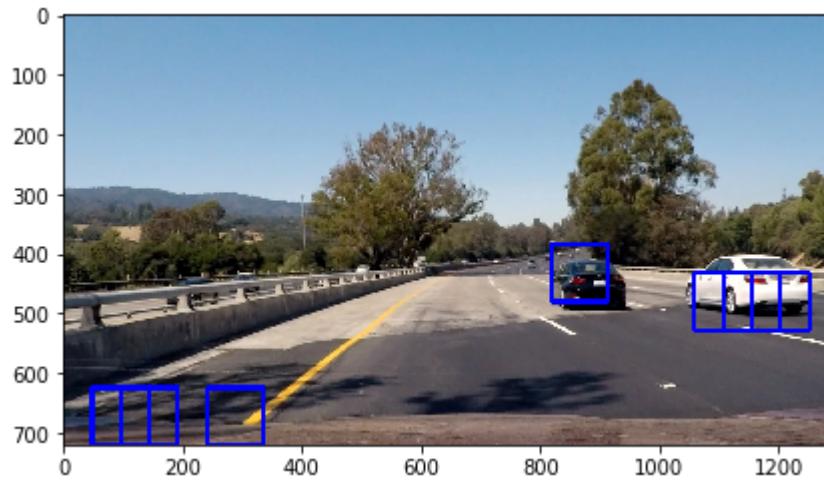
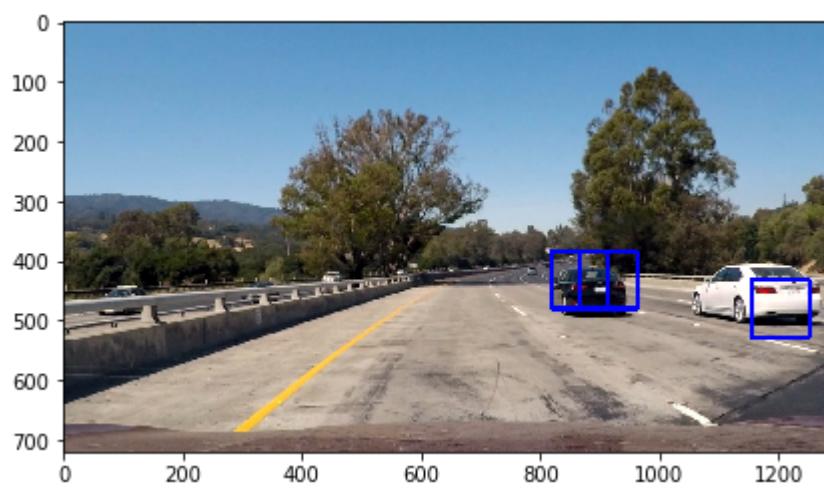
    draw_image = np.copy(image)
    # Uncomment the following line if you extracted training
    # data from .png images (scaled 0 to 1 by mpimg) and the
    # image you are searching is a .jpg (scaled 0 to 255)
    image = image.astype(np.float32)/255

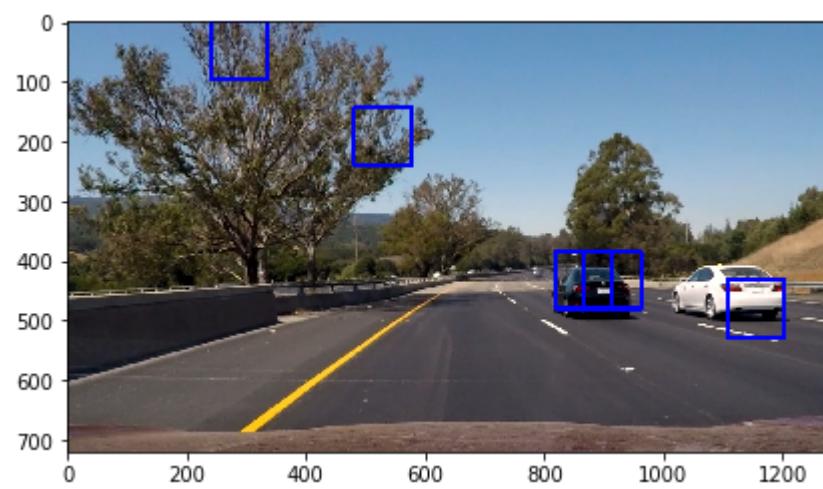
    windows = slide_window(image, x_start_stop=[None, None], y_start_stop=y_start_stop,
                           xy_window=(96, 96), xy_overlap=(0.5, 0.5))

    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)

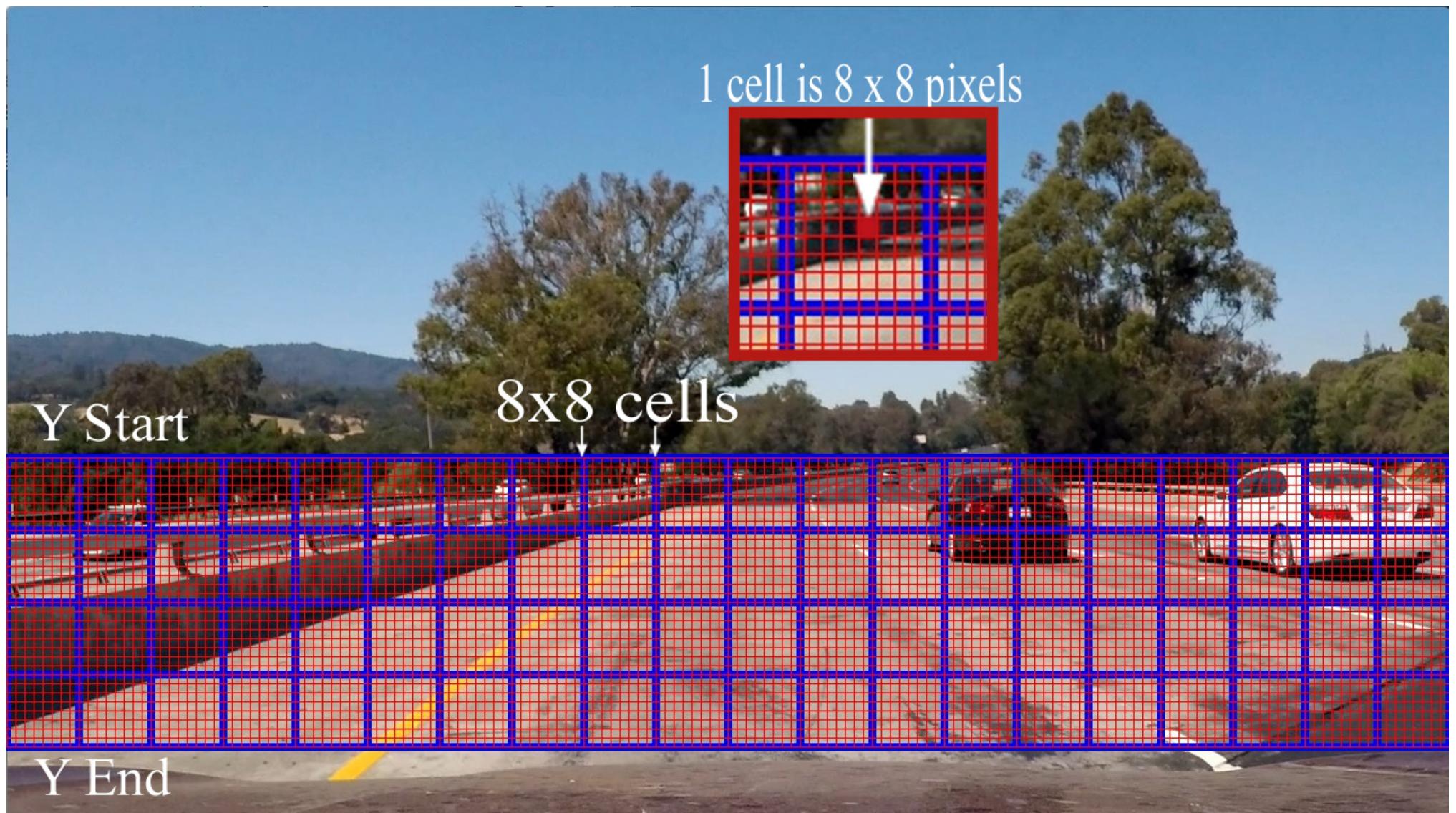
    window_img = draw_boxes(draw_image, hot_windows, color=(0, 0, 255), thick=6)

    fig = plt.figure()
    plt.imshow(window_img)
    fig.tight_layout()
```





The sliding window approach detected cars in the images but still found some false positives in the skies and treetops. So I used Hog Sub-Sampling Window Search implemented with `find_cars()` function to extract hog features once and then sub-sampled to get all of its overlaying windows. Each window is defined by a scaling factor where a scale of 1 would result in a window that's 8×8 cells then the overlap of each window is in terms of the cell distance. This means that a `cells_per_step = 2` would result in a search window overlap of 75%. To remove the false positives in the skies and treetops, I limited the search area from `y_start` to `y_stop`. I ran this `find_cars()` function multiple times for different scale values to generate multiple-scaled search windows.



Source: From Udacity lecture

Ultimately I searched on two scales using YCrCb 3-channel HOG features plus spatially binned color and histograms of color in the feature vector.

```
In [20]: # Define a single function that can extract features using hog sub-sampling and make predictions
def find_cars(img, ystart, ystop, scale, svc, X_scaler, orient, pix_per_cell, cell_per_block, spatial_size, hist_bins):

    draw_img = np.copy(img)
    img = img.astype(np.float32)/255

    img_tosearch = img[ystart:ystop,:,:]
    ctrans_tosearch = convert_color(img_tosearch, conv='RGB2YCrCb')
    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]

    # 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

    # 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
```

```

nysteps = (nyblocks - nblocks_per_window) // cells_per_step

# 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)

bboxes = []

for xb in range(nxsteps):
    for yb in range(nysteps):
        ypos = yb*cells_per_step
        xpos = xb*cells_per_step
        # Extract HOG for this patch
        hog_feat1 = hog1[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window].ravel()
        hog_feat2 = hog2[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window].ravel()
        hog_feat3 = hog3[ypos:ypos+nblocks_per_window, xpos:xpos+nblocks_per_window].ravel()
        hog_features = np.hstack((hog_feat1, hog_feat2, hog_feat3))

        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], (64,64))

        # 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, hog_features)).reshape(1, -1))
        #test_features = X_scaler.transform(np.hstack((shape_feat, hist_feat)).reshape(1, -1))
        test_prediction = svc.predict(test_features)

        if test_prediction == 1:
            xbox_left = np.int(xleft*scale)
            ytop_draw = np.int(ytop*scale)
            win_draw = np.int(window*scale)
            cv2.rectangle(draw_img,(xbox_left, ytop_draw+ystart),(xbox_left+win_draw,ytop_draw+win_draw+ystart),(0,0,255),6)
            bbox = ((xbox_left, ytop_draw+ystart),(xbox_left+win_draw,ytop_draw+win_draw+ystart))
            bboxes.append(bbox)
    return draw_img, bboxes

y_start = 400
y_stop = 656
scales = [1, 1.5, 2, 2.5, 3]

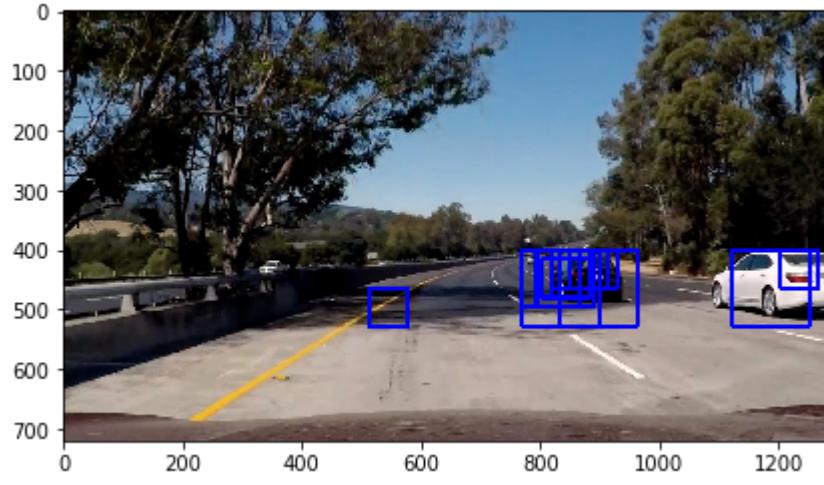
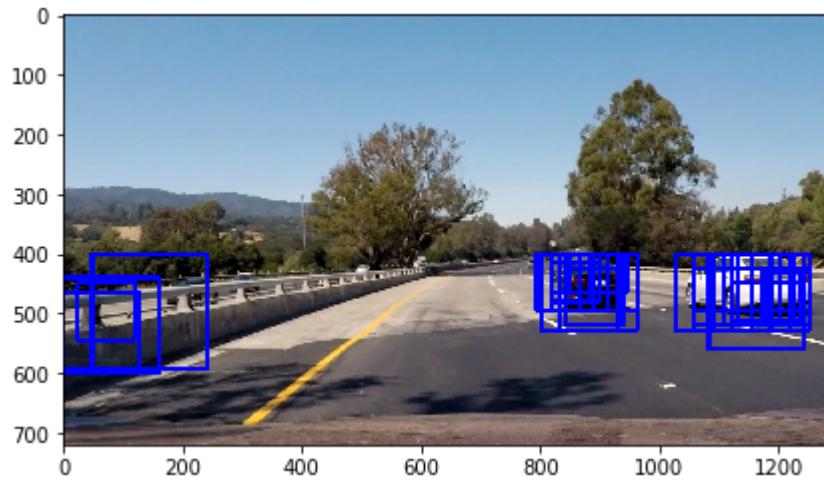
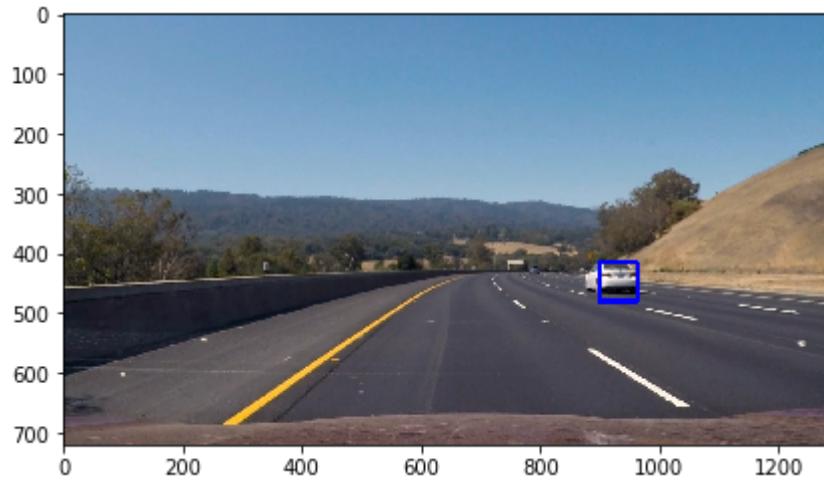
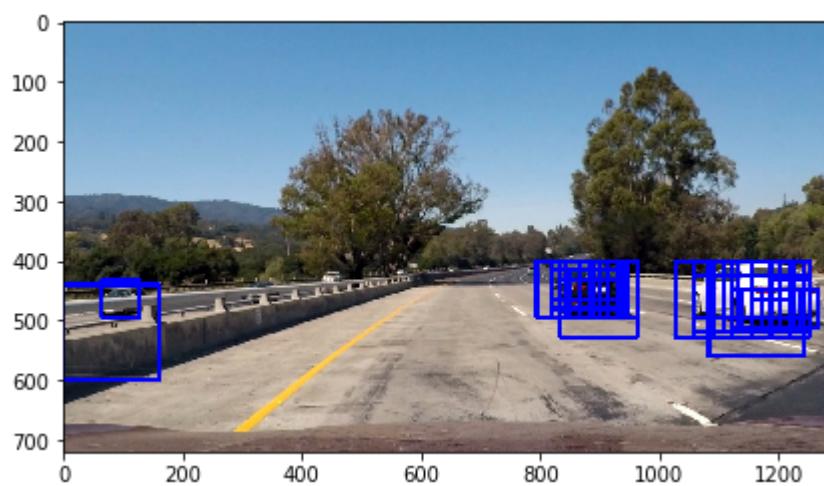
images = glob.glob('./test_images/*.jpg')

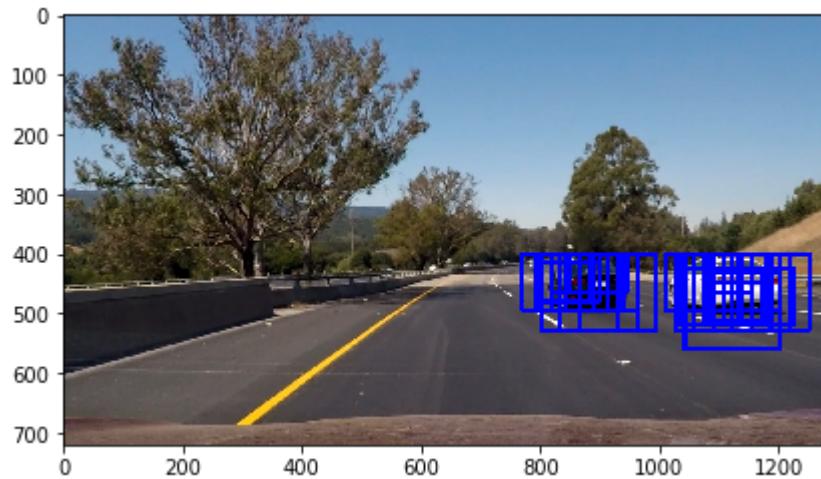
for fname in images:
    image = mpimg.imread(fname)

    box_list = []
    for scale in scales:
        out_img, boxes = find_cars(image, y_start, y_stop, scale, svc, X_scaler, orient, pix_per_cell, cell_per_block, spatial_size, hist_bins)
        box_list = box_list + boxes
    draw_image = np.copy(image)
    out_img = draw_boxes(draw_image, box_list, color=(0, 0, 255), thick=6)

    fig = plt.figure()
    plt.imshow(out_img)
    fig.tight_layout()

```





Step 5 Multiple Detections & False Positives

The Window Search methods and predictions returned positive detections for each images. From the positive detections I created a heatmap by adding "heat" ($+1$) to a map for a list of bounding boxes and then thresholded that map to identify vehicle positions and remove false positives.

I then used `scipy.ndimage.measurements.label()` to identify individual blobs in the heatmap. I then assumed each blob corresponded to a vehicle.

```
In [22]: 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))
        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
```

Step 6 Estimate a bounding box for vehicles detected.

I constructed bounding boxes to cover the area of each blob detected.

```
In [23]: def draw_labeled_bboxes(img, labels):
    # Iterate through all detected cars
    for car_number in range(1, labels[1]+1):
        # Find pixels with each car_number label value
        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)))
        # Draw the box on the image
        cv2.rectangle(img, bbox[0], bbox[1], (0,0,255), 6)
    # Return the image
    return img
```

Here's an example result showing the heatmap from testing images, the result of `scipy.ndimage.measurements.label()` and the bounding boxes:

```
In [24]: y_start = 400
y_stop = 656
scales = [1, 1.5, 2, 2.5, 3]

images = glob.glob('./test_images/*.jpg')

for fname in images:
    image = mpimg.imread(fname)
    heat = np.zeros_like(image[:, :, 0]).astype(np.float)

    box_list = []
    for scale in scales:
        out_img, boxes = find_cars(image, y_start, y_stop, scale, svc, x_scaler, orient, pix_per_cell, cell_per_block, spatial_size, hist_bins)
        box_list = box_list + boxes

    positions_img = draw_boxes(np.copy(image), box_list, color=(0, 0, 255), thick=6)

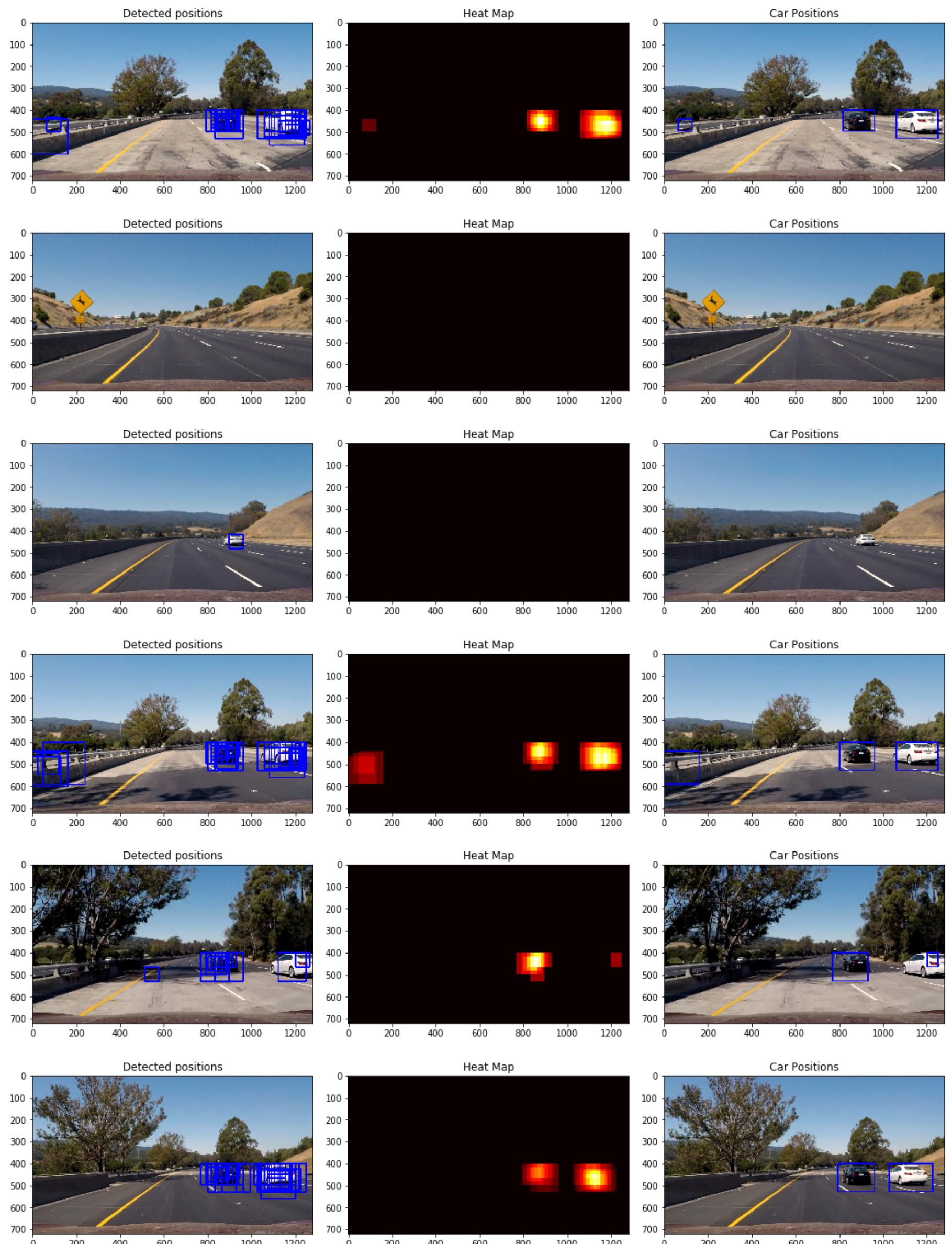
    # Add heat to each box in box list
    heat = add_heat(heat, box_list)

    # Apply threshold to help remove false positives
    heat = apply_threshold(heat, 1)

    # 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(image), labels)

    fig = plt.figure(figsize=(15, 15))
    plt.subplot(131)
    plt.imshow(positions_img)
    plt.title('Detected positions')
    plt.subplot(132)
    plt.imshow(heatmap, cmap='hot')
    plt.title('Heat Map')
    plt.subplot(133)
    plt.imshow(draw_img)
    plt.title('Car Positions')
    fig.tight_layout()
```



Video Implementation

For a video stream, I run the pipeline and integrate a heat map over several frames of video, such that areas of multiple detections get "hot", while transient false positives stay "cool". I can then simply threshold my heatmap to remove false positives.

```
In [25]: # Import everything needed to edit/save/watch video clips
from moviepy.editor import VideoFileClip
from IPython.display import HTML
```

```
In [26]: import imageio  
imageio.plugins.ffmpeg.download()
```

```
In [66]: def process_frame(img):  
    heat = np.zeros_like(img[:, :, 0]).astype(np.float)  
  
    # Find bounding boxes of current frame  
    box_list = []  
    for scale in scales:  
        out_img, boxes = find_cars(img, y_start, y_stop, scale, svc, x_scaler, orient, pix_per_cell, cell_per  
        _block, spatial_size, hist_bins)  
        box_list = box_list + boxes  
  
    # Add bounding boxes of current frame to buffer  
    global box_buffer  
    box_buffer.append(box_list)  
  
    # Get only last nb_frames frames to build heatmap  
    tmp_box_list = box_buffer[-nb_frames:]  
    new_box_list = []  
    for tmp in tmp_box_list:  
        new_box_list = new_box_list + tmp  
  
    # Add heat to each box in box list  
    heat = add_heat(heat, new_box_list)  
  
    # Apply threshold to help remove false positives  
    heat = apply_threshold(heat, heat_threshold)  
  
    # Visualize the heatmap when displaying  
    heatmap = np.clip(heat, 0, 255)  
    #avg_heatmap= new_frame_factor * heatmap + (1-new_frame_factor) * avg_heatmap  
  
    # Find final boxes from heatmap using label function  
    labels = label(heatmap)  
    draw_img = draw_labeled_bboxes(np.copy(img), labels)  
  
    return draw_img
```

Below are the video results with different scales settings. The former is set with only one scale = 1.5. The latter is run multi-scale at 1 and 1.5. The multi-scaled video gave better detection and tracking result. Still, there is room for improvements by twisting parameters such as threshold of the heatmap, number of frames for integrating the heatmap.

```
In [70]: ystart = 400  
ystop = 656  
scales = [1.5]  
box_buffer = []  
heat_threshold = 5  
nb_frames = 5
```

```
In [71]: ## To speed up the testing process you may want to try your pipeline on a shorter subclip of the video
## 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 subclip
## You may also uncomment the following line for a subclip of the first 5 seconds
##clip1 = VideoFileClip("test_videos/solidWhiteRight.mp4").subclip(0,5)
in_vid = './project_video.mp4'
out_vid = './project_video_out_3.mp4'

clip = VideoFileClip(in_vid)
video_clip = clip.fl_image(process_frame)
%time video_clip.write_videofile(out_vid, audio=False)
```

```
[MoviePy] >>> Building video ./project_video_out_3.mp4
[MoviePy] Writing video ./project_video_out_3.mp4
```

0%	0/1261 [00:00<?, ?it/s]
0%	1/1261 [00:03<1:22:16, 3.92s/it]
0%	2/1261 [00:07<1:22:03, 3.91s/it]
0%	3/1261 [00:11<1:23:15, 3.97s/it]
0%	4/1261 [00:15<1:23:06, 3.97s/it]
0%	5/1261 [00:19<1:22:47, 3.95s/it]
0%	6/1261 [00:23<1:23:45, 4.00s/it]
1%	7/1261 [00:28<1:26:28, 4.14s/it]
1%	8/1261 [00:32<1:28:57, 4.26s/it]
1%	9/1261 [00:37<1:30:51, 4.35s/it]
1%	10/1261 [00:41<1:30:07, 4.32s/it]
1%	11/1261 [00:45<1:28:17, 4.24s/it]
1%	12/1261 [00:49<1:28:01, 4.23s/it]
1%	13/1261 [00:54<1:28:56, 4.28s/it]
1%	14/1261 [00:58<1:30:31, 4.36s/it]
1%	15/1261 [01:03<1:31:52, 4.42s/it]
1%	16/1261 [01:08<1:34:49, 4.57s/it]
1%	17/1261 [01:12<1:32:34, 4.47s/it]
1%	18/1261 [01:17<1:33:04, 4.49s/it]
2%	19/1261 [01:21<1:32:38, 4.48s/it]
2%	20/1261 [01:26<1:32:06, 4.45s/it]
2%	21/1261 [01:30<1:35:01, 4.60s/it]
2%	22/1261 [01:35<1:33:49, 4.54s/it]
2%	23/1261 [01:39<1:31:18, 4.43s/it]
2%	24/1261 [01:43<1:28:25, 4.29s/it]
2%	25/1261 [01:47<1:26:23, 4.19s/it]
2%	26/1261 [01:51<1:26:18, 4.19s/it]
2%	27/1261 [01:55<1:24:03, 4.09s/it]
2%	28/1261 [01:59<1:23:42, 4.07s/it]
2%	29/1261 [02:04<1:27:05, 4.24s/it]
2%	30/1261 [02:08<1:29:38, 4.37s/it]
2%	31/1261 [02:12<1:27:31, 4.27s/it]
3%	32/1261 [02:16<1:26:01, 4.20s/it]
3%	33/1261 [02:21<1:25:20, 4.17s/it]
3%	34/1261 [02:25<1:24:27, 4.13s/it]
3%	35/1261 [02:29<1:24:03, 4.11s/it]
3%	36/1261 [02:33<1:23:58, 4.11s/it]
3%	37/1261 [02:37<1:23:12, 4.08s/it]
3%	38/1261 [02:41<1:23:23, 4.09s/it]
3%	39/1261 [02:45<1:23:11, 4.08s/it]
3%	40/1261 [02:49<1:23:23, 4.10s/it]
3%	41/1261 [02:53<1:23:06, 4.09s/it]
3%	42/1261 [02:57<1:22:59, 4.08s/it]
3%	43/1261 [03:02<1:24:23, 4.16s/it]
3%	44/1261 [03:06<1:24:51, 4.18s/it]
4%	45/1261 [03:10<1:24:06, 4.15s/it]
4%	46/1261 [03:14<1:23:35, 4.13s/it]
4%	47/1261 [03:18<1:22:37, 4.08s/it]
4%	48/1261 [03:22<1:22:23, 4.08s/it]
4%	49/1261 [03:26<1:22:58, 4.11s/it]
4%	50/1261 [03:30<1:22:53, 4.11s/it]
4%	51/1261 [03:34<1:22:28, 4.09s/it]
4%	52/1261 [03:38<1:22:37, 4.10s/it]
4%	53/1261 [03:42<1:22:08, 4.08s/it]
4%	54/1261 [03:47<1:22:20, 4.09s/it]
4%	55/1261 [03:51<1:21:50, 4.07s/it]
4%	56/1261 [03:55<1:21:06, 4.04s/it]
5%	57/1261 [03:59<1:20:59, 4.04s/it]
5%	58/1261 [04:03<1:20:34, 4.02s/it]
5%	59/1261 [04:07<1:20:26, 4.02s/it]
5%	60/1261 [04:11<1:20:18, 4.01s/it]
5%	61/1261 [04:15<1:20:55, 4.05s/it]
5%	62/1261 [04:19<1:21:15, 4.07s/it]
5%	63/1261 [04:23<1:21:26, 4.08s/it]
5%	64/1261 [04:27<1:21:09, 4.07s/it]
5%	65/1261 [04:31<1:22:07, 4.12s/it]
5%	66/1261 [04:35<1:21:52, 4.11s/it]
5%	67/1261 [04:40<1:22:44, 4.16s/it]
5%	68/1261 [04:44<1:21:37, 4.11s/it]
5%	69/1261 [04:48<1:22:57, 4.18s/it]
6%	70/1261 [04:52<1:21:43, 4.12s/it]
6%	71/1261 [04:57<1:24:44, 4.27s/it]
6%	72/1261 [05:01<1:26:01, 4.34s/it]
6%	73/1261 [05:05<1:26:32, 4.37s/it]
6%	74/1261 [05:10<1:28:55, 4.50s/it]
6%	75/1261 [05:15<1:31:14, 4.62s/it]
6%	76/1261 [05:20<1:31:55, 4.65s/it]
6%	77/1261 [05:24<1:30:41, 4.60s/it]
6%	78/1261 [05:29<1:29:04, 4.52s/it]
6%	79/1261 [05:33<1:28:07, 4.47s/it]
6%	80/1261 [05:37<1:26:06, 4.38s/it]
6%	81/1261 [05:41<1:24:34, 4.30s/it]
7%	82/1261 [05:45<1:23:12, 4.23s/it]
7%	83/1261 [05:49<1:22:18, 4.19s/it]
7%	84/1261 [05:54<1:22:15, 4.19s/it]
7%	85/1261 [05:59<1:27:36, 4.47s/it]
7%	86/1261 [06:03<1:28:41, 4.53s/it]
7%	87/1261 [06:07<1:24:48, 4.33s/it]

7% [88/1261 [06:12<1:27:34, 4.48s/it]
7% [89/1261 [06:16<1:26:22, 4.42s/it]
7% [90/1261 [06:21<1:28:35, 4.54s/it]
7% [91/1261 [06:25<1:26:31, 4.44s/it]
7% [92/1261 [06:30<1:28:09, 4.52s/it]
7% [93/1261 [06:35<1:30:35, 4.65s/it]
7% [94/1261 [06:40<1:31:24, 4.70s/it]
8% [95/1261 [06:45<1:30:48, 4.67s/it]
8% [96/1261 [06:49<1:29:09, 4.59s/it]
8% [97/1261 [06:53<1:26:59, 4.48s/it]
8% [98/1261 [06:58<1:26:25, 4.46s/it]
8% [99/1261 [07:02<1:24:28, 4.36s/it]
8% [100/1261 [07:06<1:23:44, 4.33s/it]
8% [101/1261 [07:10<1:21:24, 4.21s/it]
8% [102/1261 [07:14<1:20:20, 4.16s/it]
8% [103/1261 [07:18<1:19:36, 4.12s/it]
8% [104/1261 [07:22<1:20:22, 4.17s/it]
8% [105/1261 [07:27<1:22:13, 4.27s/it]
8% [106/1261 [07:32<1:26:20, 4.49s/it]
8% [107/1261 [07:37<1:28:23, 4.60s/it]
9% [108/1261 [07:41<1:25:47, 4.46s/it]
9% [109/1261 [07:45<1:23:39, 4.36s/it]
9% [110/1261 [07:49<1:21:53, 4.27s/it]
9% [111/1261 [07:53<1:20:03, 4.18s/it]
9% [112/1261 [07:57<1:17:52, 4.07s/it]
9% [113/1261 [08:01<1:16:58, 4.02s/it]
9% [114/1261 [08:05<1:16:38, 4.01s/it]
9% [115/1261 [08:09<1:16:46, 4.02s/it]
9% [116/1261 [08:13<1:16:57, 4.03s/it]
9% [117/1261 [08:17<1:16:29, 4.01s/it]
9% [118/1261 [08:21<1:15:40, 3.97s/it]
9% [119/1261 [08:24<1:15:03, 3.94s/it]
10% [120/1261 [08:28<1:14:38, 3.93s/it]
10% [121/1261 [08:32<1:14:23, 3.92s/it]
10% [122/1261 [08:36<1:14:05, 3.90s/it]
10% [123/1261 [08:40<1:13:53, 3.90s/it]
10% [124/1261 [08:44<1:13:36, 3.88s/it]
10% [125/1261 [08:48<1:13:54, 3.90s/it]
10% [126/1261 [08:52<1:14:54, 3.96s/it]
10% [127/1261 [08:56<1:15:26, 3.99s/it]
10% [128/1261 [09:00<1:16:33, 4.05s/it]
10% [129/1261 [09:05<1:20:25, 4.26s/it]
10% [130/1261 [09:09<1:21:03, 4.30s/it]
10% [131/1261 [09:13<1:19:25, 4.22s/it]
10% [132/1261 [09:17<1:18:55, 4.19s/it]
11% [133/1261 [09:22<1:19:30, 4.23s/it]
11% [134/1261 [09:26<1:19:00, 4.21s/it]
11% [135/1261 [09:30<1:17:23, 4.12s/it]
11% [136/1261 [09:34<1:17:33, 4.14s/it]
11% [137/1261 [09:38<1:18:08, 4.17s/it]
11% [138/1261 [09:43<1:21:14, 4.34s/it]
11% [139/1261 [09:47<1:21:23, 4.35s/it]
11% [140/1261 [09:52<1:20:54, 4.33s/it]
11% [141/1261 [09:56<1:19:26, 4.26s/it]
11% [142/1261 [10:00<1:19:31, 4.26s/it]
11% [143/1261 [10:04<1:17:47, 4.17s/it]
11% [144/1261 [10:08<1:18:21, 4.21s/it]
11% [145/1261 [10:13<1:21:33, 4.39s/it]
12% [146/1261 [10:18<1:23:29, 4.49s/it]
12% [147/1261 [10:22<1:21:58, 4.41s/it]
12% [148/1261 [10:26<1:20:18, 4.33s/it]
12% [149/1261 [10:30<1:18:14, 4.22s/it]
12% [150/1261 [10:34<1:17:22, 4.18s/it]
12% [151/1261 [10:39<1:19:38, 4.31s/it]
12% [152/1261 [10:44<1:24:07, 4.55s/it]
12% [153/1261 [10:49<1:27:40, 4.75s/it]
12% [154/1261 [10:54<1:30:41, 4.92s/it]
12% [155/1261 [11:00<1:34:17, 5.12s/it]
12% [156/1261 [11:06<1:36:14, 5.23s/it]
12% [157/1261 [11:10<1:32:28, 5.03s/it]
13% [158/1261 [11:15<1:30:54, 4.95s/it]
13% [159/1261 [11:19<1:28:55, 4.84s/it]
13% [160/1261 [11:24<1:28:59, 4.85s/it]
13% [161/1261 [11:29<1:26:37, 4.73s/it]
13% [162/1261 [11:33<1:26:34, 4.73s/it]
13% [163/1261 [11:38<1:26:06, 4.71s/it]
13% [164/1261 [11:43<1:25:42, 4.69s/it]
13% [165/1261 [11:47<1:24:45, 4.64s/it]
13% [166/1261 [11:52<1:22:23, 4.51s/it]
13% [167/1261 [11:56<1:22:29, 4.52s/it]
13% [168/1261 [12:01<1:22:17, 4.52s/it]
13% [169/1261 [12:05<1:21:06, 4.46s/it]
13% [170/1261 [12:09<1:20:19, 4.42s/it]
14% [171/1261 [12:13<1:18:47, 4.34s/it]
14% [172/1261 [12:17<1:17:17, 4.26s/it]
14% [173/1261 [12:22<1:16:23, 4.21s/it]
14% [174/1261 [12:26<1:17:12, 4.26s/it]
14% [175/1261 [12:30<1:16:26, 4.22s/it]
14% [176/1261 [12:34<1:17:32, 4.29s/it]

14%	177/1261	[12:39<1:16:32,	4.24s/it]
14%	178/1261	[12:43<1:16:12,	4.22s/it]
14%	179/1261	[12:47<1:14:35,	4.14s/it]
14%	180/1261	[12:51<1:15:19,	4.18s/it]
14%	181/1261	[12:55<1:13:59,	4.11s/it]
14%	182/1261	[12:59<1:13:46,	4.10s/it]
15%	183/1261	[13:04<1:16:10,	4.24s/it]
15%	184/1261	[13:08<1:14:50,	4.17s/it]
15%	185/1261	[13:12<1:14:39,	4.16s/it]
15%	186/1261	[13:16<1:14:30,	4.16s/it]
15%	187/1261	[13:21<1:17:07,	4.31s/it]
15%	188/1261	[13:25<1:20:04,	4.48s/it]
15%	189/1261	[13:30<1:19:23,	4.44s/it]
15%	190/1261	[13:34<1:17:13,	4.33s/it]
15%	191/1261	[13:38<1:14:35,	4.18s/it]
15%	192/1261	[13:42<1:12:37,	4.08s/it]
15%	193/1261	[13:45<1:11:23,	4.01s/it]
15%	194/1261	[13:49<1:10:23,	3.96s/it]
15%	195/1261	[13:53<1:09:53,	3.93s/it]
16%	196/1261	[13:57<1:09:17,	3.90s/it]
16%	197/1261	[14:01<1:08:48,	3.88s/it]
16%	198/1261	[14:05<1:08:30,	3.87s/it]
16%	199/1261	[14:08<1:08:26,	3.87s/it]
16%	200/1261	[14:12<1:08:42,	3.89s/it]
16%	201/1261	[14:16<1:08:31,	3.88s/it]
16%	202/1261	[14:20<1:08:26,	3.88s/it]
16%	203/1261	[14:24<1:08:11,	3.87s/it]
16%	204/1261	[14:28<1:07:59,	3.86s/it]
16%	205/1261	[14:32<1:07:55,	3.86s/it]
16%	206/1261	[14:36<1:07:48,	3.86s/it]
16%	207/1261	[14:39<1:07:51,	3.86s/it]
16%	208/1261	[14:43<1:07:48,	3.86s/it]
17%	209/1261	[14:47<1:07:31,	3.85s/it]
17%	210/1261	[14:51<1:07:40,	3.86s/it]
17%	211/1261	[14:55<1:07:26,	3.85s/it]
17%	212/1261	[14:59<1:07:28,	3.86s/it]
17%	213/1261	[15:03<1:07:13,	3.85s/it]
17%	214/1261	[15:06<1:07:40,	3.88s/it]
17%	215/1261	[15:10<1:07:46,	3.89s/it]
17%	216/1261	[15:14<1:07:47,	3.89s/it]
17%	217/1261	[15:18<1:07:23,	3.87s/it]
17%	218/1261	[15:22<1:07:19,	3.87s/it]
17%	219/1261	[15:26<1:07:07,	3.86s/it]
17%	220/1261	[15:30<1:06:58,	3.86s/it]
18%	221/1261	[15:33<1:06:43,	3.85s/it]
18%	222/1261	[15:37<1:06:45,	3.86s/it]
18%	223/1261	[15:41<1:07:02,	3.87s/it]
18%	224/1261	[15:45<1:06:51,	3.87s/it]
18%	225/1261	[15:49<1:07:02,	3.88s/it]
18%	226/1261	[15:53<1:06:56,	3.88s/it]
18%	227/1261	[15:57<1:06:50,	3.88s/it]
18%	228/1261	[16:01<1:06:52,	3.88s/it]
18%	229/1261	[16:05<1:06:37,	3.87s/it]
18%	230/1261	[16:08<1:06:25,	3.87s/it]
18%	231/1261	[16:12<1:05:59,	3.84s/it]
18%	232/1261	[16:16<1:06:08,	3.86s/it]
18%	233/1261	[16:20<1:06:04,	3.86s/it]
19%	234/1261	[16:24<1:05:55,	3.85s/it]
19%	235/1261	[16:28<1:05:52,	3.85s/it]
19%	236/1261	[16:31<1:05:56,	3.86s/it]
19%	237/1261	[16:35<1:06:03,	3.87s/it]
19%	238/1261	[16:39<1:06:00,	3.87s/it]
19%	239/1261	[16:43<1:05:51,	3.87s/it]
19%	240/1261	[16:47<1:05:29,	3.85s/it]
19%	241/1261	[16:51<1:05:23,	3.85s/it]
19%	242/1261	[16:55<1:05:34,	3.86s/it]
19%	243/1261	[16:59<1:05:30,	3.86s/it]
19%	244/1261	[17:02<1:05:35,	3.87s/it]
19%	245/1261	[17:06<1:05:20,	3.86s/it]
20%	246/1261	[17:10<1:05:29,	3.87s/it]
20%	247/1261	[17:14<1:05:35,	3.88s/it]
20%	248/1261	[17:18<1:05:30,	3.88s/it]
20%	249/1261	[17:22<1:05:27,	3.88s/it]
20%	250/1261	[17:26<1:05:17,	3.87s/it]
20%	251/1261	[17:30<1:05:02,	3.86s/it]
20%	252/1261	[17:33<1:04:53,	3.86s/it]
20%	253/1261	[17:37<1:04:47,	3.86s/it]
20%	254/1261	[17:41<1:05:09,	3.88s/it]
20%	255/1261	[17:45<1:05:28,	3.90s/it]
20%	256/1261	[17:49<1:05:41,	3.92s/it]
20%	257/1261	[17:53<1:05:39,	3.92s/it]
20%	258/1261	[17:57<1:05:32,	3.92s/it]
21%	259/1261	[18:01<1:05:24,	3.92s/it]
21%	260/1261	[18:05<1:05:00,	3.90s/it]
21%	261/1261	[18:08<1:04:27,	3.87s/it]
21%	262/1261	[18:12<1:04:34,	3.88s/it]
21%	263/1261	[18:16<1:04:16,	3.86s/it]
21%	264/1261	[18:20<1:04:13,	3.86s/it]
21%	265/1261	[18:24<1:04:18,	3.87s/it]

21%	266/1261	[18:28<1:04:23, 3.88s/it]
21%	267/1261	[18:32<1:04:35, 3.90s/it]
21%	268/1261	[18:36<1:04:06, 3.87s/it]
21%	269/1261	[18:39<1:03:49, 3.86s/it]
21%	270/1261	[18:43<1:04:02, 3.88s/it]
21%	271/1261	[18:47<1:03:36, 3.86s/it]
22%	272/1261	[18:51<1:03:42, 3.87s/it]
22%	273/1261	[18:55<1:03:18, 3.84s/it]
22%	274/1261	[18:59<1:03:09, 3.84s/it]
22%	275/1261	[19:03<1:03:05, 3.84s/it]
22%	276/1261	[19:06<1:03:11, 3.85s/it]
22%	277/1261	[19:10<1:03:22, 3.86s/it]
22%	278/1261	[19:14<1:03:49, 3.90s/it]
22%	279/1261	[19:18<1:03:31, 3.88s/it]
22%	280/1261	[19:22<1:03:05, 3.86s/it]
22%	281/1261	[19:26<1:02:57, 3.85s/it]
22%	282/1261	[19:30<1:02:42, 3.84s/it]
22%	283/1261	[19:33<1:02:35, 3.84s/it]
23%	284/1261	[19:37<1:02:32, 3.84s/it]
23%	285/1261	[19:41<1:02:28, 3.84s/it]
23%	286/1261	[19:45<1:02:11, 3.83s/it]
23%	287/1261	[19:49<1:02:11, 3.83s/it]
23%	288/1261	[19:53<1:01:49, 3.81s/it]
23%	289/1261	[19:56<1:01:44, 3.81s/it]
23%	290/1261	[20:00<1:01:41, 3.81s/it]
23%	291/1261	[20:04<1:01:40, 3.81s/it]
23%	292/1261	[20:08<1:01:44, 3.82s/it]
23%	293/1261	[20:12<1:01:31, 3.81s/it]
23%	294/1261	[20:15<1:01:30, 3.82s/it]
23%	295/1261	[20:19<1:01:34, 3.82s/it]
23%	296/1261	[20:23<1:01:31, 3.83s/it]
24%	297/1261	[20:27<1:01:29, 3.83s/it]
24%	298/1261	[20:31<1:01:13, 3.82s/it]
24%	299/1261	[20:35<1:01:19, 3.82s/it]
24%	300/1261	[20:38<1:01:25, 3.83s/it]
24%	301/1261	[20:42<1:01:10, 3.82s/it]
24%	302/1261	[20:46<1:01:03, 3.82s/it]
24%	303/1261	[20:50<1:01:17, 3.84s/it]
24%	304/1261	[20:54<1:01:12, 3.84s/it]
24%	305/1261	[20:58<1:01:27, 3.86s/it]
24%	306/1261	[21:01<1:01:21, 3.86s/it]
24%	307/1261	[21:05<1:01:15, 3.85s/it]
24%	308/1261	[21:09<1:00:59, 3.84s/it]
25%	309/1261	[21:13<1:01:13, 3.86s/it]
25%	310/1261	[21:17<1:00:54, 3.84s/it]
25%	311/1261	[21:21<1:01:02, 3.85s/it]
25%	312/1261	[21:25<1:02:20, 3.94s/it]
25%	313/1261	[21:30<1:07:14, 4.26s/it]
25%	314/1261	[21:35<1:10:04, 4.44s/it]
25%	315/1261	[21:39<1:10:45, 4.49s/it]
25%	316/1261	[21:43<1:08:36, 4.36s/it]
25%	317/1261	[21:48<1:07:33, 4.29s/it]
25%	318/1261	[21:52<1:09:59, 4.45s/it]
25%	319/1261	[21:56<1:08:19, 4.35s/it]
25%	320/1261	[22:00<1:05:48, 4.20s/it]
25%	321/1261	[22:04<1:04:01, 4.09s/it]
26%	322/1261	[22:08<1:02:58, 4.02s/it]
26%	323/1261	[22:12<1:01:49, 3.95s/it]
26%	324/1261	[22:16<1:00:59, 3.91s/it]
26%	325/1261	[22:19<1:00:38, 3.89s/it]
26%	326/1261	[22:23<1:00:14, 3.87s/it]
26%	327/1261	[22:27<59:46, 3.84s/it]
26%	328/1261	[22:31<59:42, 3.84s/it]
26%	329/1261	[22:35<59:47, 3.85s/it]
26%	330/1261	[22:39<59:34, 3.84s/it]
26%	331/1261	[22:42<59:18, 3.83s/it]
26%	332/1261	[22:46<59:11, 3.82s/it]
26%	333/1261	[22:50<59:06, 3.82s/it]
26%	334/1261	[22:54<58:57, 3.82s/it]
27%	335/1261	[22:58<58:54, 3.82s/it]
27%	336/1261	[23:01<58:36, 3.80s/it]
27%	337/1261	[23:05<58:33, 3.80s/it]
27%	338/1261	[23:09<58:39, 3.81s/it]
27%	339/1261	[23:13<58:54, 3.83s/it]
27%	340/1261	[23:17<58:50, 3.83s/it]
27%	341/1261	[23:21<58:37, 3.82s/it]
27%	342/1261	[23:24<58:30, 3.82s/it]
27%	343/1261	[23:28<58:32, 3.83s/it]
27%	344/1261	[23:32<58:33, 3.83s/it]
27%	345/1261	[23:36<58:33, 3.84s/it]
27%	346/1261	[23:40<58:28, 3.83s/it]
28%	347/1261	[23:44<58:24, 3.83s/it]
28%	348/1261	[23:47<58:07, 3.82s/it]
28%	349/1261	[23:51<57:50, 3.81s/it]
28%	350/1261	[23:55<57:50, 3.81s/it]
28%	351/1261	[23:59<57:52, 3.82s/it]
28%	352/1261	[24:03<57:49, 3.82s/it]
28%	353/1261	[24:06<57:45, 3.82s/it]
28%	354/1261	[24:10<57:39, 3.81s/it]

28%		355/1261 [24:14<57:36, 3.81s/it]
28%		356/1261 [24:18<57:22, 3.80s/it]
28%		357/1261 [24:22<57:23, 3.81s/it]
28%		358/1261 [24:25<57:16, 3.81s/it]
28%		359/1261 [24:29<57:17, 3.81s/it]
29%		360/1261 [24:33<57:06, 3.80s/it]
29%		361/1261 [24:37<57:05, 3.81s/it]
29%		362/1261 [24:41<57:08, 3.81s/it]
29%		363/1261 [24:45<57:18, 3.83s/it]
29%		364/1261 [24:48<56:55, 3.81s/it]
29%		365/1261 [24:52<56:51, 3.81s/it]
29%		366/1261 [24:56<56:46, 3.81s/it]
29%		367/1261 [25:00<56:36, 3.80s/it]
29%		368/1261 [25:03<56:32, 3.80s/it]
29%		369/1261 [25:07<56:27, 3.80s/it]
29%		370/1261 [25:11<56:30, 3.80s/it]
29%		371/1261 [25:15<56:30, 3.81s/it]
30%		372/1261 [25:19<56:39, 3.82s/it]
30%		373/1261 [25:23<56:41, 3.83s/it]
30%		374/1261 [25:26<56:28, 3.82s/it]
30%		375/1261 [25:30<56:26, 3.82s/it]
30%		376/1261 [25:34<56:14, 3.81s/it]
30%		377/1261 [25:38<56:17, 3.82s/it]
30%		378/1261 [25:42<56:08, 3.81s/it]
30%		379/1261 [25:45<56:07, 3.82s/it]
30%		380/1261 [25:49<56:00, 3.81s/it]
30%		381/1261 [25:53<55:50, 3.81s/it]
30%		382/1261 [25:57<56:09, 3.83s/it]
30%		383/1261 [26:01<56:03, 3.83s/it]
30%		384/1261 [26:05<55:50, 3.82s/it]
31%		385/1261 [26:08<55:45, 3.82s/it]
31%		386/1261 [26:12<56:20, 3.86s/it]
31%		387/1261 [26:16<55:51, 3.83s/it]
31%		388/1261 [26:20<55:54, 3.84s/it]
31%		389/1261 [26:24<55:37, 3.83s/it]
31%		390/1261 [26:28<55:23, 3.82s/it]
31%		391/1261 [26:31<55:14, 3.81s/it]
31%		392/1261 [26:35<55:12, 3.81s/it]
31%		393/1261 [26:39<55:08, 3.81s/it]
31%		394/1261 [26:43<54:47, 3.79s/it]
31%		395/1261 [26:47<54:43, 3.79s/it]
31%		396/1261 [26:50<54:46, 3.80s/it]
31%		397/1261 [26:54<54:39, 3.80s/it]
32%		398/1261 [26:58<54:31, 3.79s/it]
32%		399/1261 [27:02<54:26, 3.79s/it]
32%		400/1261 [27:06<54:24, 3.79s/it]
32%		401/1261 [27:09<54:17, 3.79s/it]
32%		402/1261 [27:13<54:23, 3.80s/it]
32%		403/1261 [27:17<54:18, 3.80s/it]
32%		404/1261 [27:21<54:08, 3.79s/it]
32%		405/1261 [27:25<54:12, 3.80s/it]
32%		406/1261 [27:28<54:04, 3.79s/it]
32%		407/1261 [27:32<54:01, 3.80s/it]
32%		408/1261 [27:36<54:06, 3.81s/it]
32%		409/1261 [27:40<54:06, 3.81s/it]
33%		410/1261 [27:44<54:26, 3.84s/it]
33%		411/1261 [27:47<54:10, 3.82s/it]
33%		412/1261 [27:51<53:57, 3.81s/it]
33%		413/1261 [27:55<54:25, 3.85s/it]
33%		414/1261 [27:59<54:14, 3.84s/it]
33%		415/1261 [28:03<54:01, 3.83s/it]
33%		416/1261 [28:07<53:45, 3.82s/it]
33%		417/1261 [28:10<53:51, 3.83s/it]
33%		418/1261 [28:14<53:49, 3.83s/it]
33%		419/1261 [28:18<53:45, 3.83s/it]
33%		420/1261 [28:22<53:36, 3.82s/it]
33%		421/1261 [28:26<53:26, 3.82s/it]
33%		422/1261 [28:30<53:21, 3.82s/it]
34%		423/1261 [28:33<53:15, 3.81s/it]
34%		424/1261 [28:37<53:04, 3.80s/it]
34%		425/1261 [28:41<52:56, 3.80s/it]
34%		426/1261 [28:45<52:57, 3.81s/it]
34%		427/1261 [28:48<52:44, 3.79s/it]
34%		428/1261 [28:52<52:41, 3.80s/it]
34%		429/1261 [28:56<52:32, 3.79s/it]
34%		430/1261 [29:00<52:37, 3.80s/it]
34%		431/1261 [29:04<52:38, 3.81s/it]
34%		432/1261 [29:07<52:26, 3.80s/it]
34%		433/1261 [29:11<52:32, 3.81s/it]
34%		434/1261 [29:15<52:27, 3.81s/it]
34%		435/1261 [29:19<52:20, 3.80s/it]
35%		436/1261 [29:23<52:18, 3.80s/it]
35%		437/1261 [29:26<52:05, 3.79s/it]
35%		438/1261 [29:30<52:11, 3.81s/it]
35%		439/1261 [29:34<52:00, 3.80s/it]
35%		440/1261 [29:38<52:00, 3.80s/it]
35%		441/1261 [29:42<51:56, 3.80s/it]
35%		442/1261 [29:46<51:58, 3.81s/it]
35%		443/1261 [29:49<51:58, 3.81s/it]

35%		444/1261 [29:53<51:46, 3.80s/it]
35%		445/1261 [29:57<51:43, 3.80s/it]
35%		446/1261 [30:01<51:32, 3.79s/it]
35%		447/1261 [30:05<51:31, 3.80s/it]
36%		448/1261 [30:08<51:36, 3.81s/it]
36%		449/1261 [30:12<51:21, 3.80s/it]
36%		450/1261 [30:16<51:35, 3.82s/it]
36%		451/1261 [30:20<51:39, 3.83s/it]
36%		452/1261 [30:24<51:43, 3.84s/it]
36%		453/1261 [30:28<51:37, 3.83s/it]
36%		454/1261 [30:31<51:18, 3.81s/it]
36%		455/1261 [30:35<51:15, 3.82s/it]
36%		456/1261 [30:39<51:06, 3.81s/it]
36%		457/1261 [30:43<50:58, 3.80s/it]
36%		458/1261 [30:47<51:05, 3.82s/it]
36%		459/1261 [30:50<50:51, 3.81s/it]
36%		460/1261 [30:54<50:47, 3.81s/it]
37%		461/1261 [30:58<50:40, 3.80s/it]
37%		462/1261 [31:02<50:45, 3.81s/it]
37%		463/1261 [31:06<50:42, 3.81s/it]
37%		464/1261 [31:09<50:27, 3.80s/it]
37%		465/1261 [31:13<50:24, 3.80s/it]
37%		466/1261 [31:17<50:23, 3.80s/it]
37%		467/1261 [31:21<50:33, 3.82s/it]
37%		468/1261 [31:25<50:29, 3.82s/it]
37%		469/1261 [31:28<50:22, 3.82s/it]
37%		470/1261 [31:32<50:23, 3.82s/it]
37%		471/1261 [31:36<50:17, 3.82s/it]
37%		472/1261 [31:40<50:09, 3.81s/it]
38%		473/1261 [31:44<50:08, 3.82s/it]
38%		474/1261 [31:48<50:05, 3.82s/it]
38%		475/1261 [31:51<50:01, 3.82s/it]
38%		476/1261 [31:55<49:54, 3.81s/it]
38%		477/1261 [31:59<49:41, 3.80s/it]
38%		478/1261 [32:03<49:44, 3.81s/it]
38%		479/1261 [32:07<49:36, 3.81s/it]
38%		480/1261 [32:11<50:07, 3.85s/it]
38%		481/1261 [32:14<49:51, 3.83s/it]
38%		482/1261 [32:18<49:22, 3.80s/it]
38%		483/1261 [32:22<49:22, 3.81s/it]
38%		484/1261 [32:26<49:07, 3.79s/it]
38%		485/1261 [32:29<49:00, 3.79s/it]
39%		486/1261 [32:33<48:53, 3.79s/it]
39%		487/1261 [32:37<48:43, 3.78s/it]
39%		488/1261 [32:41<50:30, 3.92s/it]
39%		489/1261 [32:45<49:58, 3.88s/it]
39%		490/1261 [32:49<49:29, 3.85s/it]
39%		491/1261 [32:53<49:08, 3.83s/it]
39%		492/1261 [32:56<48:39, 3.80s/it]
39%		493/1261 [33:00<48:26, 3.79s/it]
39%		494/1261 [33:04<48:19, 3.78s/it]
39%		495/1261 [33:08<48:09, 3.77s/it]
39%		496/1261 [33:11<48:09, 3.78s/it]
39%		497/1261 [33:15<48:17, 3.79s/it]
39%		498/1261 [33:19<48:09, 3.79s/it]
40%		499/1261 [33:23<47:56, 3.77s/it]
40%		500/1261 [33:26<47:53, 3.78s/it]
40%		501/1261 [33:30<47:54, 3.78s/it]
40%		502/1261 [33:34<47:42, 3.77s/it]
40%		503/1261 [33:38<47:39, 3.77s/it]
40%		504/1261 [33:42<47:35, 3.77s/it]
40%		505/1261 [33:45<47:33, 3.77s/it]
40%		506/1261 [33:49<47:23, 3.77s/it]
40%		507/1261 [33:53<47:20, 3.77s/it]
40%		508/1261 [33:57<47:19, 3.77s/it]
40%		509/1261 [34:00<47:13, 3.77s/it]
40%		510/1261 [34:04<47:10, 3.77s/it]
41%		511/1261 [34:08<47:08, 3.77s/it]
41%		512/1261 [34:12<46:56, 3.76s/it]
41%		513/1261 [34:15<47:02, 3.77s/it]
41%		514/1261 [34:19<47:01, 3.78s/it]
41%		515/1261 [34:23<46:53, 3.77s/it]
41%		516/1261 [34:27<46:47, 3.77s/it]
41%		517/1261 [34:31<46:46, 3.77s/it]
41%		518/1261 [34:34<46:43, 3.77s/it]
41%		519/1261 [34:38<46:37, 3.77s/it]
41%		520/1261 [34:42<46:33, 3.77s/it]
41%		521/1261 [34:46<46:42, 3.79s/it]
41%		522/1261 [34:49<46:33, 3.78s/it]
41%		523/1261 [34:53<46:26, 3.78s/it]
42%		524/1261 [34:57<46:15, 3.77s/it]
42%		525/1261 [35:01<46:07, 3.76s/it]
42%		526/1261 [35:04<46:09, 3.77s/it]
42%		527/1261 [35:08<45:55, 3.75s/it]
42%		528/1261 [35:12<45:56, 3.76s/it]
42%		529/1261 [35:16<45:51, 3.76s/it]
42%		530/1261 [35:20<45:48, 3.76s/it]
42%		531/1261 [35:23<46:04, 3.79s/it]
42%		532/1261 [35:27<45:52, 3.78s/it]

42%		533/1261 [35:31<45:46, 3.77s/it]
42%		534/1261 [35:35<45:36, 3.76s/it]
42%		535/1261 [35:38<45:32, 3.76s/it]
43%		536/1261 [35:42<45:30, 3.77s/it]
43%		537/1261 [35:46<45:26, 3.77s/it]
43%		538/1261 [35:50<45:23, 3.77s/it]
43%		539/1261 [35:53<45:19, 3.77s/it]
43%		540/1261 [35:57<45:12, 3.76s/it]
43%		541/1261 [36:01<45:17, 3.77s/it]
43%		542/1261 [36:05<45:08, 3.77s/it]
43%		543/1261 [36:09<45:05, 3.77s/it]
43%		544/1261 [36:12<45:02, 3.77s/it]
43%		545/1261 [36:16<44:58, 3.77s/it]
43%		546/1261 [36:20<45:00, 3.78s/it]
43%		547/1261 [36:24<44:51, 3.77s/it]
43%		548/1261 [36:27<44:55, 3.78s/it]
44%		549/1261 [36:31<44:55, 3.79s/it]
44%		550/1261 [36:35<44:47, 3.78s/it]
44%		551/1261 [36:39<44:48, 3.79s/it]
44%		552/1261 [36:43<44:48, 3.79s/it]
44%		553/1261 [36:46<44:39, 3.79s/it]
44%		554/1261 [36:50<44:30, 3.78s/it]
44%		555/1261 [36:54<44:24, 3.77s/it]
44%		556/1261 [36:58<44:22, 3.78s/it]
44%		557/1261 [37:01<44:11, 3.77s/it]
44%		558/1261 [37:05<44:13, 3.77s/it]
44%		559/1261 [37:09<44:09, 3.77s/it]
44%		560/1261 [37:13<43:58, 3.76s/it]
44%		561/1261 [37:16<43:48, 3.76s/it]
45%		562/1261 [37:20<43:44, 3.75s/it]
45%		563/1261 [37:24<43:33, 3.74s/it]
45%		564/1261 [37:28<43:30, 3.75s/it]
45%		565/1261 [37:31<43:17, 3.73s/it]
45%		566/1261 [37:35<43:17, 3.74s/it]
45%		567/1261 [37:39<43:21, 3.75s/it]
45%		568/1261 [37:43<43:16, 3.75s/it]
45%		569/1261 [37:46<43:09, 3.74s/it]
45%		570/1261 [37:50<43:04, 3.74s/it]
45%		571/1261 [37:54<42:55, 3.73s/it]
45%		572/1261 [37:58<42:51, 3.73s/it]
45%		573/1261 [38:01<42:45, 3.73s/it]
46%		574/1261 [38:05<42:49, 3.74s/it]
46%		575/1261 [38:09<42:50, 3.75s/it]
46%		576/1261 [38:13<42:47, 3.75s/it]
46%		577/1261 [38:16<42:48, 3.75s/it]
46%		578/1261 [38:20<42:36, 3.74s/it]
46%		579/1261 [38:24<42:38, 3.75s/it]
46%		580/1261 [38:28<42:34, 3.75s/it]
46%		581/1261 [38:31<42:27, 3.75s/it]
46%		582/1261 [38:35<42:32, 3.76s/it]
46%		583/1261 [38:39<42:24, 3.75s/it]
46%		584/1261 [38:43<42:17, 3.75s/it]
46%		585/1261 [38:46<42:18, 3.76s/it]
46%		586/1261 [38:50<42:02, 3.74s/it]
47%		587/1261 [38:54<41:59, 3.74s/it]
47%		588/1261 [38:58<42:20, 3.78s/it]
47%		589/1261 [39:01<42:08, 3.76s/it]
47%		590/1261 [39:05<41:59, 3.76s/it]
47%		591/1261 [39:09<41:40, 3.73s/it]
47%		592/1261 [39:13<41:46, 3.75s/it]
47%		593/1261 [39:16<41:35, 3.74s/it]
47%		594/1261 [39:20<41:42, 3.75s/it]
47%		595/1261 [39:24<41:40, 3.75s/it]
47%		596/1261 [39:28<42:29, 3.83s/it]
47%		597/1261 [39:32<44:36, 4.03s/it]
47%		598/1261 [39:37<45:42, 4.14s/it]
48%		599/1261 [39:41<45:59, 4.17s/it]
48%		600/1261 [39:45<45:20, 4.12s/it]
48%		601/1261 [39:49<46:02, 4.19s/it]
48%		602/1261 [39:54<46:27, 4.23s/it]
48%		603/1261 [39:58<45:38, 4.16s/it]
48%		604/1261 [40:02<45:53, 4.19s/it]
48%		605/1261 [40:07<47:35, 4.35s/it]
48%		606/1261 [40:11<47:30, 4.35s/it]
48%		607/1261 [40:15<47:10, 4.33s/it]
48%		608/1261 [40:19<46:31, 4.27s/it]
48%		609/1261 [40:24<47:41, 4.39s/it]
48%		610/1261 [40:28<46:10, 4.26s/it]
48%		611/1261 [40:32<45:05, 4.16s/it]
49%		612/1261 [40:36<45:06, 4.17s/it]
49%		613/1261 [40:40<44:59, 4.17s/it]
49%		614/1261 [40:44<44:52, 4.16s/it]
49%		615/1261 [40:49<44:43, 4.15s/it]
49%		616/1261 [40:53<44:57, 4.18s/it]
49%		617/1261 [40:57<44:56, 4.19s/it]
49%		618/1261 [41:01<45:12, 4.22s/it]
49%		619/1261 [41:05<44:19, 4.14s/it]
49%		620/1261 [41:09<43:35, 4.08s/it]
49%		621/1261 [41:13<42:49, 4.01s/it]

49%	[REDACTED]	622/1261 [41:17<42:20, 3.98s/it]
49%	[REDACTED]	623/1261 [41:21<41:58, 3.95s/it]
49%	[REDACTED]	624/1261 [41:25<41:29, 3.91s/it]
50%	[REDACTED]	625/1261 [41:29<41:16, 3.89s/it]
50%	[REDACTED]	626/1261 [41:32<41:01, 3.88s/it]
50%	[REDACTED]	627/1261 [41:36<40:52, 3.87s/it]
50%	[REDACTED]	628/1261 [41:40<41:20, 3.92s/it]
50%	[REDACTED]	629/1261 [41:44<41:38, 3.95s/it]
50%	[REDACTED]	630/1261 [41:48<41:46, 3.97s/it]
50%	[REDACTED]	631/1261 [41:53<42:26, 4.04s/it]
50%	[REDACTED]	632/1261 [41:57<42:18, 4.04s/it]
50%	[REDACTED]	633/1261 [42:01<42:31, 4.06s/it]
50%	[REDACTED]	634/1261 [42:05<42:37, 4.08s/it]
50%	[REDACTED]	635/1261 [42:09<43:16, 4.15s/it]
50%	[REDACTED]	636/1261 [42:13<43:47, 4.20s/it]
51%	[REDACTED]	637/1261 [42:17<42:56, 4.13s/it]
51%	[REDACTED]	638/1261 [42:21<42:01, 4.05s/it]
51%	[REDACTED]	639/1261 [42:25<42:22, 4.09s/it]
51%	[REDACTED]	640/1261 [42:30<42:34, 4.11s/it]
51%	[REDACTED]	641/1261 [42:34<42:52, 4.15s/it]
51%	[REDACTED]	642/1261 [42:38<42:27, 4.12s/it]
51%	[REDACTED]	643/1261 [42:42<41:32, 4.03s/it]
51%	[REDACTED]	644/1261 [42:46<40:48, 3.97s/it]
51%	[REDACTED]	645/1261 [42:49<40:21, 3.93s/it]
51%	[REDACTED]	646/1261 [42:53<39:58, 3.90s/it]
51%	[REDACTED]	647/1261 [42:57<39:42, 3.88s/it]
51%	[REDACTED]	648/1261 [43:01<39:28, 3.86s/it]
51%	[REDACTED]	649/1261 [43:05<39:14, 3.85s/it]
52%	[REDACTED]	650/1261 [43:08<39:02, 3.83s/it]
52%	[REDACTED]	651/1261 [43:12<39:11, 3.85s/it]
52%	[REDACTED]	652/1261 [43:16<39:05, 3.85s/it]
52%	[REDACTED]	653/1261 [43:20<39:02, 3.85s/it]
52%	[REDACTED]	654/1261 [43:24<38:54, 3.85s/it]
52%	[REDACTED]	655/1261 [43:28<38:43, 3.83s/it]
52%	[REDACTED]	656/1261 [43:32<38:59, 3.87s/it]
52%	[REDACTED]	657/1261 [43:36<38:57, 3.87s/it]
52%	[REDACTED]	658/1261 [43:39<38:53, 3.87s/it]
52%	[REDACTED]	659/1261 [43:43<38:45, 3.86s/it]
52%	[REDACTED]	660/1261 [43:47<38:30, 3.85s/it]
52%	[REDACTED]	661/1261 [43:51<38:19, 3.83s/it]
52%	[REDACTED]	662/1261 [43:55<38:06, 3.82s/it]
53%	[REDACTED]	663/1261 [43:58<37:59, 3.81s/it]
53%	[REDACTED]	664/1261 [44:02<37:52, 3.81s/it]
53%	[REDACTED]	665/1261 [44:06<37:47, 3.80s/it]
53%	[REDACTED]	666/1261 [44:10<37:40, 3.80s/it]
53%	[REDACTED]	667/1261 [44:14<37:40, 3.81s/it]
53%	[REDACTED]	668/1261 [44:17<37:35, 3.80s/it]
53%	[REDACTED]	669/1261 [44:21<37:33, 3.81s/it]
53%	[REDACTED]	670/1261 [44:25<37:28, 3.81s/it]
53%	[REDACTED]	671/1261 [44:29<37:22, 3.80s/it]
53%	[REDACTED]	672/1261 [44:33<37:15, 3.79s/it]
53%	[REDACTED]	673/1261 [44:36<37:10, 3.79s/it]
53%	[REDACTED]	674/1261 [44:40<37:02, 3.79s/it]
54%	[REDACTED]	675/1261 [44:44<36:56, 3.78s/it]
54%	[REDACTED]	676/1261 [44:48<36:51, 3.78s/it]
54%	[REDACTED]	677/1261 [44:51<36:46, 3.78s/it]
54%	[REDACTED]	678/1261 [44:55<36:43, 3.78s/it]
54%	[REDACTED]	679/1261 [44:59<36:38, 3.78s/it]
54%	[REDACTED]	680/1261 [45:03<36:42, 3.79s/it]
54%	[REDACTED]	681/1261 [45:07<36:40, 3.79s/it]
54%	[REDACTED]	682/1261 [45:10<36:34, 3.79s/it]
54%	[REDACTED]	683/1261 [45:14<36:37, 3.80s/it]
54%	[REDACTED]	684/1261 [45:18<36:29, 3.80s/it]
54%	[REDACTED]	685/1261 [45:22<36:30, 3.80s/it]
54%	[REDACTED]	686/1261 [45:26<36:22, 3.80s/it]
54%	[REDACTED]	687/1261 [45:29<36:14, 3.79s/it]
55%	[REDACTED]	688/1261 [45:33<36:14, 3.80s/it]
55%	[REDACTED]	689/1261 [45:37<36:07, 3.79s/it]
55%	[REDACTED]	690/1261 [45:41<36:03, 3.79s/it]
55%	[REDACTED]	691/1261 [45:45<35:58, 3.79s/it]
55%	[REDACTED]	692/1261 [45:48<35:59, 3.80s/it]
55%	[REDACTED]	693/1261 [45:52<36:04, 3.81s/it]
55%	[REDACTED]	694/1261 [45:56<36:10, 3.83s/it]
55%	[REDACTED]	695/1261 [46:00<36:37, 3.88s/it]
55%	[REDACTED]	696/1261 [46:04<36:30, 3.88s/it]
55%	[REDACTED]	697/1261 [46:08<36:10, 3.85s/it]
55%	[REDACTED]	698/1261 [46:12<36:04, 3.85s/it]
55%	[REDACTED]	699/1261 [46:15<35:53, 3.83s/it]
56%	[REDACTED]	700/1261 [46:19<35:46, 3.83s/it]
56%	[REDACTED]	701/1261 [46:23<35:38, 3.82s/it]
56%	[REDACTED]	702/1261 [46:27<35:30, 3.81s/it]
56%	[REDACTED]	703/1261 [46:31<35:27, 3.81s/it]
56%	[REDACTED]	704/1261 [46:34<35:18, 3.80s/it]
56%	[REDACTED]	705/1261 [46:38<35:16, 3.81s/it]
56%	[REDACTED]	706/1261 [46:42<35:12, 3.81s/it]
56%	[REDACTED]	707/1261 [46:46<35:06, 3.80s/it]
56%	[REDACTED]	708/1261 [46:50<35:05, 3.81s/it]
56%	[REDACTED]	709/1261 [46:53<35:00, 3.81s/it]
56%	[REDACTED]	710/1261 [46:57<34:54, 3.80s/it]

56%	[REDACTED]	711/1261 [47:01<34:51, 3.80s/it]
56%	[REDACTED]	712/1261 [47:05<34:50, 3.81s/it]
57%	[REDACTED]	713/1261 [47:09<34:46, 3.81s/it]
57%	[REDACTED]	714/1261 [47:12<34:40, 3.80s/it]
57%	[REDACTED]	715/1261 [47:16<34:35, 3.80s/it]
57%	[REDACTED]	716/1261 [47:20<34:35, 3.81s/it]
57%	[REDACTED]	717/1261 [47:24<34:36, 3.82s/it]
57%	[REDACTED]	718/1261 [47:28<34:28, 3.81s/it]
57%	[REDACTED]	719/1261 [47:31<34:19, 3.80s/it]
57%	[REDACTED]	720/1261 [47:35<34:14, 3.80s/it]
57%	[REDACTED]	721/1261 [47:39<34:12, 3.80s/it]
57%	[REDACTED]	722/1261 [47:43<34:04, 3.79s/it]
57%	[REDACTED]	723/1261 [47:47<34:01, 3.79s/it]
57%	[REDACTED]	724/1261 [47:50<33:55, 3.79s/it]
57%	[REDACTED]	725/1261 [47:54<33:51, 3.79s/it]
58%	[REDACTED]	726/1261 [47:58<33:46, 3.79s/it]
58%	[REDACTED]	727/1261 [48:02<33:41, 3.79s/it]
58%	[REDACTED]	728/1261 [48:06<33:38, 3.79s/it]
58%	[REDACTED]	729/1261 [48:09<33:40, 3.80s/it]
58%	[REDACTED]	730/1261 [48:13<33:46, 3.82s/it]
58%	[REDACTED]	731/1261 [48:17<33:44, 3.82s/it]
58%	[REDACTED]	732/1261 [48:21<33:36, 3.81s/it]
58%	[REDACTED]	733/1261 [48:25<33:31, 3.81s/it]
58%	[REDACTED]	734/1261 [48:28<33:26, 3.81s/it]
58%	[REDACTED]	735/1261 [48:32<33:22, 3.81s/it]
58%	[REDACTED]	736/1261 [48:36<33:15, 3.80s/it]
58%	[REDACTED]	737/1261 [48:40<33:10, 3.80s/it]
59%	[REDACTED]	738/1261 [48:44<33:06, 3.80s/it]
59%	[REDACTED]	739/1261 [48:47<32:58, 3.79s/it]
59%	[REDACTED]	740/1261 [48:51<32:56, 3.79s/it]
59%	[REDACTED]	741/1261 [48:55<32:54, 3.80s/it]
59%	[REDACTED]	742/1261 [48:59<32:49, 3.79s/it]
59%	[REDACTED]	743/1261 [49:03<32:45, 3.80s/it]
59%	[REDACTED]	744/1261 [49:06<32:43, 3.80s/it]
59%	[REDACTED]	745/1261 [49:10<32:40, 3.80s/it]
59%	[REDACTED]	746/1261 [49:14<32:49, 3.82s/it]
59%	[REDACTED]	747/1261 [49:18<32:44, 3.82s/it]
59%	[REDACTED]	748/1261 [49:22<32:44, 3.83s/it]
59%	[REDACTED]	749/1261 [49:26<32:36, 3.82s/it]
59%	[REDACTED]	750/1261 [49:29<32:29, 3.82s/it]
60%	[REDACTED]	751/1261 [49:33<32:21, 3.81s/it]
60%	[REDACTED]	752/1261 [49:37<32:13, 3.80s/it]
60%	[REDACTED]	753/1261 [49:41<32:12, 3.80s/it]
60%	[REDACTED]	754/1261 [49:45<32:06, 3.80s/it]
60%	[REDACTED]	755/1261 [49:48<32:00, 3.80s/it]
60%	[REDACTED]	756/1261 [49:52<32:02, 3.81s/it]
60%	[REDACTED]	757/1261 [49:56<32:01, 3.81s/it]
60%	[REDACTED]	758/1261 [50:00<31:55, 3.81s/it]
60%	[REDACTED]	759/1261 [50:04<31:50, 3.81s/it]
60%	[REDACTED]	760/1261 [50:07<31:46, 3.80s/it]
60%	[REDACTED]	761/1261 [50:11<31:41, 3.80s/it]
60%	[REDACTED]	762/1261 [50:15<31:35, 3.80s/it]
61%	[REDACTED]	763/1261 [50:19<31:33, 3.80s/it]
61%	[REDACTED]	764/1261 [50:23<31:34, 3.81s/it]
61%	[REDACTED]	765/1261 [50:26<31:25, 3.80s/it]
61%	[REDACTED]	766/1261 [50:30<31:34, 3.83s/it]
61%	[REDACTED]	767/1261 [50:34<31:28, 3.82s/it]
61%	[REDACTED]	768/1261 [50:38<31:21, 3.82s/it]
61%	[REDACTED]	769/1261 [50:42<31:42, 3.87s/it]
61%	[REDACTED]	770/1261 [50:46<32:07, 3.93s/it]
61%	[REDACTED]	771/1261 [50:50<31:44, 3.89s/it]
61%	[REDACTED]	772/1261 [50:54<31:32, 3.87s/it]
61%	[REDACTED]	773/1261 [50:57<31:18, 3.85s/it]
61%	[REDACTED]	774/1261 [51:01<31:07, 3.83s/it]
61%	[REDACTED]	775/1261 [51:05<30:58, 3.82s/it]
62%	[REDACTED]	776/1261 [51:09<30:52, 3.82s/it]
62%	[REDACTED]	777/1261 [51:13<30:47, 3.82s/it]
62%	[REDACTED]	778/1261 [51:16<30:43, 3.82s/it]
62%	[REDACTED]	779/1261 [51:20<30:37, 3.81s/it]
62%	[REDACTED]	780/1261 [51:24<30:31, 3.81s/it]
62%	[REDACTED]	781/1261 [51:28<30:26, 3.80s/it]
62%	[REDACTED]	782/1261 [51:32<30:25, 3.81s/it]
62%	[REDACTED]	783/1261 [51:35<30:19, 3.81s/it]
62%	[REDACTED]	784/1261 [51:39<30:11, 3.80s/it]
62%	[REDACTED]	785/1261 [51:43<30:07, 3.80s/it]
62%	[REDACTED]	786/1261 [51:47<30:06, 3.80s/it]
62%	[REDACTED]	787/1261 [51:51<30:16, 3.83s/it]
62%	[REDACTED]	788/1261 [51:55<30:35, 3.88s/it]
63%	[REDACTED]	789/1261 [51:59<31:03, 3.95s/it]
63%	[REDACTED]	790/1261 [52:03<31:27, 4.01s/it]
63%	[REDACTED]	791/1261 [52:07<31:46, 4.06s/it]
63%	[REDACTED]	792/1261 [52:11<31:11, 3.99s/it]
63%	[REDACTED]	793/1261 [52:15<30:53, 3.96s/it]
63%	[REDACTED]	794/1261 [52:19<30:41, 3.94s/it]
63%	[REDACTED]	795/1261 [52:23<31:21, 4.04s/it]
63%	[REDACTED]	796/1261 [52:27<31:56, 4.12s/it]
63%	[REDACTED]	797/1261 [52:31<31:14, 4.04s/it]
63%	[REDACTED]	798/1261 [52:35<30:54, 4.01s/it]
63%	[REDACTED]	799/1261 [52:39<30:52, 4.01s/it]

63%	[██████]	800/1261 [52:43<30:41, 3.99s/it]
64%	[██████]	801/1261 [52:48<32:00, 4.18s/it]
64%	[██████]	802/1261 [52:52<32:20, 4.23s/it]
64%	[██████]	803/1261 [52:56<32:42, 4.28s/it]
64%	[██████]	804/1261 [53:01<32:59, 4.33s/it]
64%	[██████]	805/1261 [53:05<32:26, 4.27s/it]
64%	[██████]	806/1261 [53:09<32:04, 4.23s/it]
64%	[██████]	807/1261 [53:13<32:08, 4.25s/it]
64%	[██████]	808/1261 [53:18<31:46, 4.21s/it]
64%	[██████]	809/1261 [53:22<31:29, 4.18s/it]
64%	[██████]	810/1261 [53:26<31:48, 4.23s/it]
64%	[██████]	811/1261 [53:30<31:27, 4.20s/it]
64%	[██████]	812/1261 [53:34<31:09, 4.16s/it]
64%	[██████]	813/1261 [53:39<31:29, 4.22s/it]
65%	[██████]	814/1261 [53:43<31:19, 4.21s/it]
65%	[██████]	815/1261 [53:47<31:21, 4.22s/it]
65%	[██████]	816/1261 [53:51<31:00, 4.18s/it]
65%	[██████]	817/1261 [53:55<31:12, 4.22s/it]
65%	[██████]	818/1261 [54:00<30:56, 4.19s/it]
65%	[██████]	819/1261 [54:04<30:44, 4.17s/it]
65%	[██████]	820/1261 [54:08<31:29, 4.28s/it]
65%	[██████]	821/1261 [54:12<31:15, 4.26s/it]
65%	[██████]	822/1261 [54:17<30:48, 4.21s/it]
65%	[██████]	823/1261 [54:21<30:49, 4.22s/it]
65%	[██████]	824/1261 [54:25<30:32, 4.19s/it]
65%	[██████]	825/1261 [54:29<30:14, 4.16s/it]
66%	[██████]	826/1261 [54:33<30:25, 4.20s/it]
66%	[██████]	827/1261 [54:38<30:54, 4.27s/it]
66%	[██████]	828/1261 [54:42<30:37, 4.24s/it]
66%	[██████]	829/1261 [54:46<30:15, 4.20s/it]
66%	[██████]	830/1261 [54:50<30:18, 4.22s/it]
66%	[██████]	831/1261 [54:54<30:06, 4.20s/it]
66%	[██████]	832/1261 [54:59<29:51, 4.18s/it]
66%	[██████]	833/1261 [55:03<29:40, 4.16s/it]
66%	[██████]	834/1261 [55:07<29:30, 4.15s/it]
66%	[██████]	835/1261 [55:11<29:22, 4.14s/it]
66%	[██████]	836/1261 [55:15<29:17, 4.14s/it]
66%	[██████]	837/1261 [55:19<29:08, 4.12s/it]
66%	[██████]	838/1261 [55:23<29:00, 4.11s/it]
67%	[██████]	839/1261 [55:27<28:55, 4.11s/it]
67%	[██████]	840/1261 [55:31<28:52, 4.11s/it]
67%	[██████]	841/1261 [55:36<28:46, 4.11s/it]
67%	[██████]	842/1261 [55:40<28:47, 4.12s/it]
67%	[██████]	843/1261 [55:44<28:44, 4.13s/it]
67%	[██████]	844/1261 [55:48<28:36, 4.12s/it]
67%	[██████]	845/1261 [55:52<28:34, 4.12s/it]
67%	[██████]	846/1261 [55:56<28:24, 4.11s/it]
67%	[██████]	847/1261 [56:00<28:22, 4.11s/it]
67%	[██████]	848/1261 [56:04<28:20, 4.12s/it]
67%	[██████]	849/1261 [56:08<28:12, 4.11s/it]
67%	[██████]	850/1261 [56:13<28:07, 4.11s/it]
67%	[██████]	851/1261 [56:17<28:00, 4.10s/it]
68%	[██████]	852/1261 [56:21<27:58, 4.11s/it]
68%	[██████]	853/1261 [56:25<27:53, 4.10s/it]
68%	[██████]	854/1261 [56:29<27:49, 4.10s/it]
68%	[██████]	855/1261 [56:34<28:38, 4.23s/it]
68%	[██████]	856/1261 [56:38<28:42, 4.25s/it]
68%	[██████]	857/1261 [56:42<28:30, 4.23s/it]
68%	[██████]	858/1261 [56:46<28:10, 4.19s/it]
68%	[██████]	859/1261 [56:51<28:47, 4.30s/it]
68%	[██████]	860/1261 [56:55<28:43, 4.30s/it]
68%	[██████]	861/1261 [56:59<28:16, 4.24s/it]
68%	[██████]	862/1261 [57:03<27:57, 4.20s/it]
68%	[██████]	863/1261 [57:07<27:41, 4.17s/it]
69%	[██████]	864/1261 [57:11<27:30, 4.16s/it]
69%	[██████]	865/1261 [57:16<27:21, 4.15s/it]
69%	[██████]	866/1261 [57:20<27:12, 4.13s/it]
69%	[██████]	867/1261 [57:24<27:05, 4.12s/it]
69%	[██████]	868/1261 [57:28<26:57, 4.12s/it]
69%	[██████]	869/1261 [57:32<26:48, 4.10s/it]
69%	[██████]	870/1261 [57:36<26:45, 4.11s/it]
69%	[██████]	871/1261 [57:40<26:39, 4.10s/it]
69%	[██████]	872/1261 [57:44<26:41, 4.12s/it]
69%	[██████]	873/1261 [57:49<26:54, 4.16s/it]
69%	[██████]	874/1261 [57:53<26:40, 4.14s/it]
69%	[██████]	875/1261 [57:57<26:32, 4.12s/it]
69%	[██████]	876/1261 [58:01<26:25, 4.12s/it]
70%	[██████]	877/1261 [58:05<26:23, 4.12s/it]
70%	[██████]	878/1261 [58:09<26:19, 4.12s/it]
70%	[██████]	879/1261 [58:13<26:30, 4.16s/it]
70%	[██████]	880/1261 [58:18<26:46, 4.22s/it]
70%	[██████]	881/1261 [58:22<26:42, 4.22s/it]
70%	[██████]	882/1261 [58:26<26:35, 4.21s/it]
70%	[██████]	883/1261 [58:30<26:17, 4.17s/it]
70%	[██████]	884/1261 [58:34<25:44, 4.10s/it]
70%	[██████]	885/1261 [58:38<25:26, 4.06s/it]
70%	[██████]	886/1261 [58:42<25:01, 4.00s/it]
70%	[██████]	887/1261 [58:46<24:48, 3.98s/it]
70%	[██████]	888/1261 [58:50<24:53, 4.00s/it]

70%	[REDACTED]	889/1261 [58:54<24:36, 3.97s/it]
71%	[REDACTED]	890/1261 [58:58<24:39, 3.99s/it]
71%	[REDACTED]	891/1261 [59:02<24:35, 3.99s/it]
71%	[REDACTED]	892/1261 [59:06<24:24, 3.97s/it]
71%	[REDACTED]	893/1261 [59:10<24:52, 4.06s/it]
71%	[REDACTED]	894/1261 [59:14<24:31, 4.01s/it]
71%	[REDACTED]	895/1261 [59:18<24:26, 4.01s/it]
71%	[REDACTED]	896/1261 [59:22<24:10, 3.97s/it]
71%	[REDACTED]	897/1261 [59:26<24:16, 4.00s/it]
71%	[REDACTED]	898/1261 [59:30<24:02, 3.97s/it]
71%	[REDACTED]	899/1261 [59:34<23:44, 3.94s/it]
71%	[REDACTED]	900/1261 [59:37<23:34, 3.92s/it]
71%	[REDACTED]	901/1261 [59:41<23:23, 3.90s/it]
72%	[REDACTED]	902/1261 [59:45<23:18, 3.90s/it]
72%	[REDACTED]	903/1261 [59:49<23:16, 3.90s/it]
72%	[REDACTED]	904/1261 [59:53<23:11, 3.90s/it]
72%	[REDACTED]	905/1261 [59:57<23:02, 3.88s/it]
72%	[REDACTED]	906/1261 [1:00:01<22:54, 3.87s/it]
72%	[REDACTED]	907/1261 [1:00:05<22:50, 3.87s/it]
72%	[REDACTED]	908/1261 [1:00:08<22:47, 3.87s/it]
72%	[REDACTED]	909/1261 [1:00:12<22:45, 3.88s/it]
72%	[REDACTED]	910/1261 [1:00:16<22:41, 3.88s/it]
72%	[REDACTED]	911/1261 [1:00:20<22:39, 3.89s/it]
72%	[REDACTED]	912/1261 [1:00:24<22:36, 3.89s/it]
72%	[REDACTED]	913/1261 [1:00:28<22:31, 3.88s/it]
72%	[REDACTED]	914/1261 [1:00:32<22:25, 3.88s/it]
73%	[REDACTED]	915/1261 [1:00:36<22:20, 3.87s/it]
73%	[REDACTED]	916/1261 [1:00:40<22:18, 3.88s/it]
73%	[REDACTED]	917/1261 [1:00:43<22:10, 3.87s/it]
73%	[REDACTED]	918/1261 [1:00:47<22:09, 3.88s/it]
73%	[REDACTED]	919/1261 [1:00:51<22:08, 3.88s/it]
73%	[REDACTED]	920/1261 [1:00:55<22:13, 3.91s/it]
73%	[REDACTED]	921/1261 [1:00:59<22:05, 3.90s/it]
73%	[REDACTED]	922/1261 [1:01:03<21:58, 3.89s/it]
73%	[REDACTED]	923/1261 [1:01:07<21:58, 3.90s/it]
73%	[REDACTED]	924/1261 [1:01:11<21:51, 3.89s/it]
73%	[REDACTED]	925/1261 [1:01:15<21:47, 3.89s/it]
73%	[REDACTED]	926/1261 [1:01:18<21:45, 3.90s/it]
74%	[REDACTED]	927/1261 [1:01:22<21:41, 3.90s/it]
74%	[REDACTED]	928/1261 [1:01:26<21:34, 3.89s/it]
74%	[REDACTED]	929/1261 [1:01:30<21:33, 3.90s/it]
74%	[REDACTED]	930/1261 [1:01:34<21:28, 3.89s/it]
74%	[REDACTED]	931/1261 [1:01:38<21:22, 3.88s/it]
74%	[REDACTED]	932/1261 [1:01:42<21:18, 3.89s/it]
74%	[REDACTED]	933/1261 [1:01:46<21:12, 3.88s/it]
74%	[REDACTED]	934/1261 [1:01:49<21:02, 3.86s/it]
74%	[REDACTED]	935/1261 [1:01:53<21:01, 3.87s/it]
74%	[REDACTED]	936/1261 [1:01:57<20:57, 3.87s/it]
74%	[REDACTED]	937/1261 [1:02:01<21:02, 3.90s/it]
74%	[REDACTED]	938/1261 [1:02:05<20:58, 3.90s/it]
74%	[REDACTED]	939/1261 [1:02:09<20:56, 3.90s/it]
75%	[REDACTED]	940/1261 [1:02:13<20:54, 3.91s/it]
75%	[REDACTED]	941/1261 [1:02:17<20:46, 3.89s/it]
75%	[REDACTED]	942/1261 [1:02:21<20:42, 3.90s/it]
75%	[REDACTED]	943/1261 [1:02:25<20:41, 3.90s/it]
75%	[REDACTED]	944/1261 [1:02:28<20:35, 3.90s/it]
75%	[REDACTED]	945/1261 [1:02:32<20:29, 3.89s/it]
75%	[REDACTED]	946/1261 [1:02:36<20:26, 3.89s/it]
75%	[REDACTED]	947/1261 [1:02:40<20:20, 3.89s/it]
75%	[REDACTED]	948/1261 [1:02:44<20:18, 3.89s/it]
75%	[REDACTED]	949/1261 [1:02:48<20:14, 3.89s/it]
75%	[REDACTED]	950/1261 [1:02:52<20:12, 3.90s/it]
75%	[REDACTED]	951/1261 [1:02:56<20:03, 3.88s/it]
75%	[REDACTED]	952/1261 [1:03:00<19:58, 3.88s/it]
76%	[REDACTED]	953/1261 [1:03:03<19:52, 3.87s/it]
76%	[REDACTED]	954/1261 [1:03:07<19:48, 3.87s/it]
76%	[REDACTED]	955/1261 [1:03:11<19:52, 3.90s/it]
76%	[REDACTED]	956/1261 [1:03:15<19:43, 3.88s/it]
76%	[REDACTED]	957/1261 [1:03:19<19:45, 3.90s/it]
76%	[REDACTED]	958/1261 [1:03:23<19:40, 3.90s/it]
76%	[REDACTED]	959/1261 [1:03:27<19:34, 3.89s/it]
76%	[REDACTED]	960/1261 [1:03:31<19:36, 3.91s/it]
76%	[REDACTED]	961/1261 [1:03:35<19:31, 3.91s/it]
76%	[REDACTED]	962/1261 [1:03:39<19:27, 3.91s/it]
76%	[REDACTED]	963/1261 [1:03:42<19:20, 3.89s/it]
76%	[REDACTED]	964/1261 [1:03:46<19:15, 3.89s/it]
77%	[REDACTED]	965/1261 [1:03:50<19:08, 3.88s/it]
77%	[REDACTED]	966/1261 [1:03:54<19:05, 3.88s/it]
77%	[REDACTED]	967/1261 [1:03:58<19:06, 3.90s/it]
77%	[REDACTED]	968/1261 [1:04:02<19:00, 3.89s/it]
77%	[REDACTED]	969/1261 [1:04:06<18:59, 3.90s/it]
77%	[REDACTED]	970/1261 [1:04:10<18:53, 3.89s/it]
77%	[REDACTED]	971/1261 [1:04:14<18:48, 3.89s/it]
77%	[REDACTED]	972/1261 [1:04:17<18:47, 3.90s/it]
77%	[REDACTED]	973/1261 [1:04:21<18:44, 3.90s/it]
77%	[REDACTED]	974/1261 [1:04:25<18:39, 3.90s/it]
77%	[REDACTED]	975/1261 [1:04:29<18:35, 3.90s/it]
77%	[REDACTED]	976/1261 [1:04:33<18:25, 3.88s/it]
77%	[REDACTED]	977/1261 [1:04:37<18:21, 3.88s/it]

78%	[REDACTED]	978/1261 [1:04:41<18:17, 3.88s/it]
78%	[REDACTED]	979/1261 [1:04:45<18:13, 3.88s/it]
78%	[REDACTED]	980/1261 [1:04:49<18:09, 3.88s/it]
78%	[REDACTED]	981/1261 [1:04:52<18:06, 3.88s/it]
78%	[REDACTED]	982/1261 [1:04:56<18:00, 3.87s/it]
78%	[REDACTED]	983/1261 [1:05:00<17:58, 3.88s/it]
78%	[REDACTED]	984/1261 [1:05:04<17:54, 3.88s/it]
78%	[REDACTED]	985/1261 [1:05:08<17:50, 3.88s/it]
78%	[REDACTED]	986/1261 [1:05:12<17:45, 3.88s/it]
78%	[REDACTED]	987/1261 [1:05:16<17:41, 3.87s/it]
78%	[REDACTED]	988/1261 [1:05:20<17:37, 3.87s/it]
78%	[REDACTED]	989/1261 [1:05:23<17:37, 3.89s/it]
79%	[REDACTED]	990/1261 [1:05:27<17:33, 3.89s/it]
79%	[REDACTED]	991/1261 [1:05:31<17:29, 3.89s/it]
79%	[REDACTED]	992/1261 [1:05:35<17:26, 3.89s/it]
79%	[REDACTED]	993/1261 [1:05:39<17:22, 3.89s/it]
79%	[REDACTED]	994/1261 [1:05:43<17:20, 3.90s/it]
79%	[REDACTED]	995/1261 [1:05:47<17:14, 3.89s/it]
79%	[REDACTED]	996/1261 [1:05:51<17:10, 3.89s/it]
79%	[REDACTED]	997/1261 [1:05:55<17:03, 3.88s/it]
79%	[REDACTED]	998/1261 [1:05:58<16:58, 3.87s/it]
79%	[REDACTED]	999/1261 [1:06:02<16:54, 3.87s/it]
79%	[REDACTED]	1000/1261 [1:06:06<16:52, 3.88s/it]
79%	[REDACTED]	1001/1261 [1:06:10<16:47, 3.87s/it]
79%	[REDACTED]	1002/1261 [1:06:14<16:43, 3.87s/it]
80%	[REDACTED]	1003/1261 [1:06:18<16:39, 3.87s/it]
80%	[REDACTED]	1004/1261 [1:06:22<16:39, 3.89s/it]
80%	[REDACTED]	1005/1261 [1:06:26<16:34, 3.88s/it]
80%	[REDACTED]	1006/1261 [1:06:29<16:34, 3.90s/it]
80%	[REDACTED]	1007/1261 [1:06:33<16:34, 3.91s/it]
80%	[REDACTED]	1008/1261 [1:06:37<16:31, 3.92s/it]
80%	[REDACTED]	1009/1261 [1:06:41<16:31, 3.93s/it]
80%	[REDACTED]	1010/1261 [1:06:45<16:32, 3.96s/it]
80%	[REDACTED]	1011/1261 [1:06:49<16:23, 3.93s/it]
80%	[REDACTED]	1012/1261 [1:06:53<16:25, 3.96s/it]
80%	[REDACTED]	1013/1261 [1:06:57<16:18, 3.95s/it]
80%	[REDACTED]	1014/1261 [1:07:01<16:17, 3.96s/it]
80%	[REDACTED]	1015/1261 [1:07:05<16:18, 3.98s/it]
81%	[REDACTED]	1016/1261 [1:07:09<16:33, 4.06s/it]
81%	[REDACTED]	1017/1261 [1:07:13<16:18, 4.01s/it]
81%	[REDACTED]	1018/1261 [1:07:17<16:07, 3.98s/it]
81%	[REDACTED]	1019/1261 [1:07:21<16:01, 3.97s/it]
81%	[REDACTED]	1020/1261 [1:07:25<15:54, 3.96s/it]
81%	[REDACTED]	1021/1261 [1:07:29<15:49, 3.96s/it]
81%	[REDACTED]	1022/1261 [1:07:33<15:41, 3.94s/it]
81%	[REDACTED]	1023/1261 [1:07:37<15:36, 3.94s/it]
81%	[REDACTED]	1024/1261 [1:07:41<15:30, 3.93s/it]
81%	[REDACTED]	1025/1261 [1:07:45<15:30, 3.94s/it]
81%	[REDACTED]	1026/1261 [1:07:49<15:24, 3.93s/it]
81%	[REDACTED]	1027/1261 [1:07:53<15:16, 3.92s/it]
82%	[REDACTED]	1028/1261 [1:07:56<15:12, 3.92s/it]
82%	[REDACTED]	1029/1261 [1:08:00<15:05, 3.91s/it]
82%	[REDACTED]	1030/1261 [1:08:04<15:02, 3.91s/it]
82%	[REDACTED]	1031/1261 [1:08:08<14:58, 3.91s/it]
82%	[REDACTED]	1032/1261 [1:08:12<14:55, 3.91s/it]
82%	[REDACTED]	1033/1261 [1:08:16<14:52, 3.91s/it]
82%	[REDACTED]	1034/1261 [1:08:20<14:48, 3.91s/it]
82%	[REDACTED]	1035/1261 [1:08:24<14:46, 3.92s/it]
82%	[REDACTED]	1036/1261 [1:08:28<14:40, 3.91s/it]
82%	[REDACTED]	1037/1261 [1:08:32<14:35, 3.91s/it]
82%	[REDACTED]	1038/1261 [1:08:36<14:31, 3.91s/it]
82%	[REDACTED]	1039/1261 [1:08:39<14:26, 3.90s/it]
82%	[REDACTED]	1040/1261 [1:08:43<14:22, 3.90s/it]
83%	[REDACTED]	1041/1261 [1:08:47<14:17, 3.90s/it]
83%	[REDACTED]	1042/1261 [1:08:51<14:12, 3.89s/it]
83%	[REDACTED]	1043/1261 [1:08:55<14:09, 3.90s/it]
83%	[REDACTED]	1044/1261 [1:08:59<14:05, 3.90s/it]
83%	[REDACTED]	1045/1261 [1:09:03<14:00, 3.89s/it]
83%	[REDACTED]	1046/1261 [1:09:07<13:56, 3.89s/it]
83%	[REDACTED]	1047/1261 [1:09:11<13:52, 3.89s/it]
83%	[REDACTED]	1048/1261 [1:09:15<13:49, 3.89s/it]
83%	[REDACTED]	1049/1261 [1:09:18<13:47, 3.90s/it]
83%	[REDACTED]	1050/1261 [1:09:22<13:42, 3.90s/it]
83%	[REDACTED]	1051/1261 [1:09:26<13:38, 3.90s/it]
83%	[REDACTED]	1052/1261 [1:09:30<13:33, 3.89s/it]
84%	[REDACTED]	1053/1261 [1:09:34<13:28, 3.89s/it]
84%	[REDACTED]	1054/1261 [1:09:38<13:24, 3.89s/it]
84%	[REDACTED]	1055/1261 [1:09:42<13:22, 3.89s/it]
84%	[REDACTED]	1056/1261 [1:09:46<13:18, 3.89s/it]
84%	[REDACTED]	1057/1261 [1:09:50<13:13, 3.89s/it]
84%	[REDACTED]	1058/1261 [1:09:53<13:11, 3.90s/it]
84%	[REDACTED]	1059/1261 [1:09:57<13:05, 3.89s/it]
84%	[REDACTED]	1060/1261 [1:10:01<13:00, 3.89s/it]
84%	[REDACTED]	1061/1261 [1:10:05<12:57, 3.89s/it]
84%	[REDACTED]	1062/1261 [1:10:09<12:54, 3.89s/it]
84%	[REDACTED]	1063/1261 [1:10:13<12:49, 3.88s/it]
84%	[REDACTED]	1064/1261 [1:10:17<12:46, 3.89s/it]
84%	[REDACTED]	1065/1261 [1:10:21<12:43, 3.89s/it]
85%	[REDACTED]	1066/1261 [1:10:25<12:38, 3.89s/it]

85%	[REDACTED]	1067/1261 [1:10:28<12:35, 3.89s/it]
85%	[REDACTED]	1068/1261 [1:10:32<12:31, 3.89s/it]
85%	[REDACTED]	1069/1261 [1:10:36<12:26, 3.89s/it]
85%	[REDACTED]	1070/1261 [1:10:40<12:21, 3.88s/it]
85%	[REDACTED]	1071/1261 [1:10:44<12:17, 3.88s/it]
85%	[REDACTED]	1072/1261 [1:10:48<12:15, 3.89s/it]
85%	[REDACTED]	1073/1261 [1:10:52<12:12, 3.89s/it]
85%	[REDACTED]	1074/1261 [1:10:56<12:09, 3.90s/it]
85%	[REDACTED]	1075/1261 [1:11:00<12:07, 3.91s/it]
85%	[REDACTED]	1076/1261 [1:11:03<12:00, 3.90s/it]
85%	[REDACTED]	1077/1261 [1:11:07<11:57, 3.90s/it]
85%	[REDACTED]	1078/1261 [1:11:11<11:51, 3.89s/it]
86%	[REDACTED]	1079/1261 [1:11:15<11:48, 3.89s/it]
86%	[REDACTED]	1080/1261 [1:11:19<11:47, 3.91s/it]
86%	[REDACTED]	1081/1261 [1:11:23<11:41, 3.90s/it]
86%	[REDACTED]	1082/1261 [1:11:27<11:37, 3.90s/it]
86%	[REDACTED]	1083/1261 [1:11:31<11:33, 3.90s/it]
86%	[REDACTED]	1084/1261 [1:11:35<11:31, 3.91s/it]
86%	[REDACTED]	1085/1261 [1:11:39<11:26, 3.90s/it]
86%	[REDACTED]	1086/1261 [1:11:43<11:23, 3.91s/it]
86%	[REDACTED]	1087/1261 [1:11:46<11:20, 3.91s/it]
86%	[REDACTED]	1088/1261 [1:11:50<11:15, 3.91s/it]
86%	[REDACTED]	1089/1261 [1:11:54<11:12, 3.91s/it]
86%	[REDACTED]	1090/1261 [1:11:58<11:10, 3.92s/it]
87%	[REDACTED]	1091/1261 [1:12:02<11:06, 3.92s/it]
87%	[REDACTED]	1092/1261 [1:12:06<10:59, 3.90s/it]
87%	[REDACTED]	1093/1261 [1:12:10<10:55, 3.90s/it]
87%	[REDACTED]	1094/1261 [1:12:14<10:51, 3.90s/it]
87%	[REDACTED]	1095/1261 [1:12:18<10:46, 3.90s/it]
87%	[REDACTED]	1096/1261 [1:12:22<10:43, 3.90s/it]
87%	[REDACTED]	1097/1261 [1:12:25<10:38, 3.90s/it]
87%	[REDACTED]	1098/1261 [1:12:29<10:35, 3.90s/it]
87%	[REDACTED]	1099/1261 [1:12:33<10:31, 3.90s/it]
87%	[REDACTED]	1100/1261 [1:12:37<10:26, 3.89s/it]
87%	[REDACTED]	1101/1261 [1:12:41<10:25, 3.91s/it]
87%	[REDACTED]	1102/1261 [1:12:45<10:19, 3.90s/it]
87%	[REDACTED]	1103/1261 [1:12:49<10:15, 3.90s/it]
88%	[REDACTED]	1104/1261 [1:12:53<10:11, 3.90s/it]
88%	[REDACTED]	1105/1261 [1:12:57<10:08, 3.90s/it]
88%	[REDACTED]	1106/1261 [1:13:01<10:03, 3.90s/it]
88%	[REDACTED]	1107/1261 [1:13:04<10:00, 3.90s/it]
88%	[REDACTED]	1108/1261 [1:13:08<09:55, 3.89s/it]
88%	[REDACTED]	1109/1261 [1:13:12<09:51, 3.89s/it]
88%	[REDACTED]	1110/1261 [1:13:16<09:46, 3.88s/it]
88%	[REDACTED]	1111/1261 [1:13:20<09:43, 3.89s/it]
88%	[REDACTED]	1112/1261 [1:13:24<09:40, 3.90s/it]
88%	[REDACTED]	1113/1261 [1:13:28<09:36, 3.90s/it]
88%	[REDACTED]	1114/1261 [1:13:32<09:33, 3.90s/it]
88%	[REDACTED]	1115/1261 [1:13:36<09:28, 3.89s/it]
89%	[REDACTED]	1116/1261 [1:13:40<09:25, 3.90s/it]
89%	[REDACTED]	1117/1261 [1:13:43<09:22, 3.91s/it]
89%	[REDACTED]	1118/1261 [1:13:47<09:17, 3.90s/it]
89%	[REDACTED]	1119/1261 [1:13:51<09:13, 3.90s/it]
89%	[REDACTED]	1120/1261 [1:13:55<09:09, 3.90s/it]
89%	[REDACTED]	1121/1261 [1:13:59<09:07, 3.91s/it]
89%	[REDACTED]	1122/1261 [1:14:03<09:02, 3.90s/it]
89%	[REDACTED]	1123/1261 [1:14:07<08:57, 3.89s/it]
89%	[REDACTED]	1124/1261 [1:14:11<08:53, 3.89s/it]
89%	[REDACTED]	1125/1261 [1:14:15<08:48, 3.89s/it]
89%	[REDACTED]	1126/1261 [1:14:18<08:43, 3.88s/it]
89%	[REDACTED]	1127/1261 [1:14:22<08:39, 3.88s/it]
89%	[REDACTED]	1128/1261 [1:14:26<08:36, 3.88s/it]
90%	[REDACTED]	1129/1261 [1:14:30<08:32, 3.89s/it]
90%	[REDACTED]	1130/1261 [1:14:34<08:30, 3.89s/it]
90%	[REDACTED]	1131/1261 [1:14:38<08:25, 3.89s/it]
90%	[REDACTED]	1132/1261 [1:14:42<08:22, 3.89s/it]
90%	[REDACTED]	1133/1261 [1:14:46<08:18, 3.89s/it]
90%	[REDACTED]	1134/1261 [1:14:50<08:17, 3.92s/it]
90%	[REDACTED]	1135/1261 [1:14:54<08:13, 3.91s/it]
90%	[REDACTED]	1136/1261 [1:14:57<08:08, 3.91s/it]
90%	[REDACTED]	1137/1261 [1:15:01<08:03, 3.90s/it]
90%	[REDACTED]	1138/1261 [1:15:05<07:59, 3.90s/it]
90%	[REDACTED]	1139/1261 [1:15:09<07:55, 3.90s/it]
90%	[REDACTED]	1140/1261 [1:15:13<07:50, 3.89s/it]
90%	[REDACTED]	1141/1261 [1:15:17<07:46, 3.89s/it]
91%	[REDACTED]	1142/1261 [1:15:21<07:44, 3.90s/it]
91%	[REDACTED]	1143/1261 [1:15:25<07:40, 3.90s/it]
91%	[REDACTED]	1144/1261 [1:15:29<07:36, 3.90s/it]
91%	[REDACTED]	1145/1261 [1:15:32<07:31, 3.89s/it]
91%	[REDACTED]	1146/1261 [1:15:36<07:27, 3.89s/it]
91%	[REDACTED]	1147/1261 [1:15:40<07:24, 3.89s/it]
91%	[REDACTED]	1148/1261 [1:15:44<07:19, 3.89s/it]
91%	[REDACTED]	1149/1261 [1:15:48<07:15, 3.89s/it]
91%	[REDACTED]	1150/1261 [1:15:52<07:13, 3.90s/it]
91%	[REDACTED]	1151/1261 [1:15:56<07:08, 3.90s/it]
91%	[REDACTED]	1152/1261 [1:16:00<07:04, 3.89s/it]
91%	[REDACTED]	1153/1261 [1:16:04<07:01, 3.90s/it]
92%	[REDACTED]	1154/1261 [1:16:08<06:56, 3.89s/it]
92%	[REDACTED]	1155/1261 [1:16:11<06:52, 3.89s/it]

92%	[REDACTED]	1156/1261 [1:16:15<06:51, 3.91s/it]
92%	[REDACTED]	1157/1261 [1:16:19<06:46, 3.91s/it]
92%	[REDACTED]	1158/1261 [1:16:23<06:41, 3.90s/it]
92%	[REDACTED]	1159/1261 [1:16:27<06:37, 3.89s/it]
92%	[REDACTED]	1160/1261 [1:16:31<06:34, 3.91s/it]
92%	[REDACTED]	1161/1261 [1:16:35<06:29, 3.90s/it]
92%	[REDACTED]	1162/1261 [1:16:39<06:24, 3.89s/it]
92%	[REDACTED]	1163/1261 [1:16:43<06:21, 3.89s/it]
92%	[REDACTED]	1164/1261 [1:16:46<06:16, 3.88s/it]
92%	[REDACTED]	1165/1261 [1:16:50<06:12, 3.88s/it]
92%	[REDACTED]	1166/1261 [1:16:54<06:10, 3.90s/it]
93%	[REDACTED]	1167/1261 [1:16:58<06:05, 3.89s/it]
93%	[REDACTED]	1168/1261 [1:17:02<06:02, 3.90s/it]
93%	[REDACTED]	1169/1261 [1:17:06<05:58, 3.90s/it]
93%	[REDACTED]	1170/1261 [1:17:10<05:54, 3.90s/it]
93%	[REDACTED]	1171/1261 [1:17:14<05:51, 3.91s/it]
93%	[REDACTED]	1172/1261 [1:17:18<05:47, 3.91s/it]
93%	[REDACTED]	1173/1261 [1:17:22<05:42, 3.90s/it]
93%	[REDACTED]	1174/1261 [1:17:25<05:38, 3.89s/it]
93%	[REDACTED]	1175/1261 [1:17:29<05:34, 3.90s/it]
93%	[REDACTED]	1176/1261 [1:17:33<05:30, 3.89s/it]
93%	[REDACTED]	1177/1261 [1:17:37<05:26, 3.89s/it]
93%	[REDACTED]	1178/1261 [1:17:41<05:22, 3.89s/it]
93%	[REDACTED]	1179/1261 [1:17:45<05:18, 3.89s/it]
94%	[REDACTED]	1180/1261 [1:17:49<05:15, 3.90s/it]
94%	[REDACTED]	1181/1261 [1:17:53<05:11, 3.90s/it]
94%	[REDACTED]	1182/1261 [1:17:57<05:07, 3.90s/it]
94%	[REDACTED]	1183/1261 [1:18:00<05:03, 3.89s/it]
94%	[REDACTED]	1184/1261 [1:18:04<04:59, 3.90s/it]
94%	[REDACTED]	1185/1261 [1:18:08<04:56, 3.89s/it]
94%	[REDACTED]	1186/1261 [1:18:12<04:53, 3.91s/it]
94%	[REDACTED]	1187/1261 [1:18:16<04:48, 3.90s/it]
94%	[REDACTED]	1188/1261 [1:18:20<04:45, 3.91s/it]
94%	[REDACTED]	1189/1261 [1:18:24<04:41, 3.91s/it]
94%	[REDACTED]	1190/1261 [1:18:28<04:36, 3.90s/it]
94%	[REDACTED]	1191/1261 [1:18:32<04:33, 3.91s/it]
95%	[REDACTED]	1192/1261 [1:18:36<04:29, 3.91s/it]
95%	[REDACTED]	1193/1261 [1:18:40<04:24, 3.90s/it]
95%	[REDACTED]	1194/1261 [1:18:43<04:20, 3.89s/it]
95%	[REDACTED]	1195/1261 [1:18:47<04:18, 3.92s/it]
95%	[REDACTED]	1196/1261 [1:18:51<04:14, 3.92s/it]
95%	[REDACTED]	1197/1261 [1:18:55<04:10, 3.91s/it]
95%	[REDACTED]	1198/1261 [1:18:59<04:06, 3.91s/it]
95%	[REDACTED]	1199/1261 [1:19:03<04:01, 3.90s/it]
95%	[REDACTED]	1200/1261 [1:19:07<03:57, 3.90s/it]
95%	[REDACTED]	1201/1261 [1:19:11<03:54, 3.91s/it]
95%	[REDACTED]	1202/1261 [1:19:15<03:50, 3.91s/it]
95%	[REDACTED]	1203/1261 [1:19:19<03:46, 3.90s/it]
95%	[REDACTED]	1204/1261 [1:19:23<03:42, 3.90s/it]
96%	[REDACTED]	1205/1261 [1:19:26<03:38, 3.90s/it]
96%	[REDACTED]	1206/1261 [1:19:30<03:34, 3.90s/it]
96%	[REDACTED]	1207/1261 [1:19:34<03:30, 3.89s/it]
96%	[REDACTED]	1208/1261 [1:19:38<03:25, 3.88s/it]
96%	[REDACTED]	1209/1261 [1:19:42<03:22, 3.89s/it]
96%	[REDACTED]	1210/1261 [1:19:46<03:18, 3.90s/it]
96%	[REDACTED]	1211/1261 [1:19:50<03:15, 3.91s/it]
96%	[REDACTED]	1212/1261 [1:19:54<03:11, 3.90s/it]
96%	[REDACTED]	1213/1261 [1:19:58<03:06, 3.89s/it]
96%	[REDACTED]	1214/1261 [1:20:01<03:03, 3.90s/it]
96%	[REDACTED]	1215/1261 [1:20:05<02:59, 3.90s/it]
96%	[REDACTED]	1216/1261 [1:20:09<02:55, 3.89s/it]
97%	[REDACTED]	1217/1261 [1:20:13<02:51, 3.90s/it]
97%	[REDACTED]	1218/1261 [1:20:17<02:47, 3.89s/it]
97%	[REDACTED]	1219/1261 [1:20:21<02:43, 3.89s/it]
97%	[REDACTED]	1220/1261 [1:20:25<02:39, 3.89s/it]
97%	[REDACTED]	1221/1261 [1:20:29<02:35, 3.89s/it]
97%	[REDACTED]	1222/1261 [1:20:33<02:32, 3.90s/it]
97%	[REDACTED]	1223/1261 [1:20:36<02:27, 3.88s/it]
97%	[REDACTED]	1224/1261 [1:20:40<02:23, 3.88s/it]
97%	[REDACTED]	1225/1261 [1:20:44<02:19, 3.88s/it]
97%	[REDACTED]	1226/1261 [1:20:48<02:15, 3.88s/it]
97%	[REDACTED]	1227/1261 [1:20:52<02:11, 3.88s/it]
97%	[REDACTED]	1228/1261 [1:20:56<02:08, 3.90s/it]
97%	[REDACTED]	1229/1261 [1:21:00<02:04, 3.89s/it]
98%	[REDACTED]	1230/1261 [1:21:04<02:00, 3.90s/it]
98%	[REDACTED]	1231/1261 [1:21:08<01:56, 3.90s/it]
98%	[REDACTED]	1232/1261 [1:21:11<01:52, 3.89s/it]
98%	[REDACTED]	1233/1261 [1:21:15<01:48, 3.89s/it]
98%	[REDACTED]	1234/1261 [1:21:19<01:45, 3.90s/it]
98%	[REDACTED]	1235/1261 [1:21:23<01:41, 3.91s/it]
98%	[REDACTED]	1236/1261 [1:21:27<01:37, 3.91s/it]
98%	[REDACTED]	1237/1261 [1:21:31<01:33, 3.92s/it]
98%	[REDACTED]	1238/1261 [1:21:35<01:30, 3.91s/it]
98%	[REDACTED]	1239/1261 [1:21:39<01:25, 3.91s/it]
98%	[REDACTED]	1240/1261 [1:21:43<01:21, 3.89s/it]
98%	[REDACTED]	1241/1261 [1:21:47<01:17, 3.89s/it]
98%	[REDACTED]	1242/1261 [1:21:50<01:13, 3.89s/it]
99%	[REDACTED]	1243/1261 [1:21:54<01:10, 3.89s/it]
99%	[REDACTED]	1244/1261 [1:21:58<01:06, 3.92s/it]

```
99% |██████████| 1245/1261 [1:22:02<01:02, 3.91s/it]
99% |██████████| 1246/1261 [1:22:06<00:58, 3.90s/it]
99% |██████████| 1247/1261 [1:22:10<00:54, 3.89s/it]
99% |██████████| 1248/1261 [1:22:14<00:50, 3.90s/it]
99% |██████████| 1249/1261 [1:22:18<00:46, 3.91s/it]
99% |██████████| 1250/1261 [1:22:22<00:42, 3.90s/it]
99% |██████████| 1251/1261 [1:22:26<00:39, 3.91s/it]
99% |██████████| 1252/1261 [1:22:30<00:35, 3.91s/it]
99% |██████████| 1253/1261 [1:22:34<00:31, 3.92s/it]
99% |██████████| 1254/1261 [1:22:37<00:27, 3.91s/it]
100% |██████████| 1255/1261 [1:22:41<00:23, 3.91s/it]
100% |██████████| 1256/1261 [1:22:45<00:19, 3.91s/it]
100% |██████████| 1257/1261 [1:22:49<00:15, 3.90s/it]
100% |██████████| 1258/1261 [1:22:53<00:11, 3.90s/it]
100% |██████████| 1259/1261 [1:22:57<00:07, 3.90s/it]
100% |██████████| 1260/1261 [1:23:01<00:03, 3.89s/it]
```

```
[MoviePy] Done.  
[MoviePy] >>> Video ready: ./project_video_out_3.mp4
```

```
CPU times: user 1h 22min 21s, sys: 49 s, total: 1h 23min 10s  
Wall time: 1h 23min 2s
```

```
In [72]: HTML("""  
<video width="960" height="540" controls>  
    <source src="{0}">4  
</video>  
""").format(out_vid))
```

```
Out[72]:
```



```
▶ 0:00 / 0:50
```

```
In [76]: ystart = 400  
ystop = 656  
scales = [1, 1.5]  
box_buffer = []  
heat_threshold = 5  
nb_frames = 5
```

```
In [77]: ## To speed up the testing process you may want to try your pipeline on a shorter subclip of the video
## 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 subclip
## You may also uncomment the following line for a subclip of the first 5 seconds
##clip1 = VideoFileClip("test_videos/solidWhiteRight.mp4").subclip(0,5)
in_vid = './project_video.mp4'
out_vid = './project_video_out_2.mp4'

clip = VideoFileClip(in_vid)
video_clip = clip.fl_image(process_frame)
%time video_clip.write_videofile(out_vid, audio=False)
```

```
[MoviePy] >>> Building video ./project_video_out_2.mp4
[MoviePy] Writing video ./project_video_out_2.mp4
```

0%	0/1261 [00:00<?, ?it/s]
0%	1/1261 [00:15<5:19:51, 15.23s/it]
0%	2/1261 [00:30<5:22:50, 15.39s/it]
0%	3/1261 [00:46<5:22:06, 15.36s/it]
0%	4/1261 [01:01<5:21:44, 15.36s/it]
0%	5/1261 [01:16<5:20:08, 15.29s/it]
0%	6/1261 [01:31<5:19:10, 15.26s/it]
1%	7/1261 [01:47<5:19:10, 15.27s/it]
1%	8/1261 [02:02<5:18:37, 15.26s/it]
1%	9/1261 [02:17<5:17:57, 15.24s/it]
1%	10/1261 [02:32<5:17:59, 15.25s/it]
1%	11/1261 [02:48<5:17:15, 15.23s/it]
1%	12/1261 [03:03<5:16:41, 15.21s/it]
1%	13/1261 [03:18<5:16:46, 15.23s/it]
1%	14/1261 [03:33<5:17:06, 15.26s/it]
1%	15/1261 [03:49<5:16:28, 15.24s/it]
1%	16/1261 [04:04<5:16:03, 15.23s/it]
1%	17/1261 [04:19<5:15:25, 15.21s/it]
1%	18/1261 [04:34<5:15:31, 15.23s/it]
2%	19/1261 [04:49<5:14:42, 15.20s/it]
2%	20/1261 [05:05<5:15:13, 15.24s/it]
2%	21/1261 [05:20<5:14:27, 15.22s/it]
2%	22/1261 [05:35<5:15:26, 15.28s/it]
2%	23/1261 [05:50<5:14:38, 15.25s/it]
2%	24/1261 [06:06<5:14:24, 15.25s/it]
2%	25/1261 [06:21<5:13:25, 15.21s/it]
2%	26/1261 [06:36<5:13:04, 15.21s/it]
2%	27/1261 [06:51<5:12:43, 15.21s/it]
2%	28/1261 [07:07<5:12:47, 15.22s/it]
2%	29/1261 [07:22<5:12:33, 15.22s/it]
2%	30/1261 [07:37<5:12:45, 15.24s/it]
2%	31/1261 [07:52<5:12:24, 15.24s/it]
3%	32/1261 [08:07<5:11:39, 15.22s/it]
3%	33/1261 [08:23<5:11:01, 15.20s/it]
3%	34/1261 [08:38<5:10:56, 15.21s/it]
3%	35/1261 [08:53<5:10:55, 15.22s/it]
3%	36/1261 [09:09<5:12:52, 15.32s/it]
3%	37/1261 [09:24<5:11:58, 15.29s/it]
3%	38/1261 [09:39<5:11:42, 15.29s/it]
3%	39/1261 [09:55<5:12:36, 15.35s/it]
3%	40/1261 [10:10<5:11:53, 15.33s/it]
3%	41/1261 [10:25<5:09:47, 15.24s/it]
3%	42/1261 [10:40<5:08:36, 15.19s/it]
3%	43/1261 [10:55<5:09:50, 15.26s/it]
3%	44/1261 [11:11<5:08:52, 15.23s/it]
4%	45/1261 [11:26<5:07:37, 15.18s/it]
4%	46/1261 [11:41<5:06:52, 15.15s/it]
4%	47/1261 [11:56<5:06:10, 15.13s/it]
4%	48/1261 [12:11<5:05:41, 15.12s/it]
4%	49/1261 [12:26<5:04:46, 15.09s/it]
4%	50/1261 [12:41<5:04:17, 15.08s/it]
4%	51/1261 [12:56<5:03:54, 15.07s/it]
4%	52/1261 [13:11<5:03:28, 15.06s/it]
4%	53/1261 [13:26<5:03:17, 15.06s/it]
4%	54/1261 [13:41<5:02:55, 15.06s/it]
4%	55/1261 [13:56<5:02:24, 15.05s/it]
4%	56/1261 [14:11<5:02:11, 15.05s/it]
5%	57/1261 [14:26<5:02:08, 15.06s/it]
5%	58/1261 [14:41<5:02:24, 15.08s/it]
5%	59/1261 [14:56<5:01:15, 15.04s/it]
5%	60/1261 [15:11<5:00:40, 15.02s/it]
5%	61/1261 [15:26<4:59:43, 14.99s/it]
5%	62/1261 [15:41<5:00:41, 15.05s/it]
5%	63/1261 [15:56<4:59:44, 15.01s/it]
5%	64/1261 [16:12<5:00:50, 15.08s/it]
5%	65/1261 [16:27<4:59:26, 15.02s/it]
5%	66/1261 [16:41<4:58:32, 14.99s/it]
5%	67/1261 [16:56<4:57:52, 14.97s/it]
5%	68/1261 [17:11<4:57:38, 14.97s/it]
5%	69/1261 [17:26<4:56:57, 14.95s/it]
6%	70/1261 [17:41<4:56:23, 14.93s/it]
6%	71/1261 [17:56<4:56:28, 14.95s/it]
6%	72/1261 [18:11<4:56:18, 14.95s/it]
6%	73/1261 [18:26<4:56:01, 14.95s/it]
6%	74/1261 [18:41<4:55:05, 14.92s/it]
6%	75/1261 [18:56<4:55:03, 14.93s/it]
6%	76/1261 [19:11<4:54:47, 14.93s/it]
6%	77/1261 [19:26<4:54:35, 14.93s/it]
6%	78/1261 [19:41<4:55:11, 14.97s/it]
6%	79/1261 [19:56<4:55:05, 14.98s/it]
6%	80/1261 [20:11<4:54:50, 14.98s/it]
6%	81/1261 [20:26<4:53:49, 14.94s/it]
7%	82/1261 [20:41<4:54:18, 14.98s/it]
7%	83/1261 [20:56<4:54:31, 15.00s/it]
7%	84/1261 [21:11<4:54:09, 15.00s/it]
7%	85/1261 [21:26<4:53:27, 14.97s/it]
7%	86/1261 [21:41<4:53:22, 14.98s/it]
7%	87/1261 [21:56<4:53:10, 14.98s/it]

7% [88/1261 [22:11<4:52:38, 14.97s/it]
7% [89/1261 [22:25<4:52:14, 14.96s/it]
7% [90/1261 [22:40<4:51:28, 14.93s/it]
7% [91/1261 [22:55<4:52:00, 14.97s/it]
7% [92/1261 [23:10<4:51:26, 14.96s/it]
7% [93/1261 [23:25<4:51:01, 14.95s/it]
7% [94/1261 [23:40<4:50:35, 14.94s/it]
8% [95/1261 [23:55<4:50:19, 14.94s/it]
8% [96/1261 [24:10<4:50:04, 14.94s/it]
8% [97/1261 [24:25<4:49:56, 14.95s/it]
8% [98/1261 [24:40<4:49:19, 14.93s/it]
8% [99/1261 [24:55<4:48:55, 14.92s/it]
8% [100/1261 [25:10<4:49:31, 14.96s/it]
8% [101/1261 [25:25<4:49:00, 14.95s/it]
8% [102/1261 [25:40<4:49:54, 15.01s/it]
8% [103/1261 [25:55<4:49:10, 14.98s/it]
8% [104/1261 [26:10<4:48:44, 14.97s/it]
8% [105/1261 [26:25<4:47:54, 14.94s/it]
8% [106/1261 [26:40<4:47:29, 14.93s/it]
8% [107/1261 [26:54<4:46:57, 14.92s/it]
9% [108/1261 [27:10<4:48:27, 15.01s/it]
9% [109/1261 [27:25<4:47:37, 14.98s/it]
9% [110/1261 [27:40<4:47:13, 14.97s/it]
9% [111/1261 [27:55<4:47:03, 14.98s/it]
9% [112/1261 [28:10<4:46:58, 14.99s/it]
9% [113/1261 [28:24<4:46:20, 14.97s/it]
9% [114/1261 [28:40<4:46:29, 14.99s/it]
9% [115/1261 [28:54<4:46:06, 14.98s/it]
9% [116/1261 [29:09<4:45:37, 14.97s/it]
9% [117/1261 [29:24<4:44:45, 14.93s/it]
9% [118/1261 [29:39<4:44:31, 14.94s/it]
9% [119/1261 [29:54<4:44:14, 14.93s/it]
10% [120/1261 [30:09<4:44:04, 14.94s/it]
10% [121/1261 [30:24<4:43:51, 14.94s/it]
10% [122/1261 [30:39<4:45:25, 15.04s/it]
10% [123/1261 [30:54<4:45:05, 15.03s/it]
10% [124/1261 [31:09<4:44:36, 15.02s/it]
10% [125/1261 [31:24<4:43:25, 14.97s/it]
10% [126/1261 [31:39<4:43:03, 14.96s/it]
10% [127/1261 [31:54<4:42:35, 14.95s/it]
10% [128/1261 [32:09<4:42:41, 14.97s/it]
10% [129/1261 [32:24<4:41:59, 14.95s/it]
10% [130/1261 [32:39<4:41:42, 14.94s/it]
10% [131/1261 [32:54<4:42:20, 14.99s/it]
10% [132/1261 [33:09<4:41:47, 14.98s/it]
11% [133/1261 [33:24<4:41:07, 14.95s/it]
11% [134/1261 [33:39<4:40:43, 14.95s/it]
11% [135/1261 [33:54<4:41:10, 14.98s/it]
11% [136/1261 [34:09<4:40:59, 14.99s/it]
11% [137/1261 [34:24<4:40:08, 14.95s/it]
11% [138/1261 [34:39<4:39:43, 14.95s/it]
11% [139/1261 [34:53<4:39:10, 14.93s/it]
11% [140/1261 [35:09<4:39:40, 14.97s/it]
11% [141/1261 [35:24<4:39:24, 14.97s/it]
11% [142/1261 [35:39<4:39:42, 15.00s/it]
11% [143/1261 [35:53<4:38:56, 14.97s/it]
11% [144/1261 [36:08<4:38:44, 14.97s/it]
11% [145/1261 [36:23<4:38:43, 14.99s/it]
12% [146/1261 [36:38<4:38:08, 14.97s/it]
12% [147/1261 [36:54<4:38:46, 15.01s/it]
12% [148/1261 [37:09<4:40:00, 15.09s/it]
12% [149/1261 [37:24<4:39:02, 15.06s/it]
12% [150/1261 [37:39<4:38:04, 15.02s/it]
12% [151/1261 [37:54<4:37:25, 15.00s/it]
12% [152/1261 [38:09<4:36:29, 14.96s/it]
12% [153/1261 [38:23<4:35:47, 14.93s/it]
12% [154/1261 [38:39<4:36:26, 14.98s/it]
12% [155/1261 [38:54<4:37:15, 15.04s/it]
12% [156/1261 [39:09<4:36:21, 15.01s/it]
12% [157/1261 [39:24<4:35:44, 14.99s/it]
13% [158/1261 [39:39<4:35:26, 14.98s/it]
13% [159/1261 [39:53<4:34:36, 14.95s/it]
13% [160/1261 [40:08<4:34:56, 14.98s/it]
13% [161/1261 [40:23<4:34:38, 14.98s/it]
13% [162/1261 [40:38<4:33:59, 14.96s/it]
13% [163/1261 [40:53<4:33:46, 14.96s/it]
13% [164/1261 [41:08<4:33:16, 14.95s/it]
13% [165/1261 [41:23<4:33:09, 14.95s/it]
13% [166/1261 [41:38<4:32:22, 14.92s/it]
13% [167/1261 [41:53<4:31:58, 14.92s/it]
13% [168/1261 [42:08<4:31:45, 14.92s/it]
13% [169/1261 [42:23<4:31:22, 14.91s/it]
13% [170/1261 [42:38<4:30:46, 14.89s/it]
14% [171/1261 [42:53<4:30:47, 14.91s/it]
14% [172/1261 [43:08<4:31:03, 14.93s/it]
14% [173/1261 [43:22<4:30:32, 14.92s/it]
14% [174/1261 [43:37<4:30:20, 14.92s/it]
14% [175/1261 [43:52<4:30:46, 14.96s/it]
14% [176/1261 [44:08<4:31:25, 15.01s/it]

14%	177/1261	[44:22<4:30:52, 14.99s/it]
14%	178/1261	[44:40<4:44:33, 15.77s/it]
14%	179/1261	[44:57<4:52:45, 16.23s/it]
14%	180/1261	[45:14<4:53:39, 16.30s/it]
14%	181/1261	[45:31<5:00:07, 16.67s/it]
14%	182/1261	[45:49<5:04:32, 16.93s/it]
15%	183/1261	[46:04<4:56:15, 16.49s/it]
15%	184/1261	[46:20<4:49:08, 16.11s/it]
15%	185/1261	[46:35<4:45:56, 15.94s/it]
15%	186/1261	[46:50<4:42:18, 15.76s/it]
15%	187/1261	[47:06<4:41:17, 15.71s/it]
15%	188/1261	[47:21<4:39:20, 15.62s/it]
15%	189/1261	[47:37<4:38:03, 15.56s/it]
15%	190/1261	[47:53<4:40:18, 15.70s/it]
15%	191/1261	[48:09<4:41:18, 15.77s/it]
15%	192/1261	[48:26<4:47:04, 16.11s/it]
15%	193/1261	[48:43<4:51:46, 16.39s/it]
15%	194/1261	[49:00<4:53:58, 16.53s/it]
15%	195/1261	[49:15<4:49:19, 16.28s/it]
16%	196/1261	[49:31<4:47:43, 16.21s/it]
16%	197/1261	[49:47<4:45:17, 16.09s/it]
16%	198/1261	[50:03<4:42:36, 15.95s/it]
16%	199/1261	[50:19<4:42:23, 15.95s/it]
16%	200/1261	[50:36<4:47:14, 16.24s/it]
16%	201/1261	[50:52<4:44:33, 16.11s/it]
16%	202/1261	[51:08<4:44:22, 16.11s/it]
16%	203/1261	[51:24<4:43:10, 16.06s/it]
16%	204/1261	[51:41<4:47:31, 16.32s/it]
16%	205/1261	[51:56<4:42:13, 16.04s/it]
16%	206/1261	[52:13<4:46:06, 16.27s/it]
16%	207/1261	[52:29<4:46:04, 16.29s/it]
16%	208/1261	[52:45<4:42:44, 16.11s/it]
17%	209/1261	[53:00<4:39:58, 15.97s/it]
17%	210/1261	[53:16<4:37:22, 15.84s/it]
17%	211/1261	[53:32<4:36:00, 15.77s/it]
17%	212/1261	[53:47<4:34:50, 15.72s/it]
17%	213/1261	[54:03<4:34:14, 15.70s/it]
17%	214/1261	[54:18<4:34:02, 15.70s/it]
17%	215/1261	[54:34<4:33:45, 15.70s/it]
17%	216/1261	[54:50<4:32:57, 15.67s/it]
17%	217/1261	[55:05<4:32:29, 15.66s/it]
17%	218/1261	[55:21<4:32:19, 15.67s/it]
17%	219/1261	[55:37<4:32:00, 15.66s/it]
17%	220/1261	[55:52<4:31:46, 15.66s/it]
18%	221/1261	[56:08<4:31:37, 15.67s/it]
18%	222/1261	[56:24<4:31:18, 15.67s/it]
18%	223/1261	[56:39<4:31:05, 15.67s/it]
18%	224/1261	[56:55<4:30:36, 15.66s/it]
18%	225/1261	[57:11<4:30:14, 15.65s/it]
18%	226/1261	[57:26<4:29:55, 15.65s/it]
18%	227/1261	[57:42<4:29:32, 15.64s/it]
18%	228/1261	[57:58<4:29:01, 15.63s/it]
18%	229/1261	[58:13<4:28:46, 15.63s/it]
18%	230/1261	[58:29<4:28:42, 15.64s/it]
18%	231/1261	[58:44<4:28:19, 15.63s/it]
18%	232/1261	[59:00<4:28:09, 15.64s/it]
18%	233/1261	[59:16<4:27:58, 15.64s/it]
19%	234/1261	[59:31<4:27:25, 15.62s/it]
19%	235/1261	[59:47<4:27:07, 15.62s/it]
19%	236/1261	[1:00:03<4:27:14, 15.64s/it]
19%	237/1261	[1:00:18<4:26:59, 15.64s/it]
19%	238/1261	[1:00:34<4:26:52, 15.65s/it]
19%	239/1261	[1:00:50<4:27:01, 15.68s/it]
19%	240/1261	[1:01:05<4:26:21, 15.65s/it]
19%	241/1261	[1:01:21<4:25:58, 15.65s/it]
19%	242/1261	[1:01:37<4:25:34, 15.64s/it]
19%	243/1261	[1:01:52<4:25:35, 15.65s/it]
19%	244/1261	[1:02:08<4:25:43, 15.68s/it]
19%	245/1261	[1:02:24<4:25:03, 15.65s/it]
20%	246/1261	[1:02:39<4:22:09, 15.50s/it]
20%	247/1261	[1:02:54<4:19:19, 15.34s/it]
20%	248/1261	[1:03:09<4:17:52, 15.27s/it]
20%	249/1261	[1:03:24<4:16:24, 15.20s/it]
20%	250/1261	[1:03:39<4:15:22, 15.16s/it]
20%	251/1261	[1:03:54<4:14:52, 15.14s/it]
20%	252/1261	[1:04:09<4:14:18, 15.12s/it]
20%	253/1261	[1:04:24<4:13:21, 15.08s/it]
20%	254/1261	[1:04:39<4:12:34, 15.05s/it]
20%	255/1261	[1:04:54<4:11:52, 15.02s/it]
20%	256/1261	[1:05:09<4:12:02, 15.05s/it]
20%	257/1261	[1:05:24<4:12:10, 15.07s/it]
20%	258/1261	[1:05:39<4:12:06, 15.08s/it]
21%	259/1261	[1:05:54<4:11:34, 15.06s/it]
21%	260/1261	[1:06:09<4:11:03, 15.05s/it]
21%	261/1261	[1:06:25<4:11:17, 15.08s/it]
21%	262/1261	[1:06:40<4:10:52, 15.07s/it]
21%	263/1261	[1:06:55<4:10:34, 15.06s/it]
21%	264/1261	[1:07:10<4:10:15, 15.06s/it]
21%	265/1261	[1:07:25<4:09:33, 15.03s/it]

21% [266/1261 [1:07:40<4:09:43, 15.06s/it]
21% [267/1261 [1:07:55<4:09:10, 15.04s/it]
21% [268/1261 [1:08:10<4:08:36, 15.02s/it]
21% [269/1261 [1:08:25<4:07:47, 14.99s/it]
21% [270/1261 [1:08:40<4:07:33, 14.99s/it]
21% [271/1261 [1:08:55<4:07:12, 14.98s/it]
22% [272/1261 [1:09:10<4:07:19, 15.01s/it]
22% [273/1261 [1:09:25<4:07:00, 15.00s/it]
22% [274/1261 [1:09:40<4:06:33, 14.99s/it]
22% [275/1261 [1:09:55<4:07:15, 15.05s/it]
22% [276/1261 [1:10:10<4:06:39, 15.02s/it]
22% [277/1261 [1:10:25<4:05:52, 14.99s/it]
22% [278/1261 [1:10:40<4:05:36, 14.99s/it]
22% [279/1261 [1:10:55<4:06:06, 15.04s/it]
22% [280/1261 [1:11:10<4:05:27, 15.01s/it]
22% [281/1261 [1:11:25<4:05:07, 15.01s/it]
22% [282/1261 [1:11:40<4:04:40, 14.99s/it]
22% [283/1261 [1:11:55<4:04:02, 14.97s/it]
23% [284/1261 [1:12:10<4:03:54, 14.98s/it]
23% [285/1261 [1:12:25<4:03:44, 14.98s/it]
23% [286/1261 [1:12:40<4:03:29, 14.98s/it]
23% [287/1261 [1:12:55<4:02:57, 14.97s/it]
23% [288/1261 [1:13:09<4:02:33, 14.96s/it]
23% [289/1261 [1:13:24<4:02:15, 14.95s/it]
23% [290/1261 [1:13:39<4:02:22, 14.98s/it]
23% [291/1261 [1:13:54<4:01:51, 14.96s/it]
23% [292/1261 [1:14:09<4:01:30, 14.95s/it]
23% [293/1261 [1:14:24<4:01:06, 14.94s/it]
23% [294/1261 [1:14:39<4:00:35, 14.93s/it]
23% [295/1261 [1:14:54<4:00:37, 14.95s/it]
23% [296/1261 [1:15:09<4:00:04, 14.93s/it]
24% [297/1261 [1:15:24<4:00:14, 14.95s/it]
24% [298/1261 [1:15:39<4:00:23, 14.98s/it]
24% [299/1261 [1:15:54<4:00:10, 14.98s/it]
24% [300/1261 [1:16:09<3:59:52, 14.98s/it]
24% [301/1261 [1:16:24<3:59:30, 14.97s/it]
24% [302/1261 [1:16:39<3:59:10, 14.96s/it]
24% [303/1261 [1:16:54<3:58:22, 14.93s/it]
24% [304/1261 [1:17:09<3:58:29, 14.95s/it]
24% [305/1261 [1:17:24<3:58:08, 14.95s/it]
24% [306/1261 [1:17:39<3:57:51, 14.94s/it]
24% [307/1261 [1:17:54<3:57:25, 14.93s/it]
24% [308/1261 [1:18:09<3:57:20, 14.94s/it]
25% [309/1261 [1:18:23<3:56:48, 14.93s/it]
25% [310/1261 [1:18:38<3:57:18, 14.97s/it]
25% [311/1261 [1:18:54<3:57:47, 15.02s/it]
25% [312/1261 [1:19:09<3:57:24, 15.01s/it]
25% [313/1261 [1:19:24<3:56:48, 14.99s/it]
25% [314/1261 [1:19:38<3:56:16, 14.97s/it]
25% [315/1261 [1:19:53<3:55:58, 14.97s/it]
25% [316/1261 [1:20:08<3:55:17, 14.94s/it]
25% [317/1261 [1:20:23<3:55:12, 14.95s/it]
25% [318/1261 [1:20:38<3:55:04, 14.96s/it]
25% [319/1261 [1:20:53<3:55:01, 14.97s/it]
25% [320/1261 [1:21:08<3:56:06, 15.05s/it]
25% [321/1261 [1:21:23<3:55:25, 15.03s/it]
26% [322/1261 [1:21:38<3:54:58, 15.01s/it]
26% [323/1261 [1:21:53<3:54:25, 14.99s/it]
26% [324/1261 [1:22:08<3:53:58, 14.98s/it]
26% [325/1261 [1:22:23<3:54:19, 15.02s/it]
26% [326/1261 [1:22:38<3:54:05, 15.02s/it]
26% [327/1261 [1:22:53<3:53:22, 14.99s/it]
26% [328/1261 [1:23:08<3:52:46, 14.97s/it]
26% [329/1261 [1:23:23<3:52:56, 15.00s/it]
26% [330/1261 [1:23:38<3:52:29, 14.98s/it]
26% [331/1261 [1:23:53<3:52:14, 14.98s/it]
26% [332/1261 [1:24:08<3:52:08, 14.99s/it]
26% [333/1261 [1:24:23<3:52:20, 15.02s/it]
26% [334/1261 [1:24:38<3:51:47, 15.00s/it]
27% [335/1261 [1:24:53<3:51:02, 14.97s/it]
27% [336/1261 [1:25:08<3:51:01, 14.99s/it]
27% [337/1261 [1:25:23<3:50:27, 14.96s/it]
27% [338/1261 [1:25:38<3:50:13, 14.97s/it]
27% [339/1261 [1:25:53<3:49:43, 14.95s/it]
27% [340/1261 [1:26:08<3:49:53, 14.98s/it]
27% [341/1261 [1:26:23<3:49:27, 14.96s/it]
27% [342/1261 [1:26:38<3:51:06, 15.09s/it]
27% [343/1261 [1:26:53<3:49:58, 15.03s/it]
27% [344/1261 [1:27:08<3:49:06, 14.99s/it]
27% [345/1261 [1:27:23<3:48:37, 14.98s/it]
27% [346/1261 [1:27:38<3:48:15, 14.97s/it]
28% [347/1261 [1:27:53<3:47:57, 14.96s/it]
28% [348/1261 [1:28:08<3:47:22, 14.94s/it]
28% [349/1261 [1:28:23<3:47:03, 14.94s/it]
28% [350/1261 [1:28:38<3:47:56, 15.01s/it]
28% [351/1261 [1:28:53<3:47:28, 15.00s/it]
28% [352/1261 [1:29:08<3:46:48, 14.97s/it]
28% [353/1261 [1:29:23<3:46:25, 14.96s/it]
28% [354/1261 [1:29:38<3:46:07, 14.96s/it]

28%	355/1261	[1:29:53<3:45:41, 14.95s/it]
28%	356/1261	[1:30:08<3:45:20, 14.94s/it]
28%	357/1261	[1:30:23<3:45:57, 15.00s/it]
28%	358/1261	[1:30:38<3:45:34, 14.99s/it]
28%	359/1261	[1:30:53<3:45:04, 14.97s/it]
29%	360/1261	[1:31:08<3:44:50, 14.97s/it]
29%	361/1261	[1:31:23<3:44:10, 14.95s/it]
29%	362/1261	[1:31:37<3:43:46, 14.93s/it]
29%	363/1261	[1:31:52<3:43:22, 14.93s/it]
29%	364/1261	[1:32:07<3:43:48, 14.97s/it]
29%	365/1261	[1:32:22<3:43:44, 14.98s/it]
29%	366/1261	[1:32:37<3:43:27, 14.98s/it]
29%	367/1261	[1:32:52<3:43:14, 14.98s/it]
29%	368/1261	[1:33:07<3:42:41, 14.96s/it]
29%	369/1261	[1:33:22<3:42:37, 14.97s/it]
29%	370/1261	[1:33:38<3:43:06, 15.02s/it]
29%	371/1261	[1:33:52<3:42:30, 15.00s/it]
30%	372/1261	[1:34:07<3:42:10, 14.99s/it]
30%	373/1261	[1:34:22<3:41:55, 14.99s/it]
30%	374/1261	[1:34:37<3:41:50, 15.01s/it]
30%	375/1261	[1:34:52<3:41:39, 15.01s/it]
30%	376/1261	[1:35:07<3:41:07, 14.99s/it]
30%	377/1261	[1:35:22<3:40:11, 14.95s/it]
30%	378/1261	[1:35:37<3:40:32, 14.99s/it]
30%	379/1261	[1:35:52<3:40:26, 15.00s/it]
30%	380/1261	[1:36:07<3:40:00, 14.98s/it]
30%	381/1261	[1:36:22<3:39:36, 14.97s/it]
30%	382/1261	[1:36:37<3:39:49, 15.01s/it]
30%	383/1261	[1:36:52<3:39:14, 14.98s/it]
30%	384/1261	[1:37:07<3:39:28, 15.02s/it]
31%	385/1261	[1:37:22<3:39:04, 15.01s/it]
31%	386/1261	[1:37:37<3:39:05, 15.02s/it]
31%	387/1261	[1:37:52<3:38:22, 14.99s/it]
31%	388/1261	[1:38:07<3:38:52, 15.04s/it]
31%	389/1261	[1:38:22<3:38:01, 15.00s/it]
31%	390/1261	[1:38:37<3:37:40, 15.00s/it]
31%	391/1261	[1:38:52<3:37:04, 14.97s/it]
31%	392/1261	[1:39:07<3:36:58, 14.98s/it]
31%	393/1261	[1:39:22<3:36:30, 14.97s/it]
31%	394/1261	[1:39:37<3:36:49, 15.01s/it]
31%	395/1261	[1:39:52<3:36:43, 15.02s/it]
31%	396/1261	[1:40:07<3:36:47, 15.04s/it]
31%	397/1261	[1:40:22<3:36:13, 15.02s/it]
32%	398/1261	[1:40:38<3:36:32, 15.06s/it]
32%	399/1261	[1:40:53<3:36:07, 15.04s/it]
32%	400/1261	[1:41:08<3:35:26, 15.01s/it]
32%	401/1261	[1:41:23<3:36:33, 15.11s/it]
32%	402/1261	[1:41:38<3:37:33, 15.20s/it]
32%	403/1261	[1:41:53<3:37:12, 15.19s/it]
32%	404/1261	[1:42:08<3:35:54, 15.12s/it]
32%	405/1261	[1:42:23<3:35:21, 15.10s/it]
32%	406/1261	[1:42:38<3:34:33, 15.06s/it]
32%	407/1261	[1:42:53<3:33:52, 15.03s/it]
32%	408/1261	[1:43:08<3:33:25, 15.01s/it]
32%	409/1261	[1:43:23<3:33:42, 15.05s/it]
33%	410/1261	[1:43:38<3:32:55, 15.01s/it]
33%	411/1261	[1:43:53<3:32:27, 15.00s/it]
33%	412/1261	[1:44:08<3:32:05, 14.99s/it]
33%	413/1261	[1:44:23<3:31:34, 14.97s/it]
33%	414/1261	[1:44:38<3:32:06, 15.03s/it]
33%	415/1261	[1:44:53<3:32:10, 15.05s/it]
33%	416/1261	[1:45:08<3:31:22, 15.01s/it]
33%	417/1261	[1:45:24<3:31:30, 15.04s/it]
33%	418/1261	[1:45:39<3:31:10, 15.03s/it]
33%	419/1261	[1:45:54<3:30:49, 15.02s/it]
33%	420/1261	[1:46:09<3:30:32, 15.02s/it]
33%	421/1261	[1:46:23<3:29:47, 14.99s/it]
33%	422/1261	[1:46:38<3:29:18, 14.97s/it]
34%	423/1261	[1:46:53<3:28:48, 14.95s/it]
34%	424/1261	[1:47:08<3:28:36, 14.95s/it]
34%	425/1261	[1:47:23<3:27:47, 14.91s/it]
34%	426/1261	[1:47:38<3:28:15, 14.96s/it]
34%	427/1261	[1:47:53<3:27:55, 14.96s/it]
34%	428/1261	[1:48:08<3:27:36, 14.95s/it]
34%	429/1261	[1:48:23<3:27:28, 14.96s/it]
34%	430/1261	[1:48:38<3:28:00, 15.02s/it]
34%	431/1261	[1:48:53<3:27:35, 15.01s/it]
34%	432/1261	[1:49:08<3:27:08, 14.99s/it]
34%	433/1261	[1:49:23<3:26:44, 14.98s/it]
34%	434/1261	[1:49:38<3:26:09, 14.96s/it]
34%	435/1261	[1:49:53<3:25:50, 14.95s/it]
35%	436/1261	[1:50:08<3:26:06, 14.99s/it]
35%	437/1261	[1:50:23<3:25:35, 14.97s/it]
35%	438/1261	[1:50:38<3:26:16, 15.04s/it]
35%	439/1261	[1:50:53<3:25:31, 15.00s/it]
35%	440/1261	[1:51:08<3:25:15, 15.00s/it]
35%	441/1261	[1:51:23<3:24:44, 14.98s/it]
35%	442/1261	[1:51:38<3:24:48, 15.00s/it]
35%	443/1261	[1:51:53<3:24:01, 14.97s/it]

35%	444/1261	[1:52:08<3:23:42, 14.96s/it]
35%	445/1261	[1:52:23<3:23:10, 14.94s/it]
35%	446/1261	[1:52:38<3:23:24, 14.98s/it]
35%	447/1261	[1:52:53<3:22:55, 14.96s/it]
36%	448/1261	[1:53:08<3:22:36, 14.95s/it]
36%	449/1261	[1:53:23<3:22:17, 14.95s/it]
36%	450/1261	[1:53:38<3:22:15, 14.96s/it]
36%	451/1261	[1:53:53<3:22:01, 14.96s/it]
36%	452/1261	[1:54:08<3:21:48, 14.97s/it]
36%	453/1261	[1:54:22<3:21:30, 14.96s/it]
36%	454/1261	[1:54:37<3:20:55, 14.94s/it]
36%	455/1261	[1:54:52<3:21:13, 14.98s/it]
36%	456/1261	[1:55:07<3:21:03, 14.99s/it]
36%	457/1261	[1:55:22<3:20:30, 14.96s/it]
36%	458/1261	[1:55:37<3:20:23, 14.97s/it]
36%	459/1261	[1:55:52<3:19:45, 14.95s/it]
36%	460/1261	[1:56:07<3:20:14, 15.00s/it]
37%	461/1261	[1:56:22<3:19:36, 14.97s/it]
37%	462/1261	[1:56:37<3:18:56, 14.94s/it]
37%	463/1261	[1:56:52<3:18:38, 14.94s/it]
37%	464/1261	[1:57:07<3:18:34, 14.95s/it]
37%	465/1261	[1:57:22<3:18:11, 14.94s/it]
37%	466/1261	[1:57:37<3:17:55, 14.94s/it]
37%	467/1261	[1:57:52<3:17:38, 14.93s/it]
37%	468/1261	[1:58:07<3:17:41, 14.96s/it]
37%	469/1261	[1:58:22<3:18:07, 15.01s/it]
37%	470/1261	[1:58:37<3:17:13, 14.96s/it]
37%	471/1261	[1:58:52<3:16:52, 14.95s/it]
37%	472/1261	[1:59:07<3:16:49, 14.97s/it]
38%	473/1261	[1:59:22<3:16:36, 14.97s/it]
38%	474/1261	[1:59:37<3:16:22, 14.97s/it]
38%	475/1261	[1:59:52<3:15:57, 14.96s/it]
38%	476/1261	[2:00:07<3:16:18, 15.00s/it]
38%	477/1261	[2:00:22<3:15:41, 14.98s/it]
38%	478/1261	[2:00:37<3:15:19, 14.97s/it]
38%	479/1261	[2:00:52<3:15:12, 14.98s/it]
38%	480/1261	[2:01:07<3:16:01, 15.06s/it]
38%	481/1261	[2:01:22<3:15:57, 15.07s/it]
38%	482/1261	[2:01:37<3:15:01, 15.02s/it]
38%	483/1261	[2:01:52<3:14:26, 15.00s/it]
38%	484/1261	[2:02:07<3:14:48, 15.04s/it]
38%	485/1261	[2:02:22<3:13:59, 15.00s/it]
39%	486/1261	[2:02:37<3:13:27, 14.98s/it]
39%	487/1261	[2:02:52<3:12:59, 14.96s/it]
39%	488/1261	[2:03:07<3:13:20, 15.01s/it]
39%	489/1261	[2:03:22<3:12:43, 14.98s/it]
39%	490/1261	[2:03:37<3:12:30, 14.98s/it]
39%	491/1261	[2:03:52<3:11:52, 14.95s/it]
39%	492/1261	[2:04:07<3:11:33, 14.95s/it]
39%	493/1261	[2:04:22<3:12:00, 15.00s/it]
39%	494/1261	[2:04:37<3:11:25, 14.97s/it]
39%	495/1261	[2:04:52<3:11:23, 14.99s/it]
39%	496/1261	[2:05:06<3:10:51, 14.97s/it]
39%	497/1261	[2:05:22<3:10:49, 14.99s/it]
39%	498/1261	[2:05:36<3:10:18, 14.97s/it]
40%	499/1261	[2:05:52<3:10:30, 15.00s/it]
40%	500/1261	[2:06:06<3:09:46, 14.96s/it]
40%	501/1261	[2:06:21<3:09:19, 14.95s/it]
40%	502/1261	[2:06:36<3:09:10, 14.95s/it]
40%	503/1261	[2:06:51<3:09:16, 14.98s/it]
40%	504/1261	[2:07:06<3:09:11, 15.00s/it]
40%	505/1261	[2:07:21<3:08:45, 14.98s/it]
40%	506/1261	[2:07:36<3:08:20, 14.97s/it]
40%	507/1261	[2:07:51<3:07:47, 14.94s/it]
40%	508/1261	[2:08:06<3:07:41, 14.96s/it]
40%	509/1261	[2:08:21<3:07:27, 14.96s/it]
40%	510/1261	[2:08:36<3:07:19, 14.97s/it]
41%	511/1261	[2:08:51<3:06:59, 14.96s/it]
41%	512/1261	[2:09:06<3:06:50, 14.97s/it]
41%	513/1261	[2:09:21<3:06:08, 14.93s/it]
41%	514/1261	[2:09:36<3:05:51, 14.93s/it]
41%	515/1261	[2:09:51<3:05:59, 14.96s/it]
41%	516/1261	[2:10:06<3:06:13, 15.00s/it]
41%	517/1261	[2:10:21<3:05:38, 14.97s/it]
41%	518/1261	[2:10:36<3:05:14, 14.96s/it]
41%	519/1261	[2:10:51<3:04:55, 14.95s/it]
41%	520/1261	[2:11:06<3:04:19, 14.92s/it]
41%	521/1261	[2:11:20<3:04:10, 14.93s/it]
41%	522/1261	[2:11:35<3:03:57, 14.94s/it]
41%	523/1261	[2:11:50<3:03:34, 14.92s/it]
42%	524/1261	[2:12:05<3:03:42, 14.96s/it]
42%	525/1261	[2:12:20<3:03:33, 14.96s/it]
42%	526/1261	[2:12:35<3:03:18, 14.96s/it]
42%	527/1261	[2:12:50<3:02:58, 14.96s/it]
42%	528/1261	[2:13:05<3:02:49, 14.97s/it]
42%	529/1261	[2:13:20<3:02:25, 14.95s/it]
42%	530/1261	[2:13:35<3:02:17, 14.96s/it]
42%	531/1261	[2:13:50<3:01:51, 14.95s/it]
42%	532/1261	[2:14:05<3:01:32, 14.94s/it]

42%	[██████████]	533/1261 [2:14:20<3:01:10, 14.93s/it]
42%	[██████████]	534/1261 [2:14:35<3:00:52, 14.93s/it]
42%	[██████████]	535/1261 [2:14:50<3:00:36, 14.93s/it]
43%	[██████████]	536/1261 [2:15:05<3:00:17, 14.92s/it]
43%	[██████████]	537/1261 [2:15:20<3:00:13, 14.94s/it]
43%	[██████████]	538/1261 [2:15:34<2:59:53, 14.93s/it]
43%	[██████████]	539/1261 [2:15:50<3:00:19, 14.99s/it]
43%	[██████████]	540/1261 [2:16:05<2:59:54, 14.97s/it]
43%	[██████████]	541/1261 [2:16:19<2:59:32, 14.96s/it]
43%	[██████████]	542/1261 [2:16:35<3:00:13, 15.04s/it]
43%	[██████████]	543/1261 [2:16:50<2:59:40, 15.02s/it]
43%	[██████████]	544/1261 [2:17:05<2:59:08, 14.99s/it]
43%	[██████████]	545/1261 [2:17:20<2:58:45, 14.98s/it]
43%	[██████████]	546/1261 [2:17:35<2:58:47, 15.00s/it]
43%	[██████████]	547/1261 [2:17:50<2:58:13, 14.98s/it]
43%	[██████████]	548/1261 [2:18:04<2:57:53, 14.97s/it]
44%	[██████████]	549/1261 [2:18:19<2:57:32, 14.96s/it]
44%	[██████████]	550/1261 [2:18:34<2:57:22, 14.97s/it]
44%	[██████████]	551/1261 [2:18:49<2:57:08, 14.97s/it]
44%	[██████████]	552/1261 [2:19:04<2:56:57, 14.97s/it]
44%	[██████████]	553/1261 [2:19:19<2:56:36, 14.97s/it]
44%	[██████████]	554/1261 [2:19:34<2:56:53, 15.01s/it]
44%	[██████████]	555/1261 [2:19:49<2:56:23, 14.99s/it]
44%	[██████████]	556/1261 [2:20:04<2:55:57, 14.97s/it]
44%	[██████████]	557/1261 [2:20:19<2:55:50, 14.99s/it]
44%	[██████████]	558/1261 [2:20:34<2:55:46, 15.00s/it]
44%	[██████████]	559/1261 [2:20:50<2:56:52, 15.12s/it]
44%	[██████████]	560/1261 [2:21:05<2:56:07, 15.08s/it]
44%	[██████████]	561/1261 [2:21:20<2:55:31, 15.04s/it]
45%	[██████████]	562/1261 [2:21:35<2:55:14, 15.04s/it]
45%	[██████████]	563/1261 [2:21:50<2:54:28, 15.00s/it]
45%	[██████████]	564/1261 [2:22:05<2:54:15, 15.00s/it]
45%	[██████████]	565/1261 [2:22:20<2:53:45, 14.98s/it]
45%	[██████████]	566/1261 [2:22:35<2:53:30, 14.98s/it]
45%	[██████████]	567/1261 [2:22:49<2:53:04, 14.96s/it]
45%	[██████████]	568/1261 [2:23:04<2:52:32, 14.94s/it]
45%	[██████████]	569/1261 [2:23:19<2:52:55, 14.99s/it]
45%	[██████████]	570/1261 [2:23:34<2:52:23, 14.97s/it]
45%	[██████████]	571/1261 [2:23:49<2:52:09, 14.97s/it]
45%	[██████████]	572/1261 [2:24:04<2:51:45, 14.96s/it]
45%	[██████████]	573/1261 [2:24:19<2:51:31, 14.96s/it]
46%	[██████████]	574/1261 [2:24:34<2:51:12, 14.95s/it]
46%	[██████████]	575/1261 [2:24:51<2:55:40, 15.36s/it]
46%	[██████████]	576/1261 [2:25:07<2:58:02, 15.59s/it]
46%	[██████████]	577/1261 [2:25:23<2:58:56, 15.70s/it]
46%	[██████████]	578/1261 [2:25:38<2:59:24, 15.76s/it]
46%	[██████████]	579/1261 [2:25:54<2:58:42, 15.72s/it]
46%	[██████████]	580/1261 [2:26:10<2:57:49, 15.67s/it]
46%	[██████████]	581/1261 [2:26:25<2:57:11, 15.63s/it]
46%	[██████████]	582/1261 [2:26:41<2:56:42, 15.61s/it]
46%	[██████████]	583/1261 [2:26:56<2:56:46, 15.64s/it]
46%	[██████████]	584/1261 [2:27:12<2:56:08, 15.61s/it]
46%	[██████████]	585/1261 [2:27:28<2:55:32, 15.58s/it]
46%	[██████████]	586/1261 [2:27:43<2:55:34, 15.61s/it]
47%	[██████████]	587/1261 [2:27:59<2:55:14, 15.60s/it]
47%	[██████████]	588/1261 [2:28:14<2:54:40, 15.57s/it]
47%	[██████████]	589/1261 [2:28:30<2:53:59, 15.53s/it]
47%	[██████████]	590/1261 [2:28:45<2:53:40, 15.53s/it]
47%	[██████████]	591/1261 [2:29:01<2:53:46, 15.56s/it]
47%	[██████████]	592/1261 [2:29:16<2:53:13, 15.54s/it]
47%	[██████████]	593/1261 [2:29:32<2:53:07, 15.55s/it]
47%	[██████████]	594/1261 [2:29:48<2:53:01, 15.56s/it]
47%	[██████████]	595/1261 [2:30:03<2:52:30, 15.54s/it]
47%	[██████████]	596/1261 [2:30:19<2:52:15, 15.54s/it]
47%	[██████████]	597/1261 [2:30:34<2:52:11, 15.56s/it]
47%	[██████████]	598/1261 [2:30:50<2:51:59, 15.57s/it]
48%	[██████████]	599/1261 [2:31:05<2:51:44, 15.57s/it]
48%	[██████████]	600/1261 [2:31:21<2:51:29, 15.57s/it]
48%	[██████████]	601/1261 [2:31:37<2:51:39, 15.60s/it]
48%	[██████████]	602/1261 [2:31:52<2:51:41, 15.63s/it]
48%	[██████████]	603/1261 [2:32:08<2:51:06, 15.60s/it]
48%	[██████████]	604/1261 [2:32:23<2:50:38, 15.58s/it]
48%	[██████████]	605/1261 [2:32:39<2:50:07, 15.56s/it]
48%	[██████████]	606/1261 [2:32:54<2:49:58, 15.57s/it]
48%	[██████████]	607/1261 [2:33:10<2:49:31, 15.55s/it]
48%	[██████████]	608/1261 [2:33:25<2:48:58, 15.53s/it]
48%	[██████████]	609/1261 [2:33:41<2:48:59, 15.55s/it]
48%	[██████████]	610/1261 [2:33:57<2:48:47, 15.56s/it]
48%	[██████████]	611/1261 [2:34:12<2:48:25, 15.55s/it]
49%	[██████████]	612/1261 [2:34:28<2:48:13, 15.55s/it]
49%	[██████████]	613/1261 [2:34:43<2:47:52, 15.54s/it]
49%	[██████████]	614/1261 [2:34:59<2:48:13, 15.60s/it]
49%	[██████████]	615/1261 [2:35:14<2:46:52, 15.50s/it]
49%	[██████████]	616/1261 [2:35:29<2:45:02, 15.35s/it]
49%	[██████████]	617/1261 [2:35:44<2:43:59, 15.28s/it]
49%	[██████████]	618/1261 [2:35:59<2:43:12, 15.23s/it]
49%	[██████████]	619/1261 [2:36:14<2:42:13, 15.16s/it]
49%	[██████████]	620/1261 [2:36:29<2:41:25, 15.11s/it]
49%	[██████████]	621/1261 [2:36:45<2:41:08, 15.11s/it]

49%	[██████]	622/1261 [2:37:00<2:41:03, 15.12s/it]
49%	[██████]	623/1261 [2:37:15<2:40:40, 15.11s/it]
49%	[██████]	624/1261 [2:37:30<2:40:01, 15.07s/it]
50%	[██████]	625/1261 [2:37:45<2:39:33, 15.05s/it]
50%	[██████]	626/1261 [2:38:00<2:39:05, 15.03s/it]
50%	[██████]	627/1261 [2:38:15<2:39:00, 15.05s/it]
50%	[██████]	628/1261 [2:38:30<2:38:45, 15.05s/it]
50%	[██████]	629/1261 [2:38:45<2:38:39, 15.06s/it]
50%	[██████]	630/1261 [2:39:00<2:38:20, 15.06s/it]
50%	[██████]	631/1261 [2:39:15<2:38:11, 15.07s/it]
50%	[██████]	632/1261 [2:39:30<2:37:49, 15.06s/it]
50%	[██████]	633/1261 [2:39:45<2:37:37, 15.06s/it]
50%	[██████]	634/1261 [2:40:00<2:37:38, 15.09s/it]
50%	[██████]	635/1261 [2:40:15<2:36:52, 15.04s/it]
50%	[██████]	636/1261 [2:40:30<2:36:22, 15.01s/it]
51%	[██████]	637/1261 [2:40:45<2:36:03, 15.01s/it]
51%	[██████]	638/1261 [2:41:00<2:35:40, 14.99s/it]
51%	[██████]	639/1261 [2:41:15<2:35:14, 14.97s/it]
51%	[██████]	640/1261 [2:41:30<2:34:51, 14.96s/it]
51%	[██████]	641/1261 [2:41:45<2:34:39, 14.97s/it]
51%	[██████]	642/1261 [2:42:00<2:34:20, 14.96s/it]
51%	[██████]	643/1261 [2:42:15<2:34:03, 14.96s/it]
51%	[██████]	644/1261 [2:42:30<2:33:32, 14.93s/it]
51%	[██████]	645/1261 [2:42:45<2:33:05, 14.91s/it]
51%	[██████]	646/1261 [2:43:00<2:34:02, 15.03s/it]
51%	[██████]	647/1261 [2:43:15<2:33:18, 14.98s/it]
51%	[██████]	648/1261 [2:43:30<2:32:56, 14.97s/it]
51%	[██████]	649/1261 [2:43:45<2:33:17, 15.03s/it]
52%	[██████]	650/1261 [2:44:00<2:32:59, 15.02s/it]
52%	[██████]	651/1261 [2:44:15<2:32:33, 15.01s/it]
52%	[██████]	652/1261 [2:44:30<2:31:52, 14.96s/it]
52%	[██████]	653/1261 [2:44:45<2:31:44, 14.97s/it]
52%	[██████]	654/1261 [2:45:00<2:31:18, 14.96s/it]
52%	[██████]	655/1261 [2:45:15<2:31:00, 14.95s/it]
52%	[██████]	656/1261 [2:45:30<2:30:50, 14.96s/it]
52%	[██████]	657/1261 [2:45:45<2:30:34, 14.96s/it]
52%	[██████]	658/1261 [2:46:00<2:30:36, 14.99s/it]
52%	[██████]	659/1261 [2:46:15<2:30:51, 15.04s/it]
52%	[██████]	660/1261 [2:46:30<2:30:53, 15.06s/it]
52%	[██████]	661/1261 [2:46:45<2:30:19, 15.03s/it]
52%	[██████]	662/1261 [2:47:00<2:30:03, 15.03s/it]
53%	[██████]	663/1261 [2:47:15<2:29:40, 15.02s/it]
53%	[██████]	664/1261 [2:47:30<2:29:09, 14.99s/it]
53%	[██████]	665/1261 [2:47:45<2:29:21, 15.04s/it]
53%	[██████]	666/1261 [2:48:00<2:28:48, 15.01s/it]
53%	[██████]	667/1261 [2:48:15<2:28:16, 14.98s/it]
53%	[██████]	668/1261 [2:48:30<2:28:00, 14.98s/it]
53%	[██████]	669/1261 [2:48:45<2:27:41, 14.97s/it]
53%	[██████]	670/1261 [2:49:00<2:27:13, 14.95s/it]
53%	[██████]	671/1261 [2:49:15<2:26:52, 14.94s/it]
53%	[██████]	672/1261 [2:49:29<2:26:33, 14.93s/it]
53%	[██████]	673/1261 [2:49:45<2:26:39, 14.97s/it]
53%	[██████]	674/1261 [2:49:59<2:26:10, 14.94s/it]
54%	[██████]	675/1261 [2:50:14<2:26:11, 14.97s/it]
54%	[██████]	676/1261 [2:50:29<2:25:49, 14.96s/it]
54%	[██████]	677/1261 [2:50:44<2:25:30, 14.95s/it]
54%	[██████]	678/1261 [2:50:59<2:25:23, 14.96s/it]
54%	[██████]	679/1261 [2:51:14<2:25:04, 14.96s/it]
54%	[██████]	680/1261 [2:51:29<2:24:36, 14.93s/it]
54%	[██████]	681/1261 [2:51:44<2:24:54, 14.99s/it]
54%	[██████]	682/1261 [2:51:59<2:25:04, 15.03s/it]
54%	[██████]	683/1261 [2:52:14<2:24:35, 15.01s/it]
54%	[██████]	684/1261 [2:52:29<2:24:09, 14.99s/it]
54%	[██████]	685/1261 [2:52:44<2:23:37, 14.96s/it]
54%	[██████]	686/1261 [2:52:59<2:23:10, 14.94s/it]
54%	[██████]	687/1261 [2:53:14<2:23:02, 14.95s/it]
55%	[██████]	688/1261 [2:53:29<2:23:08, 14.99s/it]
55%	[██████]	689/1261 [2:53:44<2:22:50, 14.98s/it]
55%	[██████]	690/1261 [2:53:59<2:22:28, 14.97s/it]
55%	[██████]	691/1261 [2:54:14<2:22:27, 15.00s/it]
55%	[██████]	692/1261 [2:54:29<2:22:04, 14.98s/it]
55%	[██████]	693/1261 [2:54:44<2:21:48, 14.98s/it]
55%	[██████]	694/1261 [2:54:59<2:21:22, 14.96s/it]
55%	[██████]	695/1261 [2:55:14<2:21:04, 14.96s/it]
55%	[██████]	696/1261 [2:55:29<2:20:51, 14.96s/it]
55%	[██████]	697/1261 [2:55:44<2:20:49, 14.98s/it]
55%	[██████]	698/1261 [2:55:59<2:20:35, 14.98s/it]
55%	[██████]	699/1261 [2:56:14<2:20:19, 14.98s/it]
56%	[██████]	700/1261 [2:56:29<2:20:01, 14.98s/it]
56%	[██████]	701/1261 [2:56:44<2:19:38, 14.96s/it]
56%	[██████]	702/1261 [2:56:59<2:19:21, 14.96s/it]
56%	[██████]	703/1261 [2:57:14<2:19:00, 14.95s/it]
56%	[██████]	704/1261 [2:57:29<2:19:10, 14.99s/it]
56%	[██████]	705/1261 [2:57:44<2:19:21, 15.04s/it]
56%	[██████]	706/1261 [2:57:59<2:18:48, 15.01s/it]
56%	[██████]	707/1261 [2:58:14<2:18:31, 15.00s/it]
56%	[██████]	708/1261 [2:58:29<2:18:01, 14.97s/it]
56%	[██████]	709/1261 [2:58:44<2:17:31, 14.95s/it]
56%	[██████]	710/1261 [2:58:59<2:17:22, 14.96s/it]

56%	[██████]	711/1261 [2:59:14<2:17:22, 14.99s/it]
56%	[██████]	712/1261 [2:59:28<2:16:55, 14.96s/it]
57%	[██████]	713/1261 [2:59:43<2:16:30, 14.95s/it]
57%	[██████]	714/1261 [2:59:58<2:16:12, 14.94s/it]
57%	[██████]	715/1261 [3:00:13<2:15:54, 14.93s/it]
57%	[██████]	716/1261 [3:00:28<2:15:34, 14.93s/it]
57%	[██████]	717/1261 [3:00:43<2:15:25, 14.94s/it]
57%	[██████]	718/1261 [3:00:58<2:15:12, 14.94s/it]
57%	[██████]	719/1261 [3:01:13<2:14:52, 14.93s/it]
57%	[██████]	720/1261 [3:01:28<2:15:08, 14.99s/it]
57%	[██████]	721/1261 [3:01:43<2:14:38, 14.96s/it]
57%	[██████]	722/1261 [3:01:58<2:14:42, 15.00s/it]
57%	[██████]	723/1261 [3:02:13<2:14:20, 14.98s/it]
57%	[██████]	724/1261 [3:02:28<2:14:16, 15.00s/it]
57%	[██████]	725/1261 [3:02:43<2:13:42, 14.97s/it]
58%	[██████]	726/1261 [3:02:58<2:13:19, 14.95s/it]
58%	[██████]	727/1261 [3:03:13<2:13:28, 15.00s/it]
58%	[██████]	728/1261 [3:03:28<2:12:57, 14.97s/it]
58%	[██████]	729/1261 [3:03:43<2:12:58, 15.00s/it]
58%	[██████]	730/1261 [3:03:58<2:12:23, 14.96s/it]
58%	[██████]	731/1261 [3:04:13<2:12:06, 14.96s/it]
58%	[██████]	732/1261 [3:04:28<2:11:41, 14.94s/it]
58%	[██████]	733/1261 [3:04:43<2:12:48, 15.09s/it]
58%	[██████]	734/1261 [3:05:01<2:19:21, 15.87s/it]
58%	[██████]	735/1261 [3:05:17<2:20:15, 16.00s/it]
58%	[██████]	736/1261 [3:05:35<2:26:21, 16.73s/it]
58%	[██████]	737/1261 [3:05:52<2:25:14, 16.63s/it]
59%	[██████]	738/1261 [3:06:08<2:24:09, 16.54s/it]
59%	[██████]	739/1261 [3:06:24<2:22:56, 16.43s/it]
59%	[██████]	740/1261 [3:06:42<2:25:24, 16.75s/it]
59%	[██████]	741/1261 [3:06:58<2:22:44, 16.47s/it]
59%	[██████]	742/1261 [3:07:14<2:21:08, 16.32s/it]
59%	[██████]	743/1261 [3:07:29<2:18:09, 16.00s/it]
59%	[██████]	744/1261 [3:07:45<2:16:45, 15.87s/it]
59%	[██████]	745/1261 [3:08:00<2:15:15, 15.73s/it]
59%	[██████]	746/1261 [3:08:16<2:15:29, 15.79s/it]
59%	[██████]	747/1261 [3:08:32<2:17:31, 16.05s/it]
59%	[██████]	748/1261 [3:08:48<2:15:35, 15.86s/it]
59%	[██████]	749/1261 [3:09:03<2:14:03, 15.71s/it]
59%	[██████]	750/1261 [3:09:18<2:12:24, 15.55s/it]
60%	[██████]	751/1261 [3:09:34<2:11:27, 15.47s/it]
60%	[██████]	752/1261 [3:09:49<2:11:01, 15.45s/it]
60%	[██████]	753/1261 [3:10:04<2:10:13, 15.38s/it]
60%	[██████]	754/1261 [3:10:20<2:09:47, 15.36s/it]
60%	[██████]	755/1261 [3:10:35<2:10:27, 15.47s/it]
60%	[██████]	756/1261 [3:10:51<2:11:20, 15.61s/it]
60%	[██████]	757/1261 [3:11:09<2:17:02, 16.31s/it]
60%	[██████]	758/1261 [3:11:25<2:14:30, 16.04s/it]
60%	[██████]	759/1261 [3:11:40<2:12:04, 15.79s/it]
60%	[██████]	760/1261 [3:11:55<2:11:21, 15.73s/it]
60%	[██████]	761/1261 [3:12:11<2:09:56, 15.59s/it]
60%	[██████]	762/1261 [3:12:27<2:10:12, 15.66s/it]
61%	[██████]	763/1261 [3:12:43<2:12:05, 15.91s/it]
61%	[██████]	764/1261 [3:13:00<2:14:39, 16.26s/it]
61%	[██████]	765/1261 [3:13:16<2:13:07, 16.10s/it]
61%	[██████]	766/1261 [3:13:32<2:11:55, 15.99s/it]
61%	[██████]	767/1261 [3:13:48<2:13:43, 16.24s/it]
61%	[██████]	768/1261 [3:14:05<2:13:54, 16.30s/it]
61%	[██████]	769/1261 [3:14:21<2:12:55, 16.21s/it]
61%	[██████]	770/1261 [3:14:37<2:12:45, 16.22s/it]
61%	[██████]	771/1261 [3:14:53<2:10:33, 15.99s/it]
61%	[██████]	772/1261 [3:15:09<2:11:02, 16.08s/it]
61%	[██████]	773/1261 [3:15:25<2:11:39, 16.19s/it]
61%	[██████]	774/1261 [3:15:43<2:14:17, 16.54s/it]
61%	[██████]	775/1261 [3:16:01<2:18:10, 17.06s/it]
62%	[██████]	776/1261 [3:16:19<2:19:56, 17.31s/it]
62%	[██████]	777/1261 [3:16:36<2:19:25, 17.28s/it]
62%	[██████]	778/1261 [3:16:53<2:19:09, 17.29s/it]
62%	[██████]	779/1261 [3:17:10<2:17:43, 17.14s/it]
62%	[██████]	780/1261 [3:17:27<2:16:06, 16.98s/it]
62%	[██████]	781/1261 [3:17:45<2:17:47, 17.22s/it]
62%	[██████]	782/1261 [3:18:00<2:14:09, 16.81s/it]
62%	[██████]	783/1261 [3:18:16<2:11:46, 16.54s/it]
62%	[██████]	784/1261 [3:18:32<2:09:55, 16.34s/it]
62%	[██████]	785/1261 [3:18:48<2:08:40, 16.22s/it]
62%	[██████]	786/1261 [3:19:04<2:08:33, 16.24s/it]
62%	[██████]	787/1261 [3:19:21<2:08:46, 16.30s/it]
62%	[██████]	788/1261 [3:19:36<2:06:06, 16.00s/it]
63%	[██████]	789/1261 [3:19:52<2:04:48, 15.87s/it]
63%	[██████]	790/1261 [3:20:07<2:03:39, 15.75s/it]
63%	[██████]	791/1261 [3:20:23<2:02:59, 15.70s/it]
63%	[██████]	792/1261 [3:20:38<2:02:13, 15.64s/it]
63%	[██████]	793/1261 [3:20:54<2:01:44, 15.61s/it]
63%	[██████]	794/1261 [3:21:09<2:00:29, 15.48s/it]
63%	[██████]	795/1261 [3:21:24<2:00:01, 15.45s/it]
63%	[██████]	796/1261 [3:21:40<1:59:56, 15.48s/it]
63%	[██████]	797/1261 [3:21:55<1:59:35, 15.46s/it]
63%	[██████]	798/1261 [3:22:11<1:59:24, 15.47s/it]
63%	[██████]	799/1261 [3:22:26<1:58:27, 15.38s/it]

63%	[██████████]	800/1261 [3:22:41<1:58:11, 15.38s/it]
64%	[██████████]	801/1261 [3:22:57<1:57:25, 15.32s/it]
64%	[██████████]	802/1261 [3:23:12<1:57:43, 15.39s/it]
64%	[██████████]	803/1261 [3:23:27<1:57:12, 15.36s/it]
64%	[██████████]	804/1261 [3:23:43<1:57:09, 15.38s/it]
64%	[██████████]	805/1261 [3:23:58<1:56:58, 15.39s/it]
64%	[██████████]	806/1261 [3:24:13<1:55:57, 15.29s/it]
64%	[██████████]	807/1261 [3:24:30<1:58:08, 15.61s/it]
64%	[██████████]	808/1261 [3:24:46<1:59:35, 15.84s/it]
64%	[██████████]	809/1261 [3:25:02<1:58:57, 15.79s/it]
64%	[██████████]	810/1261 [3:25:17<1:58:47, 15.80s/it]
64%	[██████████]	811/1261 [3:25:34<1:59:01, 15.87s/it]
64%	[██████████]	812/1261 [3:25:49<1:58:41, 15.86s/it]
64%	[██████████]	813/1261 [3:26:06<1:59:40, 16.03s/it]
65%	[██████████]	814/1261 [3:26:21<1:58:02, 15.84s/it]
65%	[██████████]	815/1261 [3:26:37<1:56:55, 15.73s/it]
65%	[██████████]	816/1261 [3:26:52<1:55:54, 15.63s/it]
65%	[██████████]	817/1261 [3:27:08<1:56:23, 15.73s/it]
65%	[██████████]	818/1261 [3:27:24<1:56:03, 15.72s/it]
65%	[██████████]	819/1261 [3:27:40<1:56:57, 15.88s/it]
65%	[██████████]	820/1261 [3:27:55<1:55:53, 15.77s/it]
65%	[██████████]	821/1261 [3:28:11<1:55:22, 15.73s/it]
65%	[██████████]	822/1261 [3:28:27<1:55:03, 15.73s/it]
65%	[██████████]	823/1261 [3:28:43<1:56:20, 15.94s/it]
65%	[██████████]	824/1261 [3:29:01<1:59:15, 16.37s/it]
65%	[██████████]	825/1261 [3:29:17<1:58:02, 16.24s/it]
66%	[██████████]	826/1261 [3:29:32<1:56:41, 16.09s/it]
66%	[██████████]	827/1261 [3:29:48<1:56:04, 16.05s/it]
66%	[██████████]	828/1261 [3:30:04<1:55:25, 15.99s/it]
66%	[██████████]	829/1261 [3:30:20<1:54:51, 15.95s/it]
66%	[██████████]	830/1261 [3:30:36<1:55:19, 16.05s/it]
66%	[██████████]	831/1261 [3:30:52<1:54:02, 15.91s/it]
66%	[██████████]	832/1261 [3:31:08<1:53:45, 15.91s/it]
66%	[██████████]	833/1261 [3:31:24<1:53:10, 15.87s/it]
66%	[██████████]	834/1261 [3:31:39<1:53:06, 15.89s/it]
66%	[██████████]	835/1261 [3:31:57<1:56:08, 16.36s/it]
66%	[██████████]	836/1261 [3:32:14<1:57:20, 16.57s/it]
66%	[██████████]	837/1261 [3:32:31<1:58:21, 16.75s/it]
66%	[██████████]	838/1261 [3:32:48<1:58:05, 16.75s/it]
67%	[██████████]	839/1261 [3:33:04<1:56:08, 16.51s/it]
67%	[██████████]	840/1261 [3:33:20<1:54:46, 16.36s/it]
67%	[██████████]	841/1261 [3:33:37<1:55:19, 16.47s/it]
67%	[██████████]	842/1261 [3:33:53<1:55:47, 16.58s/it]
67%	[██████████]	843/1261 [3:34:10<1:55:09, 16.53s/it]
67%	[██████████]	844/1261 [3:34:26<1:54:06, 16.42s/it]
67%	[██████████]	845/1261 [3:34:42<1:53:11, 16.33s/it]
67%	[██████████]	846/1261 [3:34:58<1:52:49, 16.31s/it]
67%	[██████████]	847/1261 [3:35:16<1:54:20, 16.57s/it]
67%	[██████████]	848/1261 [3:35:31<1:52:39, 16.37s/it]
67%	[██████████]	849/1261 [3:35:48<1:53:24, 16.52s/it]
67%	[██████████]	850/1261 [3:36:05<1:53:05, 16.51s/it]
67%	[██████████]	851/1261 [3:36:22<1:53:39, 16.63s/it]
68%	[██████████]	852/1261 [3:36:39<1:54:10, 16.75s/it]
68%	[██████████]	853/1261 [3:36:56<1:54:50, 16.89s/it]
68%	[██████████]	854/1261 [3:37:12<1:53:26, 16.72s/it]
68%	[██████████]	855/1261 [3:37:29<1:52:44, 16.66s/it]
68%	[██████████]	856/1261 [3:37:44<1:50:08, 16.32s/it]
68%	[██████████]	857/1261 [3:38:01<1:49:54, 16.32s/it]
68%	[██████████]	858/1261 [3:38:16<1:48:28, 16.15s/it]
68%	[██████████]	859/1261 [3:38:32<1:47:02, 15.98s/it]
68%	[██████████]	860/1261 [3:38:49<1:47:47, 16.13s/it]
68%	[██████████]	861/1261 [3:39:05<1:48:30, 16.28s/it]
68%	[██████████]	862/1261 [3:39:21<1:47:18, 16.14s/it]
68%	[██████████]	863/1261 [3:39:37<1:46:33, 16.06s/it]
69%	[██████████]	864/1261 [3:39:54<1:49:24, 16.54s/it]
69%	[██████████]	865/1261 [3:40:12<1:51:28, 16.89s/it]
69%	[██████████]	866/1261 [3:40:30<1:53:00, 17.17s/it]
69%	[██████████]	867/1261 [3:40:48<1:53:45, 17.32s/it]
69%	[██████████]	868/1261 [3:41:06<1:55:40, 17.66s/it]
69%	[██████████]	869/1261 [3:41:24<1:56:17, 17.80s/it]
69%	[██████████]	870/1261 [3:41:42<1:56:48, 17.92s/it]
69%	[██████████]	871/1261 [3:42:00<1:56:30, 17.92s/it]
69%	[██████████]	872/1261 [3:42:18<1:56:13, 17.93s/it]
69%	[██████████]	873/1261 [3:42:36<1:56:11, 17.97s/it]
69%	[██████████]	874/1261 [3:42:54<1:55:57, 17.98s/it]
69%	[██████████]	875/1261 [3:43:13<1:56:10, 18.06s/it]
69%	[██████████]	876/1261 [3:43:29<1:53:34, 17.70s/it]
70%	[██████████]	877/1261 [3:43:47<1:52:27, 17.57s/it]
70%	[██████████]	878/1261 [3:44:04<1:50:52, 17.37s/it]
70%	[██████████]	879/1261 [3:44:20<1:49:09, 17.14s/it]
70%	[██████████]	880/1261 [3:44:37<1:48:03, 17.02s/it]
70%	[██████████]	881/1261 [3:44:54<1:47:14, 16.93s/it]
70%	[██████████]	882/1261 [3:45:10<1:46:25, 16.85s/it]
70%	[██████████]	883/1261 [3:45:27<1:46:07, 16.85s/it]
70%	[██████████]	884/1261 [3:45:44<1:46:06, 16.89s/it]
70%	[██████████]	885/1261 [3:46:01<1:45:28, 16.83s/it]
70%	[██████████]	886/1261 [3:46:18<1:45:14, 16.84s/it]
70%	[██████████]	887/1261 [3:46:35<1:44:57, 16.84s/it]
70%	[██████████]	888/1261 [3:46:52<1:45:41, 17.00s/it]

70%	[██████████]	889/1261 [3:47:09<1:44:34, 16.87s/it]
71%	[██████████]	890/1261 [3:47:26<1:44:59, 16.98s/it]
71%	[██████████]	891/1261 [3:47:43<1:44:39, 16.97s/it]
71%	[██████████]	892/1261 [3:48:00<1:44:43, 17.03s/it]
71%	[██████████]	893/1261 [3:48:17<1:44:03, 16.97s/it]
71%	[██████████]	894/1261 [3:48:33<1:43:15, 16.88s/it]
71%	[██████████]	895/1261 [3:48:50<1:43:00, 16.89s/it]
71%	[██████████]	896/1261 [3:49:08<1:43:48, 17.06s/it]
71%	[██████████]	897/1261 [3:49:25<1:43:47, 17.11s/it]
71%	[██████████]	898/1261 [3:49:42<1:43:42, 17.14s/it]
71%	[██████████]	899/1261 [3:49:58<1:41:04, 16.75s/it]
71%	[██████████]	900/1261 [3:50:15<1:41:42, 16.91s/it]
71%	[██████████]	901/1261 [3:50:32<1:41:32, 16.92s/it]
72%	[██████████]	902/1261 [3:50:49<1:41:28, 16.96s/it]
72%	[██████████]	903/1261 [3:51:07<1:41:55, 17.08s/it]
72%	[██████████]	904/1261 [3:51:25<1:42:56, 17.30s/it]
72%	[██████████]	905/1261 [3:51:42<1:42:39, 17.30s/it]
72%	[██████████]	906/1261 [3:51:59<1:42:06, 17.26s/it]
72%	[██████████]	907/1261 [3:52:15<1:39:39, 16.89s/it]
72%	[██████████]	908/1261 [3:52:31<1:37:19, 16.54s/it]
72%	[██████████]	909/1261 [3:52:46<1:35:23, 16.26s/it]
72%	[██████████]	910/1261 [3:53:02<1:33:56, 16.06s/it]
72%	[██████████]	911/1261 [3:53:18<1:34:26, 16.19s/it]
72%	[██████████]	912/1261 [3:53:36<1:36:04, 16.52s/it]
72%	[██████████]	913/1261 [3:53:53<1:37:54, 16.88s/it]
72%	[██████████]	914/1261 [3:54:11<1:38:25, 17.02s/it]
73%	[██████████]	915/1261 [3:54:26<1:35:53, 16.63s/it]
73%	[██████████]	916/1261 [3:54:42<1:34:02, 16.35s/it]
73%	[██████████]	917/1261 [3:54:58<1:32:54, 16.21s/it]
73%	[██████████]	918/1261 [3:55:15<1:33:07, 16.29s/it]
73%	[██████████]	919/1261 [3:55:33<1:35:44, 16.80s/it]
73%	[██████████]	920/1261 [3:55:50<1:36:36, 17.00s/it]
73%	[██████████]	921/1261 [3:56:08<1:37:25, 17.19s/it]
73%	[██████████]	922/1261 [3:56:25<1:37:17, 17.22s/it]
73%	[██████████]	923/1261 [3:56:42<1:37:06, 17.24s/it]
73%	[██████████]	924/1261 [3:57:00<1:37:16, 17.32s/it]
73%	[██████████]	925/1261 [3:57:17<1:36:42, 17.27s/it]
73%	[██████████]	926/1261 [3:57:34<1:36:21, 17.26s/it]
74%	[██████████]	927/1261 [3:57:52<1:36:29, 17.33s/it]
74%	[██████████]	928/1261 [3:58:09<1:36:35, 17.40s/it]
74%	[██████████]	929/1261 [3:58:26<1:35:24, 17.24s/it]
74%	[██████████]	930/1261 [3:58:43<1:34:48, 17.18s/it]
74%	[██████████]	931/1261 [3:59:00<1:34:40, 17.21s/it]
74%	[██████████]	932/1261 [3:59:18<1:34:32, 17.24s/it]
74%	[██████████]	933/1261 [3:59:36<1:35:35, 17.49s/it]
74%	[██████████]	934/1261 [3:59:53<1:34:39, 17.37s/it]
74%	[██████████]	935/1261 [4:00:10<1:34:35, 17.41s/it]
74%	[██████████]	936/1261 [4:00:28<1:34:56, 17.53s/it]
74%	[██████████]	937/1261 [4:00:46<1:34:27, 17.49s/it]
74%	[██████████]	938/1261 [4:01:02<1:32:51, 17.25s/it]
74%	[██████████]	939/1261 [4:01:20<1:32:36, 17.26s/it]
75%	[██████████]	940/1261 [4:01:37<1:32:06, 17.22s/it]
75%	[██████████]	941/1261 [4:01:53<1:30:54, 17.05s/it]
75%	[██████████]	942/1261 [4:02:10<1:30:29, 17.02s/it]
75%	[██████████]	943/1261 [4:02:29<1:32:27, 17.45s/it]
75%	[██████████]	944/1261 [4:02:45<1:30:59, 17.22s/it]
75%	[██████████]	945/1261 [4:03:03<1:31:35, 17.39s/it]
75%	[██████████]	946/1261 [4:03:21<1:31:42, 17.47s/it]
75%	[██████████]	947/1261 [4:03:38<1:30:19, 17.26s/it]
75%	[██████████]	948/1261 [4:03:55<1:30:37, 17.37s/it]
75%	[██████████]	949/1261 [4:04:12<1:29:02, 17.12s/it]
75%	[██████████]	950/1261 [4:04:28<1:28:00, 16.98s/it]
75%	[██████████]	951/1261 [4:04:45<1:27:03, 16.85s/it]
75%	[██████████]	952/1261 [4:05:01<1:26:06, 16.72s/it]
76%	[██████████]	953/1261 [4:05:18<1:25:07, 16.58s/it]
76%	[██████████]	954/1261 [4:05:34<1:24:49, 16.58s/it]
76%	[██████████]	955/1261 [4:05:50<1:24:05, 16.49s/it]
76%	[██████████]	956/1261 [4:06:07<1:23:16, 16.38s/it]
76%	[██████████]	957/1261 [4:06:23<1:22:27, 16.28s/it]
76%	[██████████]	958/1261 [4:06:40<1:23:33, 16.55s/it]
76%	[██████████]	959/1261 [4:06:57<1:23:49, 16.66s/it]
76%	[██████████]	960/1261 [4:07:14<1:24:03, 16.76s/it]
76%	[██████████]	961/1261 [4:07:31<1:24:24, 16.88s/it]
76%	[██████████]	962/1261 [4:07:48<1:24:07, 16.88s/it]
76%	[██████████]	963/1261 [4:08:05<1:23:53, 16.89s/it]
76%	[██████████]	964/1261 [4:08:22<1:24:54, 17.15s/it]
77%	[██████████]	965/1261 [4:08:40<1:25:30, 17.33s/it]
77%	[██████████]	966/1261 [4:08:58<1:25:58, 17.49s/it]
77%	[██████████]	967/1261 [4:09:16<1:26:25, 17.64s/it]
77%	[██████████]	968/1261 [4:09:34<1:26:55, 17.80s/it]
77%	[██████████]	969/1261 [4:09:52<1:26:30, 17.78s/it]
77%	[██████████]	970/1261 [4:10:10<1:26:00, 17.73s/it]
77%	[██████████]	971/1261 [4:10:28<1:26:00, 17.80s/it]
77%	[██████████]	972/1261 [4:10:45<1:24:49, 17.61s/it]
77%	[██████████]	973/1261 [4:11:02<1:23:28, 17.39s/it]
77%	[██████████]	974/1261 [4:11:19<1:23:21, 17.43s/it]
77%	[██████████]	975/1261 [4:11:35<1:21:26, 17.08s/it]
77%	[██████████]	976/1261 [4:11:52<1:20:49, 17.01s/it]
77%	[██████████]	977/1261 [4:12:09<1:19:52, 16.87s/it]

78%	[██████████]	978/1261 [4:12:24<1:17:26, 16.42s/it]
78%	[██████████]	979/1261 [4:12:40<1:16:15, 16.23s/it]
78%	[██████████]	980/1261 [4:12:56<1:15:43, 16.17s/it]
78%	[██████████]	981/1261 [4:13:12<1:14:55, 16.06s/it]
78%	[██████████]	982/1261 [4:13:28<1:14:22, 16.00s/it]
78%	[██████████]	983/1261 [4:13:43<1:13:57, 15.96s/it]
78%	[██████████]	984/1261 [4:13:59<1:13:37, 15.95s/it]
78%	[██████████]	985/1261 [4:14:15<1:13:26, 15.97s/it]
78%	[██████████]	986/1261 [4:14:31<1:13:09, 15.96s/it]
78%	[██████████]	987/1261 [4:14:47<1:12:30, 15.88s/it]
78%	[██████████]	988/1261 [4:15:03<1:12:08, 15.86s/it]
78%	[██████████]	989/1261 [4:15:19<1:12:10, 15.92s/it]
79%	[██████████]	990/1261 [4:15:36<1:13:12, 16.21s/it]
79%	[██████████]	991/1261 [4:15:52<1:13:16, 16.28s/it]
79%	[██████████]	992/1261 [4:16:09<1:13:01, 16.29s/it]
79%	[██████████]	993/1261 [4:16:25<1:12:31, 16.24s/it]
79%	[██████████]	994/1261 [4:16:41<1:11:59, 16.18s/it]
79%	[██████████]	995/1261 [4:16:57<1:11:57, 16.23s/it]
79%	[██████████]	996/1261 [4:17:13<1:11:02, 16.08s/it]
79%	[██████████]	997/1261 [4:17:29<1:10:19, 15.98s/it]
79%	[██████████]	998/1261 [4:17:45<1:10:42, 16.13s/it]
79%	[██████████]	999/1261 [4:18:02<1:11:46, 16.44s/it]
79%	[██████████]	1000/1261 [4:18:19<1:11:51, 16.52s/it]
79%	[██████████]	1001/1261 [4:18:35<1:11:40, 16.54s/it]
79%	[██████████]	1002/1261 [4:18:52<1:10:59, 16.45s/it]
80%	[██████████]	1003/1261 [4:19:08<1:10:05, 16.30s/it]
80%	[██████████]	1004/1261 [4:19:24<1:09:34, 16.24s/it]
80%	[██████████]	1005/1261 [4:19:40<1:09:45, 16.35s/it]
80%	[██████████]	1006/1261 [4:19:58<1:10:30, 16.59s/it]
80%	[██████████]	1007/1261 [4:20:15<1:10:59, 16.77s/it]
80%	[██████████]	1008/1261 [4:20:31<1:10:03, 16.61s/it]
80%	[██████████]	1009/1261 [4:20:47<1:09:40, 16.59s/it]
80%	[██████████]	1010/1261 [4:21:05<1:09:55, 16.72s/it]
80%	[██████████]	1011/1261 [4:21:22<1:10:03, 16.82s/it]
80%	[██████████]	1012/1261 [4:21:37<1:08:27, 16.50s/it]
80%	[██████████]	1013/1261 [4:21:53<1:07:40, 16.37s/it]
80%	[██████████]	1014/1261 [4:22:11<1:08:33, 16.65s/it]
80%	[██████████]	1015/1261 [4:22:28<1:08:59, 16.83s/it]
81%	[██████████]	1016/1261 [4:22:45<1:09:13, 16.95s/it]
81%	[██████████]	1017/1261 [4:23:02<1:08:28, 16.84s/it]
81%	[██████████]	1018/1261 [4:23:18<1:07:55, 16.77s/it]
81%	[██████████]	1019/1261 [4:23:34<1:06:14, 16.42s/it]
81%	[██████████]	1020/1261 [4:23:51<1:06:29, 16.55s/it]
81%	[██████████]	1021/1261 [4:24:07<1:06:02, 16.51s/it]
81%	[██████████]	1022/1261 [4:24:23<1:05:27, 16.43s/it]
81%	[██████████]	1023/1261 [4:24:40<1:05:50, 16.60s/it]
81%	[██████████]	1024/1261 [4:24:57<1:05:24, 16.56s/it]
81%	[██████████]	1025/1261 [4:25:13<1:04:11, 16.32s/it]
81%	[██████████]	1026/1261 [4:25:28<1:02:53, 16.06s/it]
81%	[██████████]	1027/1261 [4:25:44<1:02:34, 16.04s/it]
82%	[██████████]	1028/1261 [4:26:00<1:01:48, 15.92s/it]
82%	[██████████]	1029/1261 [4:26:15<1:00:59, 15.77s/it]
82%	[██████████]	1030/1261 [4:26:31<1:00:11, 15.64s/it]
82%	[██████████]	1031/1261 [4:26:46<1:00:05, 15.68s/it]
82%	[██████████]	1032/1261 [4:27:03<1:00:43, 15.91s/it]
82%	[██████████]	1033/1261 [4:27:19<1:00:58, 16.04s/it]
82%	[██████████]	1034/1261 [4:27:35<1:00:37, 16.02s/it]
82%	[██████████]	1035/1261 [4:27:51<1:00:31, 16.07s/it]
82%	[██████████]	1036/1261 [4:28:07<1:00:01, 16.01s/it]
82%	[██████████]	1037/1261 [4:28:23<1:00:00, 16.07s/it]
82%	[██████████]	1038/1261 [4:28:40<1:00:18, 16.23s/it]
82%	[██████████]	1039/1261 [4:28:55<59:04, 15.97s/it]
82%	[██████████]	1040/1261 [4:29:12<59:32, 16.16s/it]
83%	[██████████]	1041/1261 [4:29:29<1:00:24, 16.48s/it]
83%	[██████████]	1042/1261 [4:29:46<1:00:37, 16.61s/it]
83%	[██████████]	1043/1261 [4:30:03<1:00:10, 16.56s/it]
83%	[██████████]	1044/1261 [4:30:19<59:54, 16.56s/it]
83%	[██████████]	1045/1261 [4:30:36<1:00:11, 16.72s/it]
83%	[██████████]	1046/1261 [4:30:53<1:00:28, 16.87s/it]
83%	[██████████]	1047/1261 [4:31:10<1:00:23, 16.93s/it]
83%	[██████████]	1048/1261 [4:31:27<59:52, 16.87s/it]
83%	[██████████]	1049/1261 [4:31:44<59:57, 16.97s/it]
83%	[██████████]	1050/1261 [4:32:00<58:14, 16.56s/it]
83%	[██████████]	1051/1261 [4:32:15<56:37, 16.18s/it]
83%	[██████████]	1052/1261 [4:32:31<55:22, 15.90s/it]
84%	[██████████]	1053/1261 [4:32:46<54:45, 15.80s/it]
84%	[██████████]	1054/1261 [4:33:03<55:48, 16.18s/it]
84%	[██████████]	1055/1261 [4:33:19<55:04, 16.04s/it]
84%	[██████████]	1056/1261 [4:33:35<55:02, 16.11s/it]
84%	[██████████]	1057/1261 [4:33:51<55:00, 16.18s/it]
84%	[██████████]	1058/1261 [4:34:09<56:00, 16.55s/it]
84%	[██████████]	1059/1261 [4:34:26<56:00, 16.64s/it]
84%	[██████████]	1060/1261 [4:34:42<55:36, 16.60s/it]
84%	[██████████]	1061/1261 [4:35:01<57:21, 17.21s/it]
84%	[██████████]	1062/1261 [4:35:19<58:09, 17.54s/it]
84%	[██████████]	1063/1261 [4:35:37<58:08, 17.62s/it]
84%	[██████████]	1064/1261 [4:35:56<59:14, 18.04s/it]
84%	[██████████]	1065/1261 [4:36:12<57:21, 17.56s/it]
85%	[██████████]	1066/1261 [4:36:29<56:32, 17.40s/it]

85%	[REDACTED]	1067/1261 [4:36:46<55:28, 17.16s/it]
85%	[REDACTED]	1068/1261 [4:37:02<54:09, 16.84s/it]
85%	[REDACTED]	1069/1261 [4:37:18<52:47, 16.50s/it]
85%	[REDACTED]	1070/1261 [4:37:34<52:26, 16.47s/it]
85%	[REDACTED]	1071/1261 [4:37:51<52:04, 16.44s/it]
85%	[REDACTED]	1072/1261 [4:38:06<50:56, 16.17s/it]
85%	[REDACTED]	1073/1261 [4:38:23<51:07, 16.31s/it]
85%	[REDACTED]	1074/1261 [4:38:39<50:57, 16.35s/it]
85%	[REDACTED]	1075/1261 [4:38:56<50:49, 16.39s/it]
85%	[REDACTED]	1076/1261 [4:39:12<50:35, 16.41s/it]
85%	[REDACTED]	1077/1261 [4:39:28<49:50, 16.25s/it]
85%	[REDACTED]	1078/1261 [4:39:44<49:16, 16.16s/it]
86%	[REDACTED]	1079/1261 [4:40:00<48:46, 16.08s/it]
86%	[REDACTED]	1080/1261 [4:40:16<48:23, 16.04s/it]
86%	[REDACTED]	1081/1261 [4:40:32<47:56, 15.98s/it]
86%	[REDACTED]	1082/1261 [4:40:48<47:54, 16.06s/it]
86%	[REDACTED]	1083/1261 [4:41:04<47:36, 16.05s/it]
86%	[REDACTED]	1084/1261 [4:41:20<46:58, 15.92s/it]
86%	[REDACTED]	1085/1261 [4:41:35<46:40, 15.91s/it]
86%	[REDACTED]	1086/1261 [4:41:51<46:18, 15.87s/it]
86%	[REDACTED]	1087/1261 [4:42:07<45:52, 15.82s/it]
86%	[REDACTED]	1088/1261 [4:42:24<46:47, 16.23s/it]
86%	[REDACTED]	1089/1261 [4:42:40<46:26, 16.20s/it]
86%	[REDACTED]	1090/1261 [4:42:56<45:50, 16.09s/it]
87%	[REDACTED]	1091/1261 [4:43:12<45:30, 16.06s/it]
87%	[REDACTED]	1092/1261 [4:43:28<45:13, 16.06s/it]
87%	[REDACTED]	1093/1261 [4:43:44<45:07, 16.12s/it]
87%	[REDACTED]	1094/1261 [4:44:00<44:41, 16.05s/it]
87%	[REDACTED]	1095/1261 [4:44:17<44:31, 16.09s/it]
87%	[REDACTED]	1096/1261 [4:44:33<44:55, 16.34s/it]
87%	[REDACTED]	1097/1261 [4:44:50<44:54, 16.43s/it]
87%	[REDACTED]	1098/1261 [4:45:08<45:48, 16.86s/it]
87%	[REDACTED]	1099/1261 [4:45:24<44:57, 16.65s/it]
87%	[REDACTED]	1100/1261 [4:45:40<44:17, 16.51s/it]
87%	[REDACTED]	1101/1261 [4:45:57<43:53, 16.46s/it]
87%	[REDACTED]	1102/1261 [4:46:13<43:36, 16.46s/it]
87%	[REDACTED]	1103/1261 [4:46:31<44:07, 16.76s/it]
88%	[REDACTED]	1104/1261 [4:46:48<44:41, 17.08s/it]
88%	[REDACTED]	1105/1261 [4:47:07<45:21, 17.44s/it]
88%	[REDACTED]	1106/1261 [4:47:25<45:37, 17.66s/it]
88%	[REDACTED]	1107/1261 [4:47:43<45:48, 17.85s/it]
88%	[REDACTED]	1108/1261 [4:48:00<44:37, 17.50s/it]
88%	[REDACTED]	1109/1261 [4:48:17<43:50, 17.30s/it]
88%	[REDACTED]	1110/1261 [4:48:34<43:34, 17.31s/it]
88%	[REDACTED]	1111/1261 [4:48:52<43:28, 17.39s/it]
88%	[REDACTED]	1112/1261 [4:49:08<42:42, 17.20s/it]
88%	[REDACTED]	1113/1261 [4:49:25<41:45, 16.93s/it]
88%	[REDACTED]	1114/1261 [4:49:40<40:38, 16.59s/it]
88%	[REDACTED]	1115/1261 [4:49:56<39:56, 16.41s/it]
89%	[REDACTED]	1116/1261 [4:50:12<39:20, 16.28s/it]
89%	[REDACTED]	1117/1261 [4:50:28<38:40, 16.11s/it]
89%	[REDACTED]	1118/1261 [4:50:44<38:21, 16.09s/it]
89%	[REDACTED]	1119/1261 [4:51:01<38:32, 16.29s/it]
89%	[REDACTED]	1120/1261 [4:51:17<38:29, 16.38s/it]
89%	[REDACTED]	1121/1261 [4:51:34<38:05, 16.32s/it]
89%	[REDACTED]	1122/1261 [4:51:49<37:21, 16.13s/it]
89%	[REDACTED]	1123/1261 [4:52:05<37:03, 16.11s/it]
89%	[REDACTED]	1124/1261 [4:52:21<36:32, 16.01s/it]
89%	[REDACTED]	1125/1261 [4:52:37<36:16, 16.00s/it]
89%	[REDACTED]	1126/1261 [4:52:53<35:55, 15.96s/it]
89%	[REDACTED]	1127/1261 [4:53:09<35:34, 15.93s/it]
89%	[REDACTED]	1128/1261 [4:53:25<35:23, 15.96s/it]
90%	[REDACTED]	1129/1261 [4:53:41<35:02, 15.93s/it]
90%	[REDACTED]	1130/1261 [4:53:57<34:59, 16.03s/it]
90%	[REDACTED]	1131/1261 [4:54:14<35:18, 16.30s/it]
90%	[REDACTED]	1132/1261 [4:54:30<34:38, 16.11s/it]
90%	[REDACTED]	1133/1261 [4:54:46<34:18, 16.08s/it]
90%	[REDACTED]	1134/1261 [4:55:01<33:47, 15.97s/it]
90%	[REDACTED]	1135/1261 [4:55:17<33:12, 15.81s/it]
90%	[REDACTED]	1136/1261 [4:55:32<32:47, 15.74s/it]
90%	[REDACTED]	1137/1261 [4:55:48<32:26, 15.70s/it]
90%	[REDACTED]	1138/1261 [4:56:04<32:11, 15.71s/it]
90%	[REDACTED]	1139/1261 [4:56:19<31:52, 15.67s/it]
90%	[REDACTED]	1140/1261 [4:56:35<31:34, 15.66s/it]
90%	[REDACTED]	1141/1261 [4:56:51<31:22, 15.69s/it]
91%	[REDACTED]	1142/1261 [4:57:06<31:03, 15.66s/it]
91%	[REDACTED]	1143/1261 [4:57:22<30:48, 15.67s/it]
91%	[REDACTED]	1144/1261 [4:57:38<30:36, 15.69s/it]
91%	[REDACTED]	1145/1261 [4:57:53<30:11, 15.62s/it]
91%	[REDACTED]	1146/1261 [4:58:08<29:46, 15.54s/it]
91%	[REDACTED]	1147/1261 [4:58:24<29:19, 15.43s/it]
91%	[REDACTED]	1148/1261 [4:58:39<29:00, 15.40s/it]
91%	[REDACTED]	1149/1261 [4:58:55<29:01, 15.55s/it]
91%	[REDACTED]	1150/1261 [4:59:11<29:12, 15.79s/it]
91%	[REDACTED]	1151/1261 [4:59:28<29:41, 16.20s/it]
91%	[REDACTED]	1152/1261 [4:59:45<29:27, 16.21s/it]
91%	[REDACTED]	1153/1261 [5:00:01<29:21, 16.31s/it]
92%	[REDACTED]	1154/1261 [5:00:18<29:14, 16.39s/it]
92%	[REDACTED]	1155/1261 [5:00:34<28:48, 16.31s/it]

92%	[REDACTED]	1156/1261 [5:00:50<28:20, 16.20s/it]
92%	[REDACTED]	1157/1261 [5:01:05<27:41, 15.97s/it]
92%	[REDACTED]	1158/1261 [5:01:21<27:10, 15.83s/it]
92%	[REDACTED]	1159/1261 [5:01:37<27:15, 16.03s/it]
92%	[REDACTED]	1160/1261 [5:01:53<26:38, 15.82s/it]
92%	[REDACTED]	1161/1261 [5:02:08<26:01, 15.62s/it]
92%	[REDACTED]	1162/1261 [5:02:23<25:33, 15.49s/it]
92%	[REDACTED]	1163/1261 [5:02:38<25:06, 15.37s/it]
92%	[REDACTED]	1164/1261 [5:02:54<24:56, 15.43s/it]
92%	[REDACTED]	1165/1261 [5:03:11<25:26, 15.90s/it]
92%	[REDACTED]	1166/1261 [5:03:27<25:26, 16.06s/it]
93%	[REDACTED]	1167/1261 [5:03:44<25:25, 16.23s/it]
93%	[REDACTED]	1168/1261 [5:04:01<25:31, 16.47s/it]
93%	[REDACTED]	1169/1261 [5:04:17<25:01, 16.32s/it]
93%	[REDACTED]	1170/1261 [5:04:33<24:42, 16.29s/it]
93%	[REDACTED]	1171/1261 [5:04:49<24:23, 16.26s/it]
93%	[REDACTED]	1172/1261 [5:05:05<23:59, 16.17s/it]
93%	[REDACTED]	1173/1261 [5:05:21<23:28, 16.01s/it]
93%	[REDACTED]	1174/1261 [5:05:36<22:43, 15.67s/it]
93%	[REDACTED]	1175/1261 [5:05:53<23:01, 16.06s/it]
93%	[REDACTED]	1176/1261 [5:06:09<23:06, 16.31s/it]
93%	[REDACTED]	1177/1261 [5:06:26<22:48, 16.30s/it]
93%	[REDACTED]	1178/1261 [5:06:42<22:27, 16.23s/it]
93%	[REDACTED]	1179/1261 [5:07:00<22:51, 16.72s/it]
94%	[REDACTED]	1180/1261 [5:07:16<22:31, 16.69s/it]
94%	[REDACTED]	1181/1261 [5:07:32<21:53, 16.42s/it]
94%	[REDACTED]	1182/1261 [5:07:47<21:11, 16.09s/it]
94%	[REDACTED]	1183/1261 [5:08:03<20:38, 15.88s/it]
94%	[REDACTED]	1184/1261 [5:08:18<20:10, 15.72s/it]
94%	[REDACTED]	1185/1261 [5:08:35<20:20, 16.05s/it]
94%	[REDACTED]	1186/1261 [5:08:52<20:36, 16.48s/it]
94%	[REDACTED]	1187/1261 [5:09:10<20:36, 16.71s/it]
94%	[REDACTED]	1188/1261 [5:09:26<20:18, 16.69s/it]
94%	[REDACTED]	1189/1261 [5:09:42<19:44, 16.46s/it]
94%	[REDACTED]	1190/1261 [5:10:00<19:47, 16.73s/it]
94%	[REDACTED]	1191/1261 [5:10:17<19:49, 16.99s/it]
95%	[REDACTED]	1192/1261 [5:10:35<19:57, 17.36s/it]
95%	[REDACTED]	1193/1261 [5:10:53<19:49, 17.49s/it]
95%	[REDACTED]	1194/1261 [5:11:10<19:21, 17.33s/it]
95%	[REDACTED]	1195/1261 [5:11:28<19:08, 17.41s/it]
95%	[REDACTED]	1196/1261 [5:11:45<18:54, 17.46s/it]
95%	[REDACTED]	1197/1261 [5:12:03<18:43, 17.56s/it]
95%	[REDACTED]	1198/1261 [5:12:21<18:29, 17.61s/it]
95%	[REDACTED]	1199/1261 [5:12:38<18:01, 17.44s/it]
95%	[REDACTED]	1200/1261 [5:12:55<17:45, 17.47s/it]
95%	[REDACTED]	1201/1261 [5:13:12<17:13, 17.22s/it]
95%	[REDACTED]	1202/1261 [5:13:29<16:53, 17.18s/it]
95%	[REDACTED]	1203/1261 [5:13:45<16:18, 16.88s/it]
95%	[REDACTED]	1204/1261 [5:14:02<15:58, 16.81s/it]
96%	[REDACTED]	1205/1261 [5:14:20<15:55, 17.06s/it]
96%	[REDACTED]	1206/1261 [5:14:37<15:35, 17.01s/it]
96%	[REDACTED]	1207/1261 [5:14:54<15:29, 17.21s/it]
96%	[REDACTED]	1208/1261 [5:15:10<14:53, 16.86s/it]
96%	[REDACTED]	1209/1261 [5:15:25<14:05, 16.27s/it]
96%	[REDACTED]	1210/1261 [5:15:42<14:02, 16.52s/it]
96%	[REDACTED]	1211/1261 [5:15:58<13:30, 16.21s/it]
96%	[REDACTED]	1212/1261 [5:16:14<13:10, 16.13s/it]
96%	[REDACTED]	1213/1261 [5:16:32<13:19, 16.65s/it]
96%	[REDACTED]	1214/1261 [5:16:49<13:13, 16.89s/it]
96%	[REDACTED]	1215/1261 [5:17:05<12:51, 16.77s/it]
96%	[REDACTED]	1216/1261 [5:17:21<12:11, 16.26s/it]
97%	[REDACTED]	1217/1261 [5:17:36<11:46, 16.05s/it]
97%	[REDACTED]	1218/1261 [5:17:51<11:10, 15.60s/it]
97%	[REDACTED]	1219/1261 [5:18:05<10:42, 15.29s/it]
97%	[REDACTED]	1220/1261 [5:18:20<10:18, 15.10s/it]
97%	[REDACTED]	1221/1261 [5:18:36<10:14, 15.37s/it]
97%	[REDACTED]	1222/1261 [5:18:51<09:59, 15.37s/it]
97%	[REDACTED]	1223/1261 [5:19:06<09:41, 15.30s/it]
97%	[REDACTED]	1224/1261 [5:19:22<09:33, 15.49s/it]
97%	[REDACTED]	1225/1261 [5:19:39<09:33, 15.94s/it]
97%	[REDACTED]	1226/1261 [5:19:57<09:32, 16.35s/it]
97%	[REDACTED]	1227/1261 [5:20:13<09:14, 16.30s/it]
97%	[REDACTED]	1228/1261 [5:20:28<08:48, 16.01s/it]
97%	[REDACTED]	1229/1261 [5:20:43<08:22, 15.71s/it]
98%	[REDACTED]	1230/1261 [5:20:59<08:05, 15.65s/it]
98%	[REDACTED]	1231/1261 [5:21:15<07:57, 15.90s/it]
98%	[REDACTED]	1232/1261 [5:21:31<07:37, 15.79s/it]
98%	[REDACTED]	1233/1261 [5:21:46<07:20, 15.72s/it]
98%	[REDACTED]	1234/1261 [5:22:02<07:04, 15.74s/it]
98%	[REDACTED]	1235/1261 [5:22:17<06:46, 15.63s/it]
98%	[REDACTED]	1236/1261 [5:22:33<06:31, 15.67s/it]
98%	[REDACTED]	1237/1261 [5:22:48<06:11, 15.48s/it]
98%	[REDACTED]	1238/1261 [5:23:03<05:50, 15.22s/it]
98%	[REDACTED]	1239/1261 [5:23:19<05:43, 15.63s/it]
98%	[REDACTED]	1240/1261 [5:23:36<05:34, 15.95s/it]
98%	[REDACTED]	1241/1261 [5:23:54<05:28, 16.43s/it]
98%	[REDACTED]	1242/1261 [5:24:11<05:18, 16.77s/it]
99%	[REDACTED]	1243/1261 [5:24:29<05:09, 17.21s/it]
99%	[REDACTED]	1244/1261 [5:24:46<04:51, 17.14s/it]