University of Cape Town

APG4011F

Geomatics IV

Individual Assignment

Student:
Moloko M. Mokubedi

Student Number:
MKBMOL006

May 13, 2022

Contents

1 Introduction

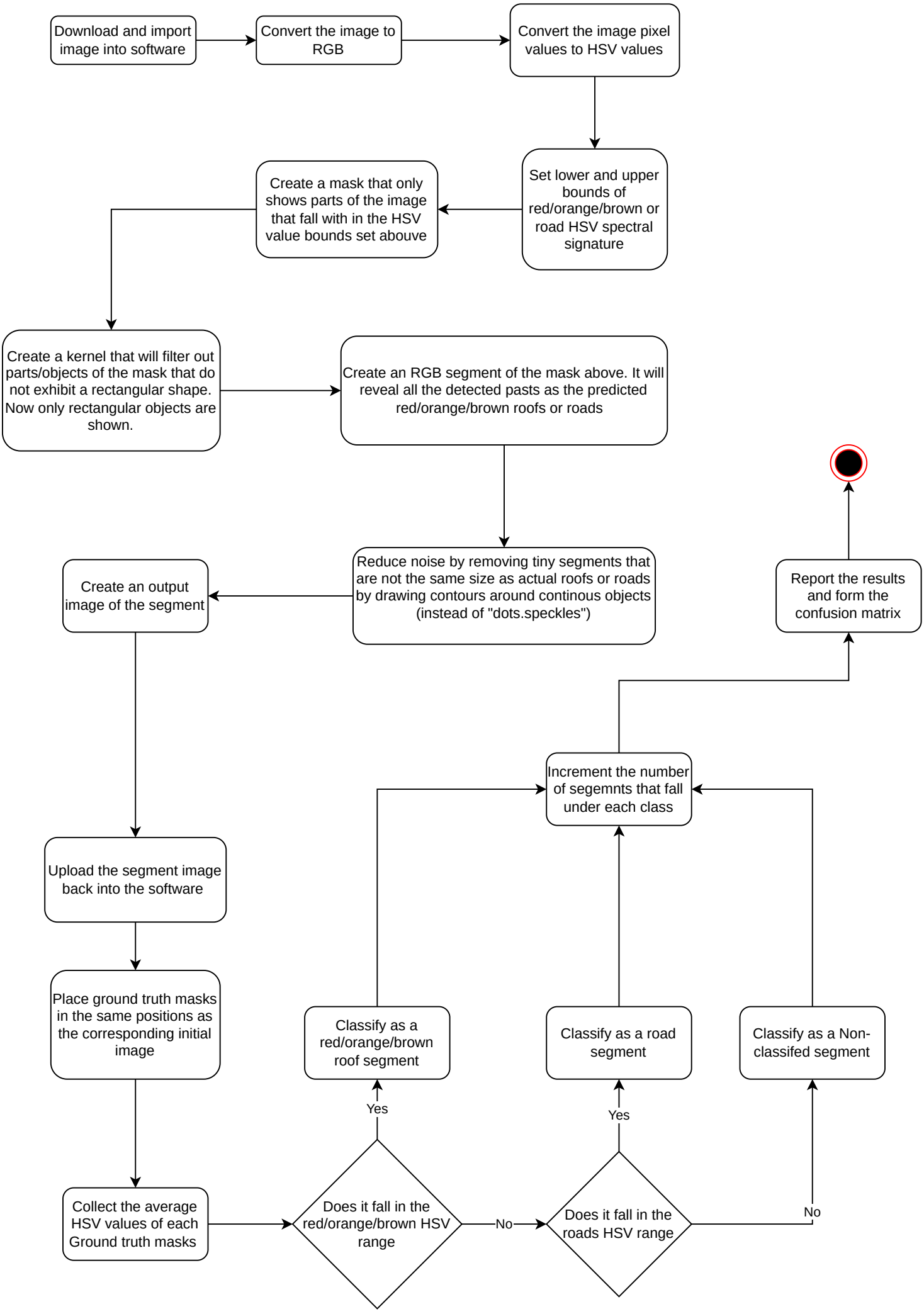
The aim of this Assignment is to demonstrate a ruleset that detects buildings with a red, brown and orange coloured roofs in a provided aerial image. The image that will be used for this image is shown in Figure 1 below:

Figure 1: The original image that would be processed and manipulated in this tutorial

The technology used to accomplish this task is the Python Programming language. The Libraries used are:

1. OpenCV
2. Matplotlib
3. Pandas
4. Numpy
5. Scikit-Image

The rule-set design is shown on the following page


Download and import image into software
Convert the image to RGB
Convert the image pixel values to HSV values
Set lower and upper bounds of red/orange/brown or road HSV spectral signature
Create a mask that only shows parts of the image that fall with in the HSV value bounds set abouve
Create a kernel that will filter out parts/objects of the mask that do not exhibit a rectangular shape. Now only rectangular objects are shown.
Create an RGB segment of the mask above. It will reveal all the detected pasts as the predicted red/orange/brown roofs or roads
Reduce noise by removing tiny segments that are not the same size as actual roofs or roads by drawing contours around continous objects (instead of "dots.speckles")
Create an output image of the segment
Report the results and form the confusion matrix
Upload the segment image back into the software
Increment the number of segemnts that fall under each class
Place ground truth masks in the same positions as the corresponding initial image
Classify as a red/orange/brown roof segment
Classify as a road segment
Classify as a Non-classifed segment
Collect the average HSV values of each Ground truth masks
Does it fall in the red/orange/brown HSV range
Yes
No
Does it fall in the roads HSV range
Yes
No

2 Rule-set Execution and results

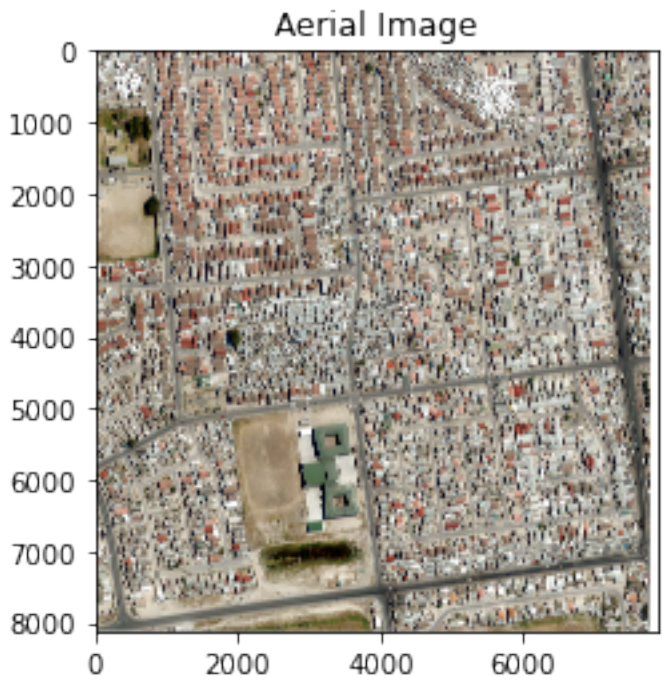
The following pages have the comprehensive Jupyter Notebooks with the outputs of each stage of the rule set displayed. Comments accompany the notebook to explain the execution.

APG4011F Roofs Object Detection

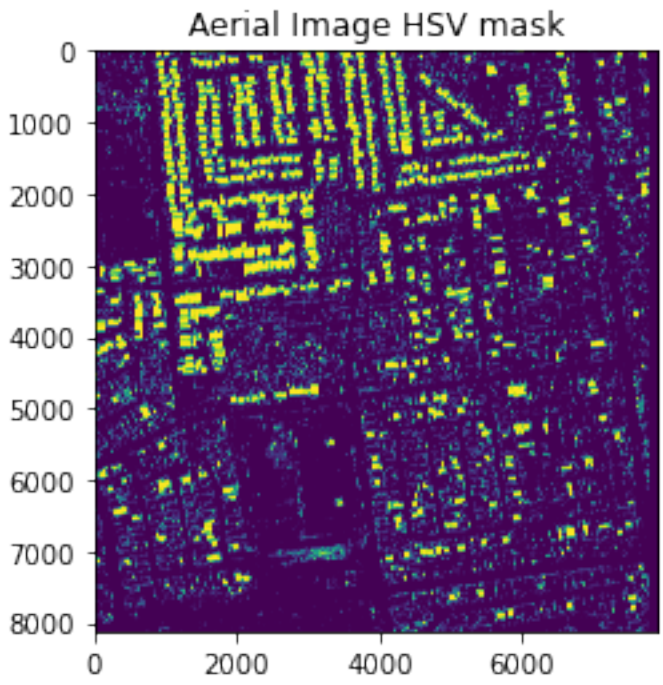
May 13, 2022

[1]:
```
#Object detection of similar color
import cv2
import numpy as np
from matplotlib import pyplot as plt
from skimage import io

# download the image using scikit-image
url = "https://molokomokubedi.s3.af-south-1.amazonaws.com/School/Map4.tif"

print("downloading image from: %s" % (url))

# Reading the image
imgRGB = io.imread(url)
#imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Showing the output
plt.imshow(imgRGB)
plt.title('Aerial Image')
plt.show()
```

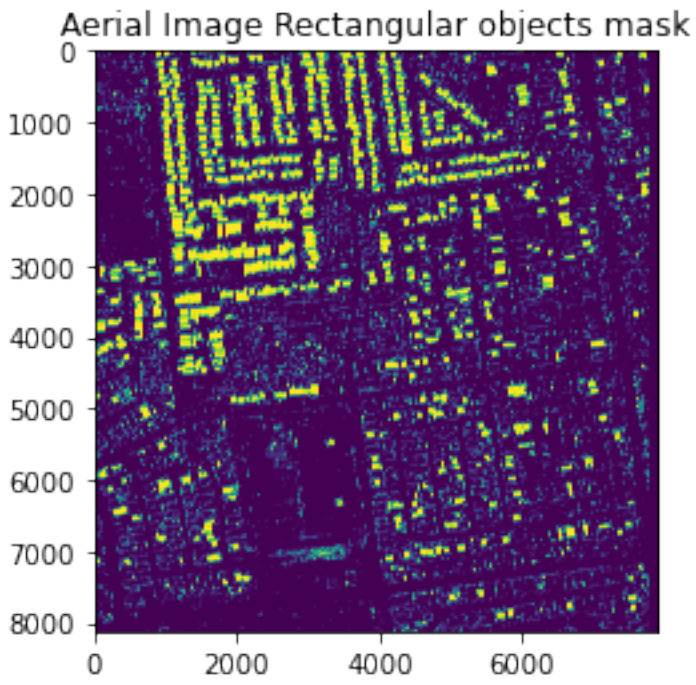
```
downloading image from: https://molokomokubedi.s3.af-
south-1.amazonaws.com/School/Map4.tif
```



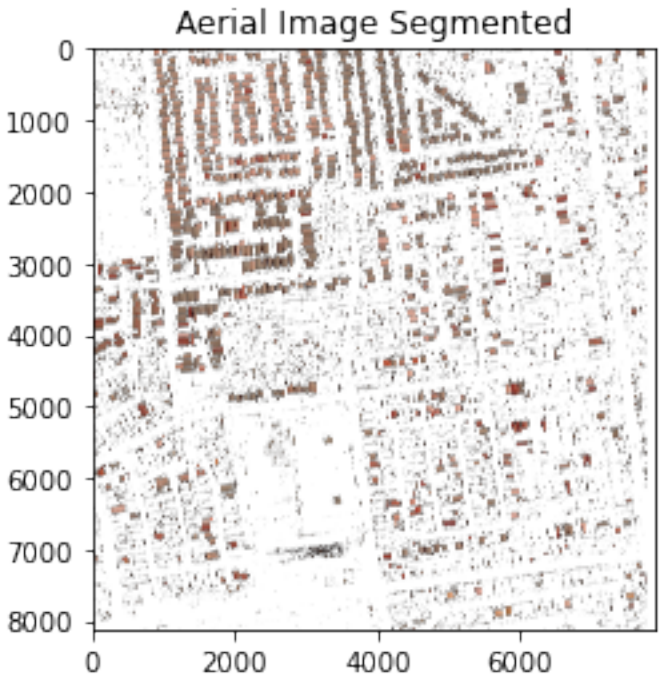
[2]:
```
# convert to hsv colorspace
hsv = cv2.cvtColor(imgRGB, cv2.COLOR_RGB2HSV)

#lower bound and upper bound for colors

#Provided Image bounds
lower_bound = np.array([0, 50, 0])
upper_bound = np.array([15, 255, 255])

#Resort Image bounds
#lower_bound = np.array([0, 0, 0]) #(5,50,50) - (15,255,255)
#upper_bound = np.array([20,255,255])

# find the colors within the boundaries
mask = cv2.inRange(hsv, lower_bound, upper_bound)

plt.imshow(mask)
plt.title('Aerial Image HSV mask')
plt.show()
```

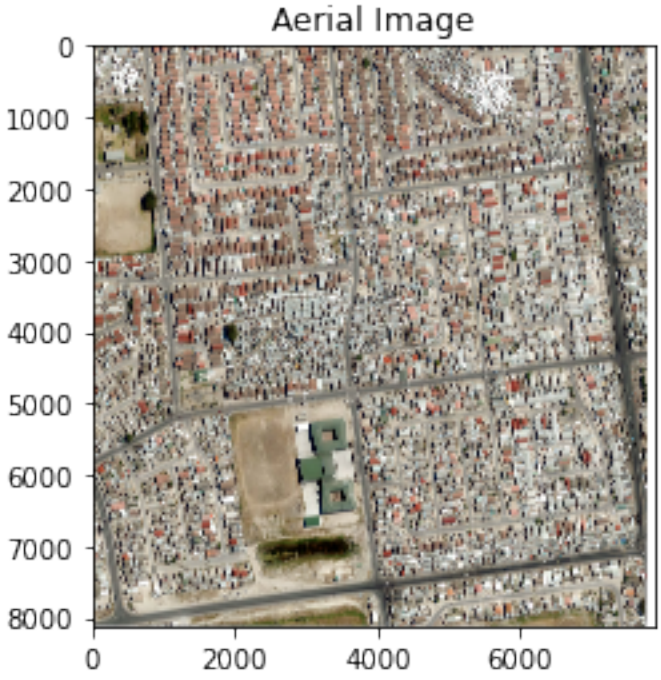


[3]:
```
#define kernel rectangular size
kernel = np.ones((1,1),np.uint8)
#print(kernel)

# Remove unnecessary noise from mask
mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)

plt.imshow(mask)
plt.title('Aerial Image Rectangular objects mask')
plt.show()
```



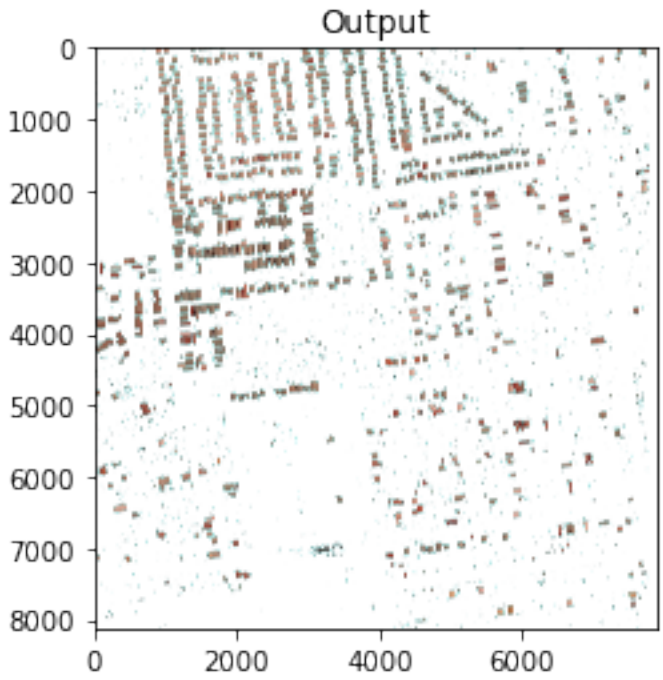
[4]:
```
# Segment only the detected region
segmented_img = cv2.bitwise_and(imgRGB, imgRGB, mask=mask)

plt.imshow(segmented_img)
plt.title('Aerial Image Segmented')
plt.show()

plt.imshow(imgRGB)
plt.title('Aerial Image')
plt.show()
```





[5]:
```
# Find contours from the mask
```

```
contours, hierarchy = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL, cv2.
↪CHAIN_APPROX_SIMPLE)
output = cv2.drawContours(segmented_img, contours, -11, (0, 255, 255), 3)

# Showing the output
plt.imshow(output)
plt.title('Output')
plt.show()
```



[6]:
```
cv2.imwrite("Red_Orange_Browns_Roofs.tif", output)
```

[6]: True

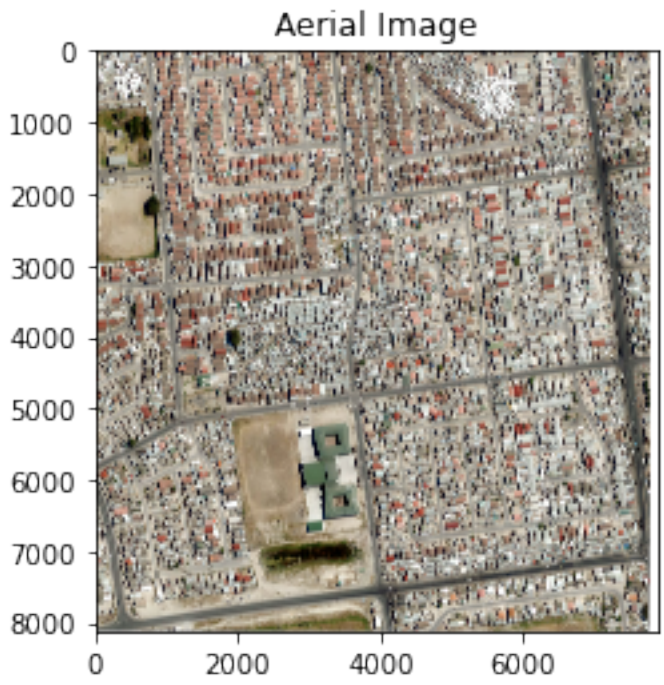
[ ]:

APG4011F Roads Object Detection

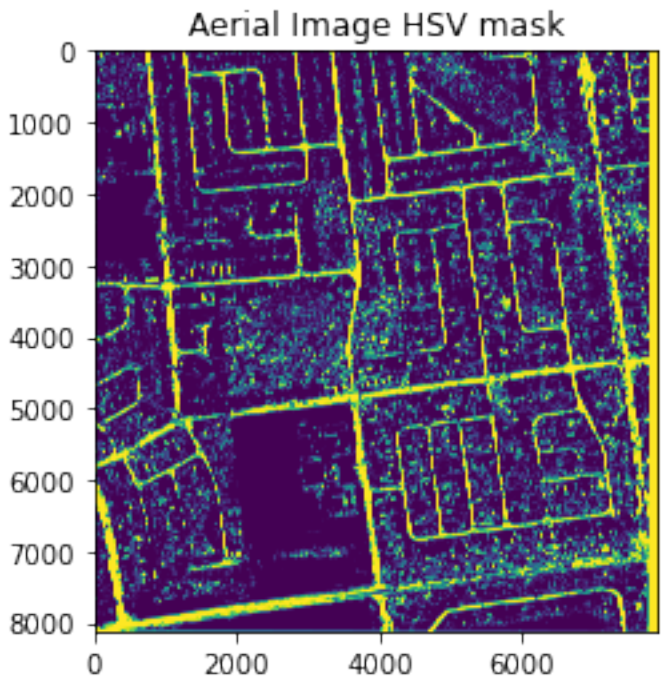
May 13, 2022

[1]:
```
#Object detection of similar color
import cv2
import numpy as np
from matplotlib import pyplot as plt
from skimage import io

# download the image using scikit-image
url = "https://molokomokubedi.s3.af-south-1.amazonaws.com/School/Map4.tif"

print("downloading image from: %s" % (url))

# Reading the image
imgRGB = io.imread(url)
#imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Showing the output
plt.imshow(imgRGB)
plt.title('Aerial Image')
plt.show()
```

```
downloading image from: https://molokomokubedi.s3.af-
south-1.amazonaws.com/School/Map4.tif
```

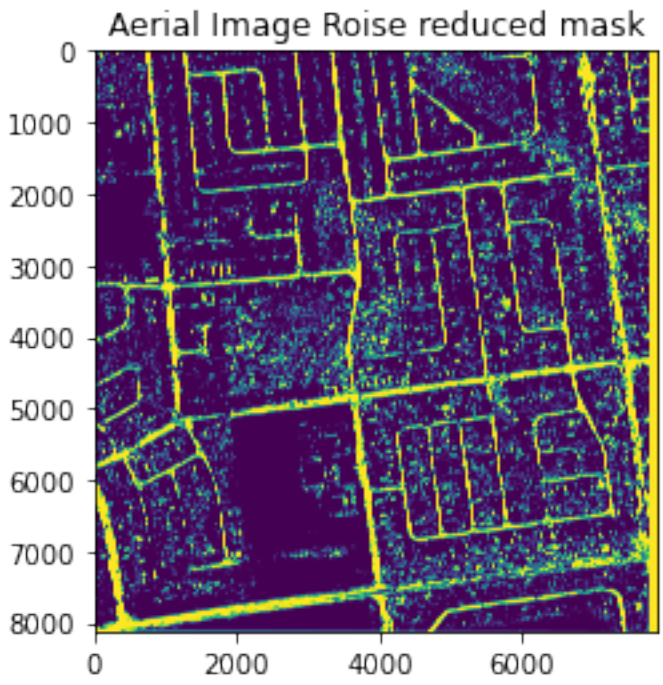


[2]:
```
# convert to hsv colorspace
hsv = cv2.cvtColor(imgRGB, cv2.COLOR_RGB2HSV)

#lower bound and upper bound for colors

#Provided Image bounds
lower_bound = np.array([0, 0, 0])
upper_bound = np.array([100, 30, 180])

#Resort Image bounds
#lower_bound = np.array([0, 0, 0]) #(5,50,50) - (15,255,255)
#upper_bound = np.array([20,255,255])

# find the colors within the boundaries
mask = cv2.inRange(hsv, lower_bound, upper_bound)

plt.imshow(mask)
plt.title('Aerial Image HSV mask')
plt.show()
```

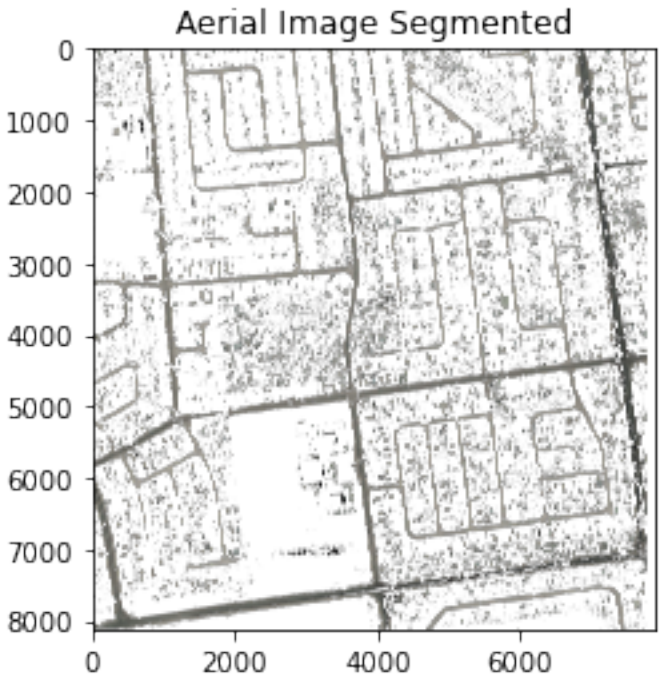


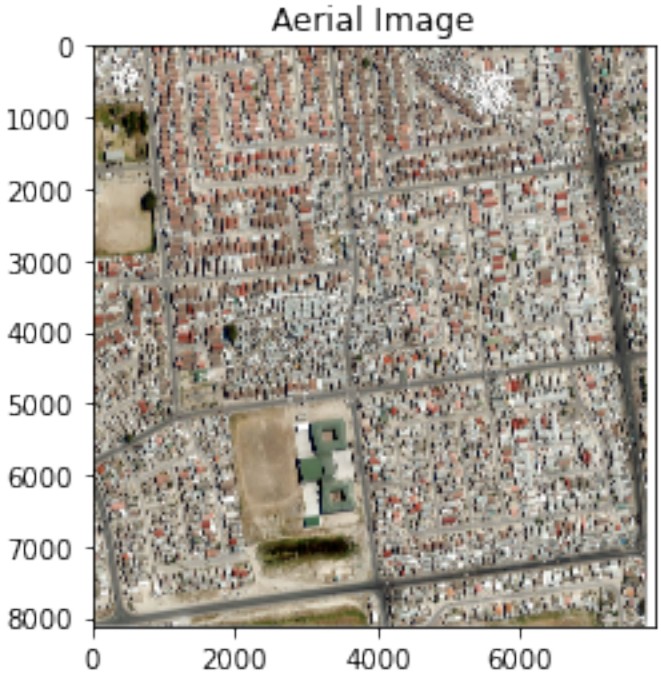
[3]:
```
#define kernel size
kernel = np.ones((2,2),np.uint8)
print(kernel)

# Remove unnecessary noise from mask
mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)

plt.imshow(mask)
plt.title('Aerial Image Roise reduced mask')
plt.show()
```

```
[[1 1]
 [1 1]]
```



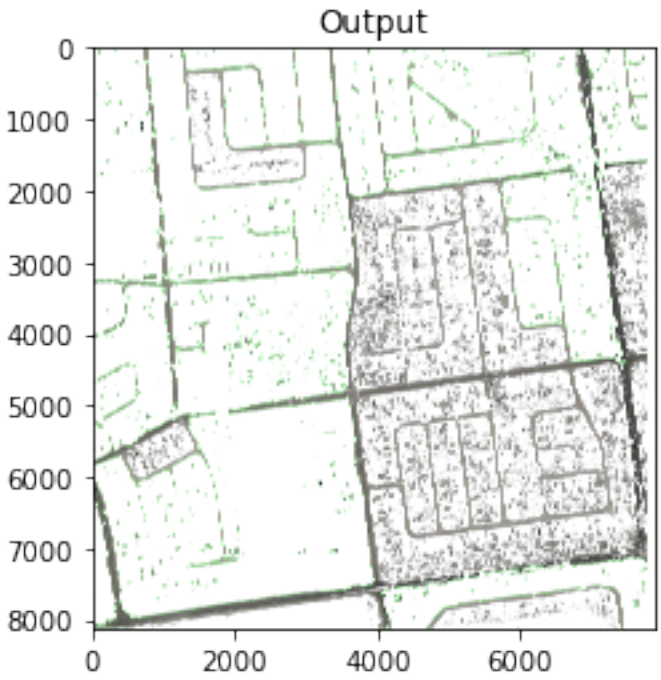
[4]:
```
# Segment only the detected region
segmented_img = cv2.bitwise_and(imgRGB, imgRGB, mask=mask)

plt.imshow(segmented_img)
plt.title('Aerial Image Segmented')
plt.show()

plt.imshow(imgRGB)
plt.title('Aerial Image')
plt.show()
```





[5]:
```
# Find contours from the mask
```

```
contours, hierarchy = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL, cv2.
↪CHAIN_APPROX_SIMPLE)
output = cv2.drawContours(segmented_img, contours, -1, (0, 255, 0), 10)

# Showing the output
plt.imshow(output)
plt.title('Output')
plt.show()
```



[6]:
```
cv2.imwrite("Tarred_Roads.tif", output)
```

[6]: True

Roof_Predictions

May 13, 2022

[8]:
```
import cv2
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np
from skimage import io

# Reading an image in default mode
image = cv2.imread("Red_Orange_Browns_Roofs.tif")

# Window name in which image is displayed
window_name = 'Image'
color = (255, 0, 0)
thickness = 5
font = cv2.FONT_HERSHEY_DUPLEX

segmentNo = []
xcoord = []
ycoord = []
avgHSV = []
Classi = []

# 1 ##############################################################################
print("Sample Ground Truth 1")
segmentNo.append(1)
xcoord.append(2259)
ycoord.append(2255)
pixArray1 = []
for i in range(2255+1, 2337):
        for j in range(2259+1, 2414):
                pix = image[i,j]
                pixArray1.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df1 = pd.DataFrame(pixArray1, columns=['Red','Green','Blue'])
df1.loc['mean'] = df1.mean()
```

```
arrMean1 = [[[]]]
arrMean1[0][0].append(df1.loc['mean'][0])
arrMean1[0][0].append(df1.loc['mean'][1])
arrMean1[0][0].append(df1.loc['mean'][2])

arrMeanNP1 = np.array(arrMean1)
col1 = np.uint8(arrMeanNP1)
print("Mean Segment RGB = ", col1)
colHSV1 = cv2.cvtColor(col1, cv2.COLOR_RGB2HSV)
HSVRanges1 = colHSV1.tolist()
avgHSV.append(HSVRanges1[0][0])
print ("Mean Segment HSV = ", HSVRanges1)
H1 = HSVRanges1[0][0][0]
S1 = HSVRanges1[0][0][1]
V1 = HSVRanges1[0][0][2]
Classification1 = ''
if (H1 > -1 and H1 < 16 and S1 > 49 and S1 < 256 and V1 > -1 and V1 < 256):
        Classification1 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H1 > -1 and H1 < 101 and S1 > 0 and S1 < 31 and V1 > -1 and V1 < 180):
        Classification1 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification1 = 'Not classified'
        print("Segment is non-classified")
start_point1 = (2259, 2255)
end_point1 = (2414, 2337)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification1, start_point1, font, 2, color, 5)
Classi.append(Classification1)

# 2 ##############################################################################
print()
segmentNo.append(2)
xcoord.append(5000)
ycoord.append(2337)
print("Sample Ground Truth 2")
pixArray2 = []
for i in range(1900+1, 1927):
        for j in range(5000+1, 5230):
                pix = image[i,j]
                pixArray2.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df2 = pd.DataFrame(pixArray2, columns=['Red','Green','Blue'])
df2.loc['mean'] = df2.mean()
```

```
arrMean2 = [[[]]]
arrMean2[0][0].append(df2.loc['mean'][0])
arrMean2[0][0].append(df2.loc['mean'][1])
arrMean2[0][0].append(df2.loc['mean'][2])

arrMeanNP2 = np.array(arrMean2)
col2 = np.uint8(arrMeanNP2)
print("Mean Segment RGB = ", col2)
colHSV2 = cv2.cvtColor(col2, cv2.COLOR_RGB2HSV)
HSVRanges2 = colHSV2.tolist()
avgHSV.append(HSVRanges2[0][0])
print ("Mean Segment HSV = ", HSVRanges1)
H2 = HSVRanges2[0][0][0]
S2 = HSVRanges2[0][0][1]
V2 = HSVRanges2[0][0][2]
Classification2 = ''
if (H2 > -1 and H2 < 16 and S2 > 49 and S2 < 256 and V2 > -1 and V2 < 256):
        Classification2 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H2 > -1 and H2 < 101 and S2 > 0 and S2 < 31 and V2 > -1 and V2 < 180):
        Classification2 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification2 = 'Not classified'
        print("Segment is non-classified")
start_point1 = (5000, 1900)
end_point1 = (5230, 1927)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification2, start_point1, font, 2, color, 5)
Classi.append(Classification2)

# 3 ##############################################################################
print()
segmentNo.append(3)
xcoord.append(2993)
ycoord.append(3001)
print("Sample Ground Truth 3")
pixArray3 = []
for i in range(3001+1, 3047):
        for j in range(2993+1, 3085):
                pix = image[i,j]
                pixArray3.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df3 = pd.DataFrame(pixArray3, columns=['Red','Green','Blue'])
```

```
df3.loc['mean'] = df3.mean()

arrMean3 = [[[]]]
arrMean3[0][0].append(df3.loc['mean'][0])
arrMean3[0][0].append(df3.loc['mean'][1])
arrMean3[0][0].append(df3.loc['mean'][2])

arrMeanNP3 = np.array(arrMean3)
col3 = np.uint8(arrMeanNP3)
print("Mean Segment RGB = ", col3)
colHSV3 = cv2.cvtColor(col3, cv2.COLOR_RGB2HSV)
HSVRanges3 = colHSV3.tolist()
avgHSV.append(HSVRanges3[0][0])
print ("Mean Segment HSV = ", HSVRanges3)
H3 = HSVRanges3[0][0][0]
S3 = HSVRanges3[0][0][1]
V3 = HSVRanges3[0][0][2]
Classification3 = ''
if (H3 > -1 and H3 < 16 and S3 > 49 and S3 < 256 and V3 > -1 and V3 < 256):
        Classification3 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H3 > -1 and H3 < 101 and S3 > 0 and S3 < 31 and V3 > -1 and V3 < 180):
        Classification3 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification3 = 'Not classified'
        print("Segment is non-classified")
start_point1 = (2993, 3001)
end_point1 = (3085, 3047)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification3, start_point1, font, 2, color, 5)
Classi.append(Classification3)
# 4 ##############################################################################
print()
print("Sample Ground Truth 4")
pixArray4 = []
for i in range(4047+1, 4082):
        for j in range(1203+1, 1348):
                pix = image[i,j]
                pixArray4.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df4 = pd.DataFrame(pixArray4, columns=['Red','Green','Blue'])
df4.loc['mean'] = df4.mean()
#print(df4.tail())
```

```
arrMean4 = [[[]]]
arrMean4[0][0].append(df4.loc['mean'][0])
arrMean4[0][0].append(df4.loc['mean'][1])
arrMean4[0][0].append(df4.loc['mean'][2])

arrMeanNP4 = np.array(arrMean4)
col4 = np.uint8(arrMeanNP4)
print("Mean Segment RGB = ", col4)
colHSV4 = cv2.cvtColor(col4, cv2.COLOR_RGB2HSV)
HSVRanges4 = colHSV4.tolist()
print ("Mean Segment HSV = ", HSVRanges4)
H4 = HSVRanges4[0][0][0]
S4 = HSVRanges4[0][0][1]
V4 = HSVRanges4[0][0][2]
Classification4 = ''
if (H4 > -1 and H4 < 16 and S4 > 49 and S4 < 256 and V4 > -1 and V4 < 256):
        Classification4 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H4 > -1 and H4 < 101 and S4 > 0 and S4 < 31 and V4 > -1 and V4 < 180):
        Classification4 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification4 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(4)
start_point1 = (1203, 4047)
end_point1 = (1348, 4082)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification4, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges4[0][0])
Classi.append(Classification4)

# 5 ##############################################################################
print()

print("Sample Ground Truth 5")
pixArray5 = []
for i in range(4907+1, 4964):
        for j in range(4156+1, 4229):
                pix = image[i,j]
                pixArray5.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df5 = pd.DataFrame(pixArray5, columns=['Red','Green','Blue'])
```

```
df5.loc['mean'] = df5.mean()
#print(df4.tail())

#print(df4.loc['mean'])

arrMean5 = [[[]]]
arrMean5[0][0].append(df5.loc['mean'][0])
arrMean5[0][0].append(df5.loc['mean'][1])
arrMean5[0][0].append(df5.loc['mean'][2])

arrMeanNP5 = np.array(arrMean5)
col5 = np.uint8(arrMeanNP5)
print("Mean Segment RGB = ", col5)
colHSV5 = cv2.cvtColor(col5, cv2.COLOR_RGB2HSV)
HSVRanges5 = colHSV5.tolist()
print ("Mean Segment HSV = ", HSVRanges5)
H5 = HSVRanges5[0][0][0]
S5 = HSVRanges5[0][0][1]
V5 = HSVRanges5[0][0][2]
Classification5 = ''
if (H5 > -1 and H5 < 16 and S5 > 49 and S5 < 256 and V5 > -1 and V5 < 256):
        Classification5 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H5 > -1 and H5 < 101 and S5 > 0 and S5 < 31 and V5 > -1 and V5 < 180):
        Classification5 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification5 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(5)
start_point1 = (4156, 4907)
end_point1 = (4229, 4964)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification5, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges5[0][0])
Classi.append(Classification5)


# 6 ##############################################################################
print()
print("Sample Ground Truth 6")
pixArray6 = []
for i in range(7133+1, 7199):
        for j in range(6740+1, 6946):
                pix = image[i,j]
```

```
                pixArray6.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df6 = pd.DataFrame(pixArray6, columns=['Red','Green','Blue'])
df6.loc['mean'] = df6.mean()
#print(df4.tail())

arrMean6 = [[[]]]
arrMean6[0][0].append(df6.loc['mean'][0])
arrMean6[0][0].append(df6.loc['mean'][1])
arrMean6[0][0].append(df6.loc['mean'][2])

arrMeanNP6 = np.array(arrMean6)
col6 = np.uint8(arrMeanNP6)
print("Mean Segment RGB = ", col6)
colHSV6 = cv2.cvtColor(col6, cv2.COLOR_RGB2HSV)
HSVRanges6 = colHSV6.tolist()
print ("Mean Segment HSV = ", HSVRanges6)
H6 = HSVRanges6[0][0][0]
S6 = HSVRanges6[0][0][1]
V6 = HSVRanges6[0][0][2]
Classification6 = ''
if (H6 > -1 and H6 < 16 and S6 > 49 and S6 < 256 and V6 > -1 and V6 < 256):
        Classification6 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H6 > -1 and H6 < 101 and S6 > 0 and S6 < 31 and V6 > -1 and V6 < 180):
        Classification6 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification6 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(6)
start_point1 = (6740, 7133)
end_point1 = (6946, 7199)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification6, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges6[0][0])
Classi.append(Classification6)

# 7 ##############################################################################
print()
print("Sample Ground Truth 7")
pixArray7 = []
for i in range(5731+1, 5881):
```

```
        for j in range(2376+1, 2648):
                pix = image[i,j]
                pixArray7.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df7 = pd.DataFrame(pixArray7, columns=['Red','Green','Blue'])
df7.loc['mean'] = df7.mean()
#print(df4.tail())

arrMean7 = [[[]]]
arrMean7[0][0].append(df7.loc['mean'][0])
arrMean7[0][0].append(df7.loc['mean'][1])
arrMean7[0][0].append(df7.loc['mean'][2])

arrMeanNP7 = np.array(arrMean7)
col7 = np.uint8(arrMeanNP7)
print("Mean Segment RGB = ", col7)
colHSV7 = cv2.cvtColor(col7, cv2.COLOR_RGB2HSV)
HSVRanges7 = colHSV7.tolist()
print ("Mean Segment HSV = ", HSVRanges7)
H7 = HSVRanges7[0][0][0]
S7 = HSVRanges7[0][0][1]
V7 = HSVRanges7[0][0][2]
Classification7 = ''
if (H7 > -1 and H7 < 16 and S7 > 49 and S7 < 256 and V7 > -1 and V7 < 256):
        Classification7 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H7 > -1 and H7 < 101 and S7 > 0 and S7 < 31 and V7 > -1 and V7 < 180):
        Classification7 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification7 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(7)
start_point1 = (2376, 5731)
end_point1 = (2648, 5881)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification7, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges7[0][0])
Classi.append(Classification7)

# 8 ##############################################################################
print()
print("Sample Ground Truth 8")
```

```
pixArray8 = []
for i in range(5216+1, 5349):
        for j in range(5932+1, 5984):
                pix = image[i,j]
                pixArray8.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df8 = pd.DataFrame(pixArray8, columns=['Red','Green','Blue'])
df8.loc['mean'] = df8.mean()
#print(df4.tail())

arrMean8 = [[[]]]
arrMean8[0][0].append(df8.loc['mean'][0])
arrMean8[0][0].append(df8.loc['mean'][1])
arrMean8[0][0].append(df8.loc['mean'][2])

arrMeanNP8 = np.array(arrMean8)
col8 = np.uint8(arrMeanNP8)
print("Mean Segment RGB = ", col8)
colHSV8 = cv2.cvtColor(col8, cv2.COLOR_RGB2HSV)
HSVRanges8 = colHSV8.tolist()
print ("Mean Segment HSV = ", HSVRanges8)
H8 = HSVRanges8[0][0][0]
S8 = HSVRanges8[0][0][1]
V8 = HSVRanges8[0][0][2]
Classification8 = ''
if (H8 > -1 and H8 < 16 and S8 > 49 and S8 < 256 and V8 > -1 and V8 < 256):
        Classification8 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H8 > -1 and H8 < 101 and S8 > 0 and S8 < 31 and V8 > -1 and V8 < 180):
        Classification8 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification8 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(8)
start_point1 = (5932, 5216)
end_point1 = (5984, 5349)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification8, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges8[0][0])
Classi.append(Classification8)

# 9 ##############################################################################
```

```
print()
print("Sample Ground Truth 9")
pixArray9 = []
for i in range(4415+1, 4485):
        for j in range(697+1, 782):
                pix = image[i,j]
                pixArray9.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df9 = pd.DataFrame(pixArray9, columns=['Red','Green','Blue'])
df9.loc['mean'] = df9.mean()
#print(df4.tail())

arrMean9 = [[[]]]
arrMean9[0][0].append(df9.loc['mean'][0])
arrMean9[0][0].append(df9.loc['mean'][1])
arrMean9[0][0].append(df9.loc['mean'][2])

arrMeanNP9 = np.array(arrMean9)
col9 = np.uint8(arrMeanNP9)
print("Mean Segment RGB = ", col9)
colHSV9 = cv2.cvtColor(col9, cv2.COLOR_RGB2HSV)
HSVRanges9 = colHSV9.tolist()
print ("Mean Segment HSV = ", HSVRanges9)
H9 = HSVRanges9[0][0][0]
S9 = HSVRanges9[0][0][1]
V9 = HSVRanges9[0][0][2]
Classification9 = ''
if (H9 > -1 and H9 < 16 and S9 > 49 and S9 < 256 and V9 > -1 and V9 < 256):
        Classification9 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H9 > -1 and H9 < 101 and S9 > 0 and S9 < 31 and V9 > -1 and V9 < 180):
        Classification9 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification9 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(9)
start_point1 = (697, 4415)
end_point1 = (782, 4485)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification9, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges9[0][0])
Classi.append(Classification9)
```

```
# 10␣
↪#############################################################################
print()

print()
print("Sample Ground Truth 10")
pixArray10 = []
for i in range(784+1, 910):
        for j in range(4525+1, 4658):
                pix = image[i,j]
                pixArray10.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df10 = pd.DataFrame(pixArray10, columns=['Red','Green','Blue'])
df10.loc['mean'] = df10.mean()
#print(df4.tail())

arrMean10 = [[[]]]
arrMean10[0][0].append(df10.loc['mean'][0])
arrMean10[0][0].append(df10.loc['mean'][1])
arrMean10[0][0].append(df10.loc['mean'][2])

arrMeanNP10 = np.array(arrMean10)
col10 = np.uint8(arrMeanNP10)
print("Mean Segment RGB = ", col10)
colHSV10 = cv2.cvtColor(col10, cv2.COLOR_RGB2HSV)
HSVRanges10 = colHSV10.tolist()
print ("Mean Segment HSV = ", HSVRanges10)
H10 = HSVRanges10[0][0][0]
S10 = HSVRanges10[0][0][1]
V10 = HSVRanges10[0][0][2]
Classification10 = ''
if (H10 > -1 and H10 < 16 and S10 > 49 and S10 < 256 and V10 > -1 and V10 <␣
↪256):
        Classification10 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H10 > -1 and H10 < 101 and S10 > 0 and S10 < 31 and V10 > -1 and V10 <␣
↪180):
        Classification10 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification10 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(10)
start_point1 = (4525, 784)
```

```
end_point1 = (4658, 910)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification10, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges10[0][0])
Classi.append(Classification10)

# 11 ##############################################################################
print()
print("Sample Ground Truth 11")
pixArray11 = []
for i in range(2094+1, 2188):
        for j in range(3550+1, 3649):
                pix = image[i,j]
                pixArray11.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df11 = pd.DataFrame(pixArray11, columns=['Red','Green','Blue'])
df11.loc['mean'] = df11.mean()
#print(df4.tail())

arrMean11 = [[[]]]
arrMean11[0][0].append(df11.loc['mean'][0])
arrMean11[0][0].append(df11.loc['mean'][1])
arrMean11[0][0].append(df11.loc['mean'][2])

arrMeanNP11 = np.array(arrMean11)
col11 = np.uint8(arrMeanNP11)
print("Mean Segment RGB = ", col11)
colHSV11 = cv2.cvtColor(col11, cv2.COLOR_RGB2HSV)
HSVRanges11 = colHSV11.tolist()
print ("Mean Segment HSV = ", HSVRanges11)
H11 = HSVRanges11[0][0][0]
S11 = HSVRanges11[0][0][1]
V11 = HSVRanges11[0][0][2]
Classification11 = ''
if (H11 > -1 and H11 < 16 and S11 > 49 and S11 < 256 and V11 > -1 and V11 <␣
↪256):
        Classification10 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H11 > -1 and H11 < 101 and S11 > 0 and S11 < 31 and V11 > -1 and V11 <␣
↪180):
        Classification11 = 'Tarred Road'
        print("Segment is a tarred road")
else:
```

```
        Classification11 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(11)
start_point1 = (3550, 2094)
end_point1 = (3649, 2188)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification11, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges11[0][0])
Classi.append(Classification11)

# 12 ##############################################################################
print()
print("Sample Ground Truth 12")
pixArray12 = []
for i in range(6942+1, 7099):
        for j in range(2966+1, 3520):
                pix = image[i,j]
                pixArray12.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df12 = pd.DataFrame(pixArray12, columns=['Red','Green','Blue'])
df12.loc['mean'] = df12.mean()
#print(df4.tail())

arrMean12 = [[[]]]
arrMean12[0][0].append(df12.loc['mean'][0])
arrMean12[0][0].append(df12.loc['mean'][1])
arrMean12[0][0].append(df12.loc['mean'][2])

arrMeanNP12 = np.array(arrMean12)
col12 = np.uint8(arrMeanNP12)
print("Mean Segment RGB = ", col12)
colHSV12 = cv2.cvtColor(col12, cv2.COLOR_RGB2HSV)
HSVRanges12 = colHSV12.tolist()
print ("Mean Segment HSV = ", HSVRanges12)
H12 = HSVRanges12[0][0][0]
S12 = HSVRanges12[0][0][1]
V12 = HSVRanges12[0][0][2]
Classification12 = ''
if (H12 > -1 and H12 < 15 and S12 > 49 and S12 < 256 and V12 > -1 and V12 <␣
 ↪256):
        Classification12 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
```

```
elif (H12 > -1 and H12 < 101 and S12 > 0 and S12 < 31 and V12 > -1 and V12 <␣
 ↪180):
        Classification12 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification12 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(12)
start_point1 = (2966, 6942)
end_point1 = (3520, 7099)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification12, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges12[0][0])
Classi.append(Classification12)

# 13 ##############################################################################
print()
print("Sample Ground Truth 13")
pixArray13 = []
for i in range(567+1, 615):
        for j in range(2131+1, 2215):
                pix = image[i,j]
                pixArray13.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df13 = pd.DataFrame(pixArray13, columns=['Red','Green','Blue'])
df13.loc['mean'] = df13.mean()
#print(df4.tail())

arrMean13 = [[[]]]
arrMean13[0][0].append(df13.loc['mean'][0])
arrMean13[0][0].append(df13.loc['mean'][1])
arrMean13[0][0].append(df13.loc['mean'][2])

arrMeanNP13 = np.array(arrMean13)
col13 = np.uint8(arrMeanNP13)
print("Mean Segment RGB = ", col13)
colHSV13 = cv2.cvtColor(col13, cv2.COLOR_RGB2HSV)
HSVRanges13 = colHSV13.tolist()
print ("Mean Segment HSV = ", HSVRanges13)
H13 = HSVRanges13[0][0][0]
S13 = HSVRanges13[0][0][1]
V13 = HSVRanges13[0][0][2]
Classification13 = ''
```

```
if (H13 > -1 and H13 < 16 and S13 > 49 and S13 < 256 and V13 > -1 and V13 <␣
 ↪256):
        Classification13 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H13 > -1 and H13 < 101 and S13 > 0 and S13 < 31 and V13 > -1 and V13 <␣
 ↪180):
        Classification13 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification13 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(13)
start_point1 = (2131, 567)
end_point1 = (2215, 615)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification13, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges13[0][0])
Classi.append(Classification13)
# 14 ##############################################################################
print()
print("Sample Ground Truth 14")
pixArray14 = []
for i in range(2354+1, 2417):
        for j in range(1088+1, 1220):
                pix = image[i,j]
                pixArray14.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df14 = pd.DataFrame(pixArray14, columns=['Red','Green','Blue'])
df14.loc['mean'] = df14.mean()
#print(df4.tail())

arrMean14 = [[[]]]
arrMean14[0][0].append(df14.loc['mean'][0])
arrMean14[0][0].append(df14.loc['mean'][1])
arrMean14[0][0].append(df14.loc['mean'][2])

arrMeanNP14 = np.array(arrMean14)
col14 = np.uint8(arrMeanNP14)
print("Mean Segment RGB = ", col14)
colHSV14 = cv2.cvtColor(col14, cv2.COLOR_RGB2HSV)
HSVRanges14 = colHSV14.tolist()
print ("Mean Segment HSV = ", HSVRanges14)
H14 = HSVRanges14[0][0][0]
```

```
S14 = HSVRanges14[0][0][1]
V14 = HSVRanges14[0][0][2]
Classification14 = ''
if (H14 > -1 and H14 < 16 and S14 > 49 and S14 < 256 and V14 > -1 and V14 <␣
 ↪256):
        Classification14 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H14 > -1 and H14 < 101 and S14 > 0 and S14 < 31 and V14 > -1 and V14 <␣
 ↪180):
        Classification14 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification14 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(14)
start_point1 = (1088, 2354)
end_point1 = (1220, 2417)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification14, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges14[0][0])
Classi.append(Classification14)
# 15 ##########################################################################
print()
print("Sample Ground Truth 15")
pixArray15 = []
for i in range(7540+1, 7614):
        for j in range(5352+1, 5425):
                pix = image[i,j]
                pixArray15.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df15 = pd.DataFrame(pixArray15, columns=['Red','Green','Blue'])
df15.loc['mean'] = df15.mean()
#print(df4.tail())

arrMean15 = [[[]]]
arrMean15[0][0].append(df15.loc['mean'][0])
arrMean15[0][0].append(df15.loc['mean'][1])
arrMean15[0][0].append(df15.loc['mean'][2])

arrMeanNP15 = np.array(arrMean15)
col15 = np.uint8(arrMeanNP15)
print("Mean Segment RGB = ", col15)
colHSV15 = cv2.cvtColor(col15, cv2.COLOR_RGB2HSV)
```

```
HSVRanges15 = colHSV15.tolist()
print ("Mean Segment HSV = ", HSVRanges15)
H15 = HSVRanges15[0][0][0]
S15 = HSVRanges15[0][0][1]
V15 = HSVRanges15[0][0][2]
Classification15 = ''
if (H15 > -1 and H15 < 16 and S15 > 49 and S15 < 256 and V15 > -1 and V15 <␣
 ↪256):
        Classification15 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H15 > -1 and H15 < 101 and S15 > 0 and S15 < 31 and V15 > -1 and V15 <␣
 ↪180):
        Classification15 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification15 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(15)
start_point1 = (5352, 7540)
end_point1 = (5425, 7614)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification15, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges15[0][0])
Classi.append(Classification15)

# 16 ##############################################################################
print()
print("Sample Ground Truth 16")
pixArray16 = []
for i in range(6792+1, 6823):
        for j in range(873+1, 980):
                pix = image[i,j]
                pixArray16.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df16 = pd.DataFrame(pixArray16, columns=['Red','Green','Blue'])
df16.loc['mean'] = df16.mean()
#print(df4.tail())

arrMean16 = [[[]]]
arrMean16[0][0].append(df16.loc['mean'][0])
arrMean16[0][0].append(df16.loc['mean'][1])
arrMean16[0][0].append(df16.loc['mean'][2])
```

```
arrMeanNP16 = np.array(arrMean16)
col16 = np.uint8(arrMeanNP16)
print("Mean Segment RGB = ", col16)
colHSV16 = cv2.cvtColor(col16, cv2.COLOR_RGB2HSV)
HSVRanges16 = colHSV16.tolist()
print ("Mean Segment HSV = ", HSVRanges16)
H16 = HSVRanges16[0][0][0]
S16 = HSVRanges16[0][0][1]
V16 = HSVRanges16[0][0][2]
Classification16 = ''
if (H16 > -1 and H16 < 16 and S16 > 49 and S16 < 256 and V16 > -1 and V16 <␣
 ↪256):
        Classification16 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H16 > -1 and H16 < 101 and S16 > 0 and S16 < 31 and V16 > -1 and V16 <␣
 ↪180):
        Classification16 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification16 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(16)
start_point1 = (873, 6792)
end_point1 = (980, 6823)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification16, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges16[0][0])
Classi.append(Classification16)

# 17 ##############################################################################
print()
print("Sample Ground Truth 17")
pixArray17 = []
for i in range(753+1, 821):
        for j in range(7120+1, 7216):
                pix = image[i,j]
                pixArray17.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df17 = pd.DataFrame(pixArray17, columns=['Red','Green','Blue'])
df17.loc['mean'] = df17.mean()
#print(df4.tail())

arrMean17 = [[[]]]
```

```
arrMean17[0][0].append(df17.loc['mean'][0])
arrMean17[0][0].append(df17.loc['mean'][1])
arrMean17[0][0].append(df17.loc['mean'][2])

arrMeanNP17 = np.array(arrMean17)
col17 = np.uint8(arrMeanNP17)
print("Mean Segment RGB = ", col17)
colHSV17 = cv2.cvtColor(col17, cv2.COLOR_RGB2HSV)
HSVRanges17 = colHSV17.tolist()
print ("Mean Segment HSV = ", HSVRanges17)
H17 = HSVRanges17[0][0][0]
S17 = HSVRanges17[0][0][1]
V17 = HSVRanges17[0][0][2]
Classification17 = ''
if (H17 > -1 and H17 < 16 and S17 > 49 and S17 < 256 and V17 > -1 and V17 <␣
 ↪256):
        Classification17 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H17 > -1 and H17 < 101 and S17 > 0 and S17 < 31 and V17 > -1 and V17 <␣
 ↪180):
        Classification17 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification17 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(17)
start_point1 = (7120, 753)
end_point1 = (7216, 821)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification17, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges17[0][0])
Classi.append(Classification17)

# 18 ##############################################################################
print()
print("Sample Ground Truth 18")
pixArray18 = []
for i in range(1029+1, 1093):
        for j in range(3583+1, 3700):
                pix = image[i,j]
                pixArray18.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df18 = pd.DataFrame(pixArray18, columns=['Red','Green','Blue'])
```

```
df18.loc['mean'] = df18.mean()
#print(df4.tail())

arrMean18 = [[[]]]
arrMean18[0][0].append(df18.loc['mean'][0])
arrMean18[0][0].append(df18.loc['mean'][1])
arrMean18[0][0].append(df18.loc['mean'][2])

arrMeanNP18 = np.array(arrMean18)
col18 = np.uint8(arrMeanNP18)
print("Mean Segment RGB = ", col18)
colHSV18 = cv2.cvtColor(col18, cv2.COLOR_RGB2HSV)
HSVRanges18 = colHSV18.tolist()
print ("Mean Segment HSV = ", HSVRanges18)
H18 = HSVRanges18[0][0][0]
S18 = HSVRanges18[0][0][1]
V18 = HSVRanges18[0][0][2]
Classification18 = ''
if (H18 > -1 and H18 < 16 and S18 > 49 and S18 < 256 and V18 > -1 and V18 <␣
 ↪256):
        Classification18 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H18 > -1 and H18 < 101 and S18 > 0 and S18 < 31 and V18 > -1 and V18 <␣
 ↪180):
        Classification18 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification18 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(18)
start_point1 = (3583, 1029)
end_point1 = (3700, 1093)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification18, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges18[0][0])
Classi.append(Classification18)

# 19 ##############################################################################
print()
print("Sample Ground Truth 19")
pixArray19 = []
for i in range(2967+1, 3028):
        for j in range(4030+1, 4132):
                pix = image[i,j]
                pixArray19.append(pix.tolist())
```

```
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df19 = pd.DataFrame(pixArray19, columns=['Red','Green','Blue'])
df19.loc['mean'] = df19.mean()
#print(df4.tail())

arrMean19 = [[[]]]
arrMean19[0][0].append(df19.loc['mean'][0])
arrMean19[0][0].append(df19.loc['mean'][1])
arrMean19[0][0].append(df19.loc['mean'][2])

arrMeanNP19 = np.array(arrMean19)
col19 = np.uint8(arrMeanNP19)
print("Mean Segment RGB = ", col19)
colHSV19 = cv2.cvtColor(col19, cv2.COLOR_RGB2HSV)
HSVRanges19 = colHSV19.tolist()
print ("Mean Segment HSV = ", HSVRanges19)
H19 = HSVRanges19[0][0][0]
S19 = HSVRanges19[0][0][1]
V19 = HSVRanges19[0][0][2]
Classification19 = ''
if (H19 > -1 and H19 < 16 and S19 > 49 and S19 < 256 and V19 > -1 and V19 <␣
 ↪256):
        Classification19 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H19 > -1 and H19 < 101 and S19 > 0 and S19 < 31 and V19 > -1 and V19 <␣
 ↪180):
        Classification19 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification19 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(19)
start_point1 = (4030, 2967)
end_point1 = (4132, 3028)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification19, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges19[0][0])
Classi.append(Classification19)
# 20 ##########################################################################
print()

print("Sample Ground Truth 20")
pixArray20 = []
```

```
for i in range(7897+1, 8104):
        for j in range(3283+1, 3712):
                pix = image[i,j]
                pixArray20.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df20 = pd.DataFrame(pixArray20, columns=['Red','Green','Blue'])
df20.loc['mean'] = df20.mean()
#print(df4.tail())

arrMean20 = [[[]]]
arrMean20[0][0].append(df20.loc['mean'][0])
arrMean20[0][0].append(df20.loc['mean'][1])
arrMean20[0][0].append(df20.loc['mean'][2])

arrMeanNP20 = np.array(arrMean20)
col20 = np.uint8(arrMeanNP20)
print("Mean Segment RGB = ", col20)
colHSV20 = cv2.cvtColor(col20, cv2.COLOR_RGB2HSV)
HSVRanges20 = colHSV20.tolist()
print ("Mean Segment HSV = ", HSVRanges20)
H20 = HSVRanges20[0][0][0]
S20 = HSVRanges20[0][0][1]
V20 = HSVRanges20[0][0][2]
Classification20 = ''
if (H20 > -1 and H20 < 16 and S20 > 49 and S20 < 256 and V20 > -1 and V20 <␣
 ↪256):
        Classification20 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H20 > -1 and H20 < 101 and S20 > 0 and S20 < 31 and V20 > -1 and V20 <␣
 ↪180):
        Classification20 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification20 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(20)
start_point1 = (3283, 7897)
end_point1 = (3712, 8104)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification20, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges20[0][0])
Classi.append(Classification20)
```

```
# 21 ##############################################################################
print()
print("Sample Ground Truth 21")
pixArray21 = []
for i in range(4912+1, 5060):
        for j in range(6756+1, 6828):
                pix = image[i,j]
                pixArray21.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df21 = pd.DataFrame(pixArray21, columns=['Red','Green','Blue'])
df21.loc['mean'] = df21.mean()
#print(df4.tail())

arrMean21 = [[[]]]
arrMean21[0][0].append(df21.loc['mean'][0])
arrMean21[0][0].append(df21.loc['mean'][1])
arrMean21[0][0].append(df21.loc['mean'][2])

arrMeanNP21 = np.array(arrMean21)
col21 = np.uint8(arrMeanNP21)
print("Mean Segment RGB = ", col21)
colHSV21 = cv2.cvtColor(col21, cv2.COLOR_RGB2HSV)
HSVRanges21 = colHSV21.tolist()
print ("Mean Segment HSV = ", HSVRanges21)
H21 = HSVRanges21[0][0][0]
S21 = HSVRanges21[0][0][1]
V21 = HSVRanges21[0][0][2]
Classification21 = ''
if (H21 > -1 and H21 < 16 and S21 > 49 and S21 < 256 and V21 > -1 and V21 <␣
 ↪256):
        Classification21 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H21 > -1 and H21 < 101 and S21 > 0 and S21 < 31 and V21 > -1 and V21 <␣
 ↪180):
        Classification21 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification21 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(21)
start_point1 = (6756, 4912)
end_point1 = (6828, 5060)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification21, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
```

```
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges21[0][0])
Classi.append(Classification21)

# 22 ##############################################################################
print()
print("Sample Ground Truth 22")
pixArray22 = []
for i in range(6444+1, 6538):
        for j in range(4797+1,4833):
                pix = image[i,j]
                pixArray22.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df22 = pd.DataFrame(pixArray22, columns=['Red','Green','Blue'])
df22.loc['mean'] = df22.mean()
#print(df4.tail())

arrMean22 = [[[]]]
arrMean22[0][0].append(df22.loc['mean'][0])
arrMean22[0][0].append(df22.loc['mean'][1])
arrMean22[0][0].append(df22.loc['mean'][2])

arrMeanNP22 = np.array(arrMean22)
col22 = np.uint8(arrMeanNP22)
print("Mean Segment RGB = ", col22)
colHSV22 = cv2.cvtColor(col22, cv2.COLOR_RGB2HSV)
HSVRanges22 = colHSV22.tolist()
print ("Mean Segment HSV = ", HSVRanges22)
H22 = HSVRanges22[0][0][0]
S22 = HSVRanges22[0][0][1]
V22 = HSVRanges22[0][0][2]
Classification22 = ''
if (H22 > -1 and H22 < 16 and S22 > 49 and S22 < 256 and V22 > -1 and V22 <␣
 ↪256):
        Classification22 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H22 > -1 and H22 < 101 and S22 > 0 and S22 < 31 and V22 > -1 and V22 <␣
 ↪180):
        Classification22 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification22 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(22)
start_point1 = (4797, 6444)
```

```
end_point1 = (4833, 6538)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification22, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges22[0][0])
Classi.append(Classification22)

# 23 ##############################################################################
print()
print("Sample Ground Truth 23")
pixArray23 = []
for i in range(7226+1, 7460):
        for j in range(226+1,336):
                pix = image[i,j]
                pixArray23.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df23 = pd.DataFrame(pixArray23, columns=['Red','Green','Blue'])
df23.loc['mean'] = df23.mean()
#print(df4.tail())

arrMean23 = [[[]]]
arrMean23[0][0].append(df23.loc['mean'][0])
arrMean23[0][0].append(df23.loc['mean'][1])
arrMean23[0][0].append(df23.loc['mean'][2])

arrMeanNP23 = np.array(arrMean23)
col23 = np.uint8(arrMeanNP23)
print("Mean Segment RGB = ", col23)
colHSV23 = cv2.cvtColor(col23, cv2.COLOR_RGB2HSV)
HSVRanges23 = colHSV23.tolist()
print ("Mean Segment HSV = ", HSVRanges23)
H23 = HSVRanges23[0][0][0]
S23 = HSVRanges23[0][0][1]
V23 = HSVRanges23[0][0][2]
Classification23 = ''
if (H23 > -1 and H23 < 16 and S23 > 49 and S23 < 256 and V23 > -1 and V23 <␣
 ↪256):
        Classification23 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H23 > -1 and H23 < 101 and S23 > 0 and S23 < 31 and V23 > -1 and V23 <␣
 ↪180):
        Classification23 = 'Tarred Road'
        print("Segment is a tarred road")
else:
```

```
        Classification23 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(23)
start_point1 = (226, 7226)
end_point1 = (336, 7460)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification23, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges23[0][0])
Classi.append(Classification23)
# 24 ##############################################################################
print()
print("Sample Ground Truth 24")
pixArray24 = []
for i in range(5392+1, 5439):
        for j in range(4297+1,4437):
                pix = image[i,j]
                pixArray24.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df24 = pd.DataFrame(pixArray24, columns=['Red','Green','Blue'])
df24.loc['mean'] = df24.mean()
#print(df4.tail())

arrMean24 = [[[]]]
arrMean24[0][0].append(df24.loc['mean'][0])
arrMean24[0][0].append(df24.loc['mean'][1])
arrMean24[0][0].append(df24.loc['mean'][2])

arrMeanNP24 = np.array(arrMean24)
col24 = np.uint8(arrMeanNP24)
print("Mean Segment RGB = ", col24)
colHSV24 = cv2.cvtColor(col24, cv2.COLOR_RGB2HSV)
HSVRanges24 = colHSV24.tolist()
print ("Mean Segment HSV = ", HSVRanges24)
H24 = HSVRanges24[0][0][0]
S24 = HSVRanges24[0][0][1]
V24 = HSVRanges24[0][0][2]
Classification24 = ''
if (H24 > -1 and H24 < 16 and S24 > 49 and S24 < 256 and V24 > -1 and V24 <␣
↪256):
        Classification24 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H24 > -1 and H24 < 101 and S24 > 0 and S24 < 31 and V24 > -1 and V24 <␣
↪180):
```

```
        Classification24 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification24 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(24)
start_point1 = (4297, 5392)
end_point1 = (4437, 5439)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification24, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges24[0][0])
Classi.append(Classification24)

# 25 ##############################################################################
print()
print("Sample Ground Truth 25")
pixArray25 = []
for i in range(4750+1, 4932):
        for j in range(6643+1,6747):
                pix = image[i,j]
                pixArray25.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df25 = pd.DataFrame(pixArray25, columns=['Red','Green','Blue'])
df25.loc['mean'] = df25.mean()
#print(df4.tail())

arrMean25 = [[[]]]
arrMean25[0][0].append(df25.loc['mean'][0])
arrMean25[0][0].append(df25.loc['mean'][1])
arrMean25[0][0].append(df25.loc['mean'][2])

arrMeanNP25 = np.array(arrMean25)
col25 = np.uint8(arrMeanNP25)
print("Mean Segment RGB = ", col25)
colHSV25 = cv2.cvtColor(col25, cv2.COLOR_RGB2HSV)
HSVRanges25 = colHSV25.tolist()
print ("Mean Segment HSV = ", HSVRanges25)
H25 = HSVRanges25[0][0][0]
S25 = HSVRanges25[0][0][1]
V25 = HSVRanges25[0][0][2]
Classification25 = ''
if (H25 > -1 and H25 < 16 and S25 > 49 and S25 < 256 and V25 > -1 and V25 <␣
 ↪256):
```

```
        Classification25 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H24 > -1 and H25 < 101 and S25 > 0 and S25 < 31 and V25 > -1 and V25 <␣
 ↪180):
        Classification25 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification25 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(25)
start_point1 = (6643, 4750)
end_point1 = (6747, 4932)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification25, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges25[0][0])
Classi.append(Classification25)

#plt.imshow(image)
#plt.title("Sample Ground Truth Segments")
#plt.show()

cv2.imwrite("Predicted_Roofs.tif", image)
print()
data = {"segmentNo":segmentNo, "X-coord": xcoord, "Y-coord": ycoord,␣
 ↪"AverageHSV":avgHSV, "Class":Classi}
groundtruths = pd.DataFrame(data)
print(groundtruths)
print()
print("These are the number of objects predicted to be under each class:")
print(groundtruths['Class'].value_counts())
```

```
Sample Ground Truth 1
Mean Segment RGB =  [[[122 120 105]]]
Mean Segment HSV =  [[[26, 36, 122]]]
Segment is non-classified

Sample Ground Truth 2
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[26, 36, 122]]]
Segment is non-classified

Sample Ground Truth 3
Mean Segment RGB =  [[[128 100  84]]]
Mean Segment HSV =  [[[11, 88, 128]]]
Segment is a red/orange/brown roof
```

```
Sample Ground Truth 4
Mean Segment RGB =  [[[150  65  52]]]
Mean Segment HSV =  [[[4, 167, 150]]]
Segment is a red/orange/brown roof

Sample Ground Truth 5
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 6
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 7
Mean Segment RGB =  [[[ 0 49 49]]]
Mean Segment HSV =  [[[90, 255, 49]]]
Segment is non-classified

Sample Ground Truth 8
Mean Segment RGB =  [[[129  89  81]]]
Mean Segment HSV =  [[[5, 95, 129]]]
Segment is a red/orange/brown roof

Sample Ground Truth 9
Mean Segment RGB =  [[[ 0 20 20]]]
Mean Segment HSV =  [[[90, 255, 20]]]
Segment is non-classified


Sample Ground Truth 10
Mean Segment RGB =  [[[ 55 107 100]]]
Mean Segment HSV =  [[[86, 124, 107]]]
Segment is non-classified

Sample Ground Truth 11
Mean Segment RGB =  [[[0 1 1]]]
Mean Segment HSV =  [[[90, 255, 1]]]
Segment is non-classified

Sample Ground Truth 12
Mean Segment RGB =  [[[  4 113 111]]]
Mean Segment HSV =  [[[89, 246, 113]]]
Segment is non-classified

Sample Ground Truth 13
```

```
Mean Segment RGB =  [[[164  81  69]]]
Mean Segment HSV =  [[[4, 148, 164]]]
Segment is a red/orange/brown roof

Sample Ground Truth 14
Mean Segment RGB =  [[[139 111  94]]]
Mean Segment HSV =  [[[11, 83, 139]]]
Segment is a red/orange/brown roof

Sample Ground Truth 15
Mean Segment RGB =  [[[212 157 144]]]
Mean Segment HSV =  [[[6, 82, 212]]]
Segment is a red/orange/brown roof

Sample Ground Truth 16
Mean Segment RGB =  [[[179  90  74]]]
Mean Segment HSV =  [[[5, 150, 179]]]
Segment is a red/orange/brown roof

Sample Ground Truth 17
Mean Segment RGB =  [[[191 152 138]]]
Mean Segment HSV =  [[[8, 71, 191]]]
Segment is a red/orange/brown roof

Sample Ground Truth 18
Mean Segment RGB =  [[[156 122 103]]]
Mean Segment HSV =  [[[11, 87, 156]]]
Segment is a red/orange/brown roof

Sample Ground Truth 19
Mean Segment RGB =  [[[205 158 137]]]
Mean Segment HSV =  [[[9, 85, 205]]]
Segment is a red/orange/brown roof

Sample Ground Truth 20
Mean Segment RGB =  [[[0 3 3]]]
Mean Segment HSV =  [[[90, 255, 3]]]
Segment is non-classified

Sample Ground Truth 21
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 22
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified
```

```
Sample Ground Truth 23
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 24
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 25
Mean Segment RGB =  [[[ 0 28 28]]]
Mean Segment HSV =  [[[90, 255, 28]]]
Segment is non-classified
```

	segmentNo	X-coord	Y-coord	AverageHSV	Class
0	1	2259	2255	[26, 36, 122]	Not classified
1	2	5000	2337	[0, 0, 0]	Not classified
2	3	2993	3001	[11, 88, 128]	Red/Orange/Brown roof
3	4	1203	4047	[4, 167, 150]	Red/Orange/Brown roof
4	5	4156	4907	[0, 0, 0]	Not classified
5	6	6740	7133	[0, 0, 0]	Not classified
6	7	2376	5731	[90, 255, 49]	Not classified
7	8	5932	5216	[5, 95, 129]	Red/Orange/Brown roof
8	9	697	4415	[90, 255, 20]	Not classified
9	10	4525	784	[86, 124, 107]	Not classified
10	11	3550	2094	[90, 255, 1]	Not classified
11	12	2966	6942	[89, 246, 113]	Not classified
12	13	2131	567	[4, 148, 164]	Red/Orange/Brown roof
13	14	1088	2354	[11, 83, 139]	Red/Orange/Brown roof
14	15	5352	7540	[6, 82, 212]	Red/Orange/Brown roof
15	16	873	6792	[5, 150, 179]	Red/Orange/Brown roof
16	17	7120	753	[8, 71, 191]	Red/Orange/Brown roof
17	18	3583	1029	[11, 87, 156]	Red/Orange/Brown roof
18	19	4030	2967	[9, 85, 205]	Red/Orange/Brown roof
19	20	3283	7897	[90, 255, 3]	Not classified
20	21	6756	4912	[0, 0, 0]	Not classified
21	22	4797	6444	[0, 0, 0]	Not classified
22	23	226	7226	[0, 0, 0]	Not classified
23	24	4297	5392	[0, 0, 0]	Not classified
24	25	6643	4750	[90, 255, 28]	Not classified

```
These are the number of objects predicted to be under each class:
Not classified           15
Red/Orange/Brown roof    10
Name: Class, dtype: int64
```

[ ]:

Road_Predictions

May 13, 2022

[2]:
```
import cv2
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np
from skimage import io

# Reading an image in default mode
image = cv2.imread("Tarred_Roads.tif")

# Window name in which image is displayed
window_name = 'Image'
color = (255, 0, 0)
thickness = 5
font = cv2.FONT_HERSHEY_DUPLEX

segmentNo = []
xcoord = []
ycoord = []
avgHSV = []
Classi = []

# 1 ##############################################################################
print("Sample Ground Truth 1")
segmentNo.append(1)
xcoord.append(2259)
ycoord.append(2255)
pixArray1 = []
for i in range(2255+1, 2337):
        for j in range(2259+1, 2414):
                pix = image[i,j]
                pixArray1.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df1 = pd.DataFrame(pixArray1, columns=['Red','Green','Blue'])
df1.loc['mean'] = df1.mean()
```

```
arrMean1 = [[[]]]
arrMean1[0][0].append(df1.loc['mean'][0])
arrMean1[0][0].append(df1.loc['mean'][1])
arrMean1[0][0].append(df1.loc['mean'][2])

arrMeanNP1 = np.array(arrMean1)
col1 = np.uint8(arrMeanNP1)
print("Mean Segment RGB = ", col1)
colHSV1 = cv2.cvtColor(col1, cv2.COLOR_RGB2HSV)
HSVRanges1 = colHSV1.tolist()
avgHSV.append(HSVRanges1[0][0])
print ("Mean Segment HSV = ", HSVRanges1)
H1 = HSVRanges1[0][0][0]
S1 = HSVRanges1[0][0][1]
V1 = HSVRanges1[0][0][2]
Classification1 = ''
if (H1 > -1 and H1 < 16 and S1 > 49 and S1 < 256 and V1 > -1 and V1 < 256):
        Classification1 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H1 > -1 and H1 < 101 and S1 > 0 and S1 < 31 and V1 > -1 and V1 < 180):
        Classification1 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification1 = 'Not classified'
        print("Segment is non-classified")
start_point1 = (2259, 2255)
end_point1 = (2414, 2337)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification1, start_point1, font, 2, color, 5)
Classi.append(Classification1)

# 2 ##############################################################################
print()
segmentNo.append(2)
xcoord.append(5000)
ycoord.append(2337)
print("Sample Ground Truth 2")
pixArray2 = []
for i in range(1900+1, 1927):
        for j in range(5000+1, 5230):
                pix = image[i,j]
                pixArray2.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df2 = pd.DataFrame(pixArray2, columns=['Red','Green','Blue'])
df2.loc['mean'] = df2.mean()
```

```
arrMean2 = [[[]]]
arrMean2[0][0].append(df2.loc['mean'][0])
arrMean2[0][0].append(df2.loc['mean'][1])
arrMean2[0][0].append(df2.loc['mean'][2])

arrMeanNP2 = np.array(arrMean2)
col2 = np.uint8(arrMeanNP2)
print("Mean Segment RGB = ", col2)
colHSV2 = cv2.cvtColor(col2, cv2.COLOR_RGB2HSV)
HSVRanges2 = colHSV2.tolist()
avgHSV.append(HSVRanges2[0][0])
print ("Mean Segment HSV = ", HSVRanges1)
H2 = HSVRanges2[0][0][0]
S2 = HSVRanges2[0][0][1]
V2 = HSVRanges2[0][0][2]
Classification2 = ''
if (H2 > -1 and H2 < 16 and S2 > 49 and S2 < 256 and V2 > -1 and V2 < 256):
        Classification2 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H2 > -1 and H2 < 101 and S2 > 0 and S2 < 31 and V2 > -1 and V2 < 180):
        Classification2 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification2 = 'Not classified'
        print("Segment is non-classified")
start_point1 = (5000, 1900)
end_point1 = (5230, 1927)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification2, start_point1, font, 2, color, 5)
Classi.append(Classification2)

# 3 ##############################################################################
print()
segmentNo.append(3)
xcoord.append(2993)
ycoord.append(3001)
print("Sample Ground Truth 3")
pixArray3 = []
for i in range(3001+1, 3047):
        for j in range(2993+1, 3085):
                pix = image[i,j]
                pixArray3.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df3 = pd.DataFrame(pixArray3, columns=['Red','Green','Blue'])
```

```
df3.loc['mean'] = df3.mean()

arrMean3 = [[[]]]
arrMean3[0][0].append(df3.loc['mean'][0])
arrMean3[0][0].append(df3.loc['mean'][1])
arrMean3[0][0].append(df3.loc['mean'][2])

arrMeanNP3 = np.array(arrMean3)
col3 = np.uint8(arrMeanNP3)
print("Mean Segment RGB = ", col3)
colHSV3 = cv2.cvtColor(col3, cv2.COLOR_RGB2HSV)
HSVRanges3 = colHSV3.tolist()
avgHSV.append(HSVRanges3[0][0])
print ("Mean Segment HSV = ", HSVRanges3)
H3 = HSVRanges3[0][0][0]
S3 = HSVRanges3[0][0][1]
V3 = HSVRanges3[0][0][2]
Classification3 = ''
if (H3 > -1 and H3 < 16 and S3 > 49 and S3 < 256 and V3 > -1 and V3 < 256):
        Classification3 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H3 > -1 and H3 < 101 and S3 > 0 and S3 < 31 and V3 > -1 and V3 < 180):
        Classification3 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification3 = 'Not classified'
        print("Segment is non-classified")
start_point1 = (2993, 3001)
end_point1 = (3085, 3047)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification3, start_point1, font, 2, color, 5)
Classi.append(Classification3)
# 4 ##############################################################################
print()
print("Sample Ground Truth 4")
pixArray4 = []
for i in range(4047+1, 4082):
        for j in range(1203+1, 1348):
                pix = image[i,j]
                pixArray4.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df4 = pd.DataFrame(pixArray4, columns=['Red','Green','Blue'])
df4.loc['mean'] = df4.mean()
#print(df4.tail())
```

```
arrMean4 = [[[]]]
arrMean4[0][0].append(df4.loc['mean'][0])
arrMean4[0][0].append(df4.loc['mean'][1])
arrMean4[0][0].append(df4.loc['mean'][2])

arrMeanNP4 = np.array(arrMean4)
col4 = np.uint8(arrMeanNP4)
print("Mean Segment RGB = ", col4)
colHSV4 = cv2.cvtColor(col4, cv2.COLOR_RGB2HSV)
HSVRanges4 = colHSV4.tolist()
print ("Mean Segment HSV = ", HSVRanges4)
H4 = HSVRanges4[0][0][0]
S4 = HSVRanges4[0][0][1]
V4 = HSVRanges4[0][0][2]
Classification4 = ''
if (H4 > -1 and H4 < 16 and S4 > 49 and S4 < 256 and V4 > -1 and V4 < 256):
        Classification4 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H4 > -1 and H4 < 101 and S4 > 0 and S4 < 31 and V4 > -1 and V4 < 180):
        Classification4 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification4 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(4)
start_point1 = (1203, 4047)
end_point1 = (1348, 4082)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification4, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges4[0][0])
Classi.append(Classification4)

# 5 ##############################################################################
print()

print("Sample Ground Truth 5")
pixArray5 = []
for i in range(4907+1, 4964):
        for j in range(4156+1, 4229):
                pix = image[i,j]
                pixArray5.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df5 = pd.DataFrame(pixArray5, columns=['Red','Green','Blue'])
```

```
df5.loc['mean'] = df5.mean()
#print(df4.tail())

#print(df4.loc['mean'])

arrMean5 = [[[]]]
arrMean5[0][0].append(df5.loc['mean'][0])
arrMean5[0][0].append(df5.loc['mean'][1])
arrMean5[0][0].append(df5.loc['mean'][2])

arrMeanNP5 = np.array(arrMean5)
col5 = np.uint8(arrMeanNP5)
print("Mean Segment RGB = ", col5)
colHSV5 = cv2.cvtColor(col5, cv2.COLOR_RGB2HSV)
HSVRanges5 = colHSV5.tolist()
print ("Mean Segment HSV = ", HSVRanges5)
H5 = HSVRanges5[0][0][0]
S5 = HSVRanges5[0][0][1]
V5 = HSVRanges5[0][0][2]
Classification5 = ''
if (H5 > -1 and H5 < 16 and S5 > 49 and S5 < 256 and V5 > -1 and V5 < 256):
        Classification5 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H5 > -1 and H5 < 101 and S5 > 0 and S5 < 31 and V5 > -1 and V5 < 180):
        Classification5 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification5 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(5)
start_point1 = (4156, 4907)
end_point1 = (4229, 4964)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification5, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges5[0][0])
Classi.append(Classification5)


# 6 ##############################################################################
print()
print("Sample Ground Truth 6")
pixArray6 = []
for i in range(7133+1, 7199):
        for j in range(6740+1, 6946):
                pix = image[i,j]
```

```
                pixArray6.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df6 = pd.DataFrame(pixArray6, columns=['Red','Green','Blue'])
df6.loc['mean'] = df6.mean()
#print(df4.tail())

arrMean6 = [[[]]]
arrMean6[0][0].append(df6.loc['mean'][0])
arrMean6[0][0].append(df6.loc['mean'][1])
arrMean6[0][0].append(df6.loc['mean'][2])

arrMeanNP6 = np.array(arrMean6)
col6 = np.uint8(arrMeanNP6)
print("Mean Segment RGB = ", col6)
colHSV6 = cv2.cvtColor(col6, cv2.COLOR_RGB2HSV)
HSVRanges6 = colHSV6.tolist()
print ("Mean Segment HSV = ", HSVRanges6)
H6 = HSVRanges6[0][0][0]
S6 = HSVRanges6[0][0][1]
V6 = HSVRanges6[0][0][2]
Classification6 = ''
if (H6 > -1 and H6 < 16 and S6 > 49 and S6 < 256 and V6 > -1 and V6 < 256):
        Classification6 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H6 > -1 and H6 < 101 and S6 > 0 and S6 < 31 and V6 > -1 and V6 < 180):
        Classification6 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification6 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(6)
start_point1 = (6740, 7133)
end_point1 = (6946, 7199)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification6, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges6[0][0])
Classi.append(Classification6)

# 7 ##############################################################################
print()
print("Sample Ground Truth 7")
pixArray7 = []
for i in range(5731+1, 5881):
```

```
        for j in range(2376+1, 2648):
                pix = image[i,j]
                pixArray7.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df7 = pd.DataFrame(pixArray7, columns=['Red','Green','Blue'])
df7.loc['mean'] = df7.mean()
#print(df4.tail())

arrMean7 = [[[]]]
arrMean7[0][0].append(df7.loc['mean'][0])
arrMean7[0][0].append(df7.loc['mean'][1])
arrMean7[0][0].append(df7.loc['mean'][2])

arrMeanNP7 = np.array(arrMean7)
col7 = np.uint8(arrMeanNP7)
print("Mean Segment RGB = ", col7)
colHSV7 = cv2.cvtColor(col7, cv2.COLOR_RGB2HSV)
HSVRanges7 = colHSV7.tolist()
print ("Mean Segment HSV = ", HSVRanges7)
H7 = HSVRanges7[0][0][0]
S7 = HSVRanges7[0][0][1]
V7 = HSVRanges7[0][0][2]
Classification7 = ''
if (H7 > -1 and H7 < 16 and S7 > 49 and S7 < 256 and V7 > -1 and V7 < 256):
        Classification7 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H7 > -1 and H7 < 101 and S7 > 0 and S7 < 31 and V7 > -1 and V7 < 180):
        Classification7 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification7 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(7)
start_point1 = (2376, 5731)
end_point1 = (2648, 5881)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification7, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges7[0][0])
Classi.append(Classification7)

# 8 ##############################################################################
print()
print("Sample Ground Truth 8")
```

```
pixArray8 = []
for i in range(5216+1, 5349):
        for j in range(5932+1, 5984):
                pix = image[i,j]
                pixArray8.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df8 = pd.DataFrame(pixArray8, columns=['Red','Green','Blue'])
df8.loc['mean'] = df8.mean()
#print(df4.tail())

arrMean8 = [[[]]]
arrMean8[0][0].append(df8.loc['mean'][0])
arrMean8[0][0].append(df8.loc['mean'][1])
arrMean8[0][0].append(df8.loc['mean'][2])

arrMeanNP8 = np.array(arrMean8)
col8 = np.uint8(arrMeanNP8)
print("Mean Segment RGB = ", col8)
colHSV8 = cv2.cvtColor(col8, cv2.COLOR_RGB2HSV)
HSVRanges8 = colHSV8.tolist()
print ("Mean Segment HSV = ", HSVRanges8)
H8 = HSVRanges8[0][0][0]
S8 = HSVRanges8[0][0][1]
V8 = HSVRanges8[0][0][2]
Classification8 = ''
if (H8 > -1 and H8 < 16 and S8 > 49 and S8 < 256 and V8 > -1 and V8 < 256):
        Classification8 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H8 > -1 and H8 < 101 and S8 > 0 and S8 < 31 and V8 > -1 and V8 < 180):
        Classification8 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification8 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(8)
start_point1 = (5932, 5216)
end_point1 = (5984, 5349)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification8, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges8[0][0])
Classi.append(Classification8)

# 9 ##############################################################################
```

```
print()
print("Sample Ground Truth 9")
pixArray9 = []
for i in range(4415+1, 4485):
        for j in range(697+1, 782):
                pix = image[i,j]
                pixArray9.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df9 = pd.DataFrame(pixArray9, columns=['Red','Green','Blue'])
df9.loc['mean'] = df9.mean()
#print(df4.tail())

arrMean9 = [[[]]]
arrMean9[0][0].append(df9.loc['mean'][0])
arrMean9[0][0].append(df9.loc['mean'][1])
arrMean9[0][0].append(df9.loc['mean'][2])

arrMeanNP9 = np.array(arrMean9)
col9 = np.uint8(arrMeanNP9)
print("Mean Segment RGB = ", col9)
colHSV9 = cv2.cvtColor(col9, cv2.COLOR_RGB2HSV)
HSVRanges9 = colHSV9.tolist()
print ("Mean Segment HSV = ", HSVRanges9)
H9 = HSVRanges9[0][0][0]
S9 = HSVRanges9[0][0][1]
V9 = HSVRanges9[0][0][2]
Classification9 = ''
if (H9 > -1 and H9 < 16 and S9 > 49 and S9 < 256 and V9 > -1 and V9 < 256):
        Classification9 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H9 > -1 and H9 < 101 and S9 > 0 and S9 < 31 and V9 > -1 and V9 < 180):
        Classification9 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification9 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(9)
start_point1 = (697, 4415)
end_point1 = (782, 4485)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification9, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges9[0][0])
Classi.append(Classification9)
```

```
# 10␣
↪#############################################################################
print()

print()
print("Sample Ground Truth 10")
pixArray10 = []
for i in range(784+1, 910):
        for j in range(4525+1, 4658):
                pix = image[i,j]
                pixArray10.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df10 = pd.DataFrame(pixArray10, columns=['Red','Green','Blue'])
df10.loc['mean'] = df10.mean()
#print(df4.tail())

arrMean10 = [[[]]]
arrMean10[0][0].append(df10.loc['mean'][0])
arrMean10[0][0].append(df10.loc['mean'][1])
arrMean10[0][0].append(df10.loc['mean'][2])

arrMeanNP10 = np.array(arrMean10)
col10 = np.uint8(arrMeanNP10)
print("Mean Segment RGB = ", col10)
colHSV10 = cv2.cvtColor(col10, cv2.COLOR_RGB2HSV)
HSVRanges10 = colHSV10.tolist()
print ("Mean Segment HSV = ", HSVRanges10)
H10 = HSVRanges10[0][0][0]
S10 = HSVRanges10[0][0][1]
V10 = HSVRanges10[0][0][2]
Classification10 = ''
if (H10 > -1 and H10 < 16 and S10 > 49 and S10 < 256 and V10 > -1 and V10 <␣
↪256):
        Classification10 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H10 > -1 and H10 < 101 and S10 > 0 and S10 < 31 and V10 > -1 and V10 <␣
↪180):
        Classification10 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification10 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(10)
start_point1 = (4525, 784)
```

```
end_point1 = (4658, 910)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification10, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges10[0][0])
Classi.append(Classification10)

# 11 ##############################################################################
print()
print("Sample Ground Truth 11")
pixArray11 = []
for i in range(2094+1, 2188):
        for j in range(3550+1, 3649):
                pix = image[i,j]
                pixArray11.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df11 = pd.DataFrame(pixArray11, columns=['Red','Green','Blue'])
df11.loc['mean'] = df11.mean()
#print(df4.tail())

arrMean11 = [[[]]]
arrMean11[0][0].append(df11.loc['mean'][0])
arrMean11[0][0].append(df11.loc['mean'][1])
arrMean11[0][0].append(df11.loc['mean'][2])

arrMeanNP11 = np.array(arrMean11)
col11 = np.uint8(arrMeanNP11)
print("Mean Segment RGB = ", col11)
colHSV11 = cv2.cvtColor(col11, cv2.COLOR_RGB2HSV)
HSVRanges11 = colHSV11.tolist()
print ("Mean Segment HSV = ", HSVRanges11)
H11 = HSVRanges11[0][0][0]
S11 = HSVRanges11[0][0][1]
V11 = HSVRanges11[0][0][2]
Classification11 = ''
if (H11 > -1 and H11 < 16 and S11 > 49 and S11 < 256 and V11 > -1 and V11 <␣
↪256):
        Classification10 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H11 > -1 and H11 < 101 and S11 > 0 and S11 < 31 and V11 > -1 and V11 <␣
↪180):
        Classification11 = 'Tarred Road'
        print("Segment is a tarred road")
else:
```

```
        Classification11 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(11)
start_point1 = (3550, 2094)
end_point1 = (3649, 2188)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification11, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges11[0][0])
Classi.append(Classification11)

# 12 ##############################################################################
print()
print("Sample Ground Truth 12")
pixArray12 = []
for i in range(6942+1, 7099):
        for j in range(2966+1, 3520):
                pix = image[i,j]
                pixArray12.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df12 = pd.DataFrame(pixArray12, columns=['Red','Green','Blue'])
df12.loc['mean'] = df12.mean()
#print(df4.tail())

arrMean12 = [[[]]]
arrMean12[0][0].append(df12.loc['mean'][0])
arrMean12[0][0].append(df12.loc['mean'][1])
arrMean12[0][0].append(df12.loc['mean'][2])

arrMeanNP12 = np.array(arrMean12)
col12 = np.uint8(arrMeanNP12)
print("Mean Segment RGB = ", col12)
colHSV12 = cv2.cvtColor(col12, cv2.COLOR_RGB2HSV)
HSVRanges12 = colHSV12.tolist()
print ("Mean Segment HSV = ", HSVRanges12)
H12 = HSVRanges12[0][0][0]
S12 = HSVRanges12[0][0][1]
V12 = HSVRanges12[0][0][2]
Classification12 = ''
if (H12 > -1 and H12 < 15 and S12 > 49 and S12 < 256 and V12 > -1 and V12 <␣
 ↪256):
        Classification12 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
```

```
elif (H12 > -1 and H12 < 101 and S12 > 0 and S12 < 31 and V12 > -1 and V12 <␣
 ↪180):
        Classification12 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification12 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(12)
start_point1 = (2966, 6942)
end_point1 = (3520, 7099)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification12, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges12[0][0])
Classi.append(Classification12)

# 13 ##############################################################################
print()
print("Sample Ground Truth 13")
pixArray13 = []
for i in range(567+1, 615):
        for j in range(2131+1, 2215):
                pix = image[i,j]
                pixArray13.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df13 = pd.DataFrame(pixArray13, columns=['Red','Green','Blue'])
df13.loc['mean'] = df13.mean()
#print(df4.tail())

arrMean13 = [[[]]]
arrMean13[0][0].append(df13.loc['mean'][0])
arrMean13[0][0].append(df13.loc['mean'][1])
arrMean13[0][0].append(df13.loc['mean'][2])

arrMeanNP13 = np.array(arrMean13)
col13 = np.uint8(arrMeanNP13)
print("Mean Segment RGB = ", col13)
colHSV13 = cv2.cvtColor(col13, cv2.COLOR_RGB2HSV)
HSVRanges13 = colHSV13.tolist()
print ("Mean Segment HSV = ", HSVRanges13)
H13 = HSVRanges13[0][0][0]
S13 = HSVRanges13[0][0][1]
V13 = HSVRanges13[0][0][2]
Classification13 = ''
```

```
if (H13 > -1 and H13 < 16 and S13 > 49 and S13 < 256 and V13 > -1 and V13 <␣
 ↪256):
        Classification13 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H13 > -1 and H13 < 101 and S13 > 0 and S13 < 31 and V13 > -1 and V13 <␣
 ↪180):
        Classification13 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification13 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(13)
start_point1 = (2131, 567)
end_point1 = (2215, 615)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification13, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges13[0][0])
Classi.append(Classification13)
# 14 ##############################################################################
print()
print("Sample Ground Truth 14")
pixArray14 = []
for i in range(2354+1, 2417):
        for j in range(1088+1, 1220):
                pix = image[i,j]
                pixArray14.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df14 = pd.DataFrame(pixArray14, columns=['Red','Green','Blue'])
df14.loc['mean'] = df14.mean()
#print(df4.tail())

arrMean14 = [[[]]]
arrMean14[0][0].append(df14.loc['mean'][0])
arrMean14[0][0].append(df14.loc['mean'][1])
arrMean14[0][0].append(df14.loc['mean'][2])

arrMeanNP14 = np.array(arrMean14)
col14 = np.uint8(arrMeanNP14)
print("Mean Segment RGB = ", col14)
colHSV14 = cv2.cvtColor(col14, cv2.COLOR_RGB2HSV)
HSVRanges14 = colHSV14.tolist()
print ("Mean Segment HSV = ", HSVRanges14)
H14 = HSVRanges14[0][0][0]
```

```
S14 = HSVRanges14[0][0][1]
V14 = HSVRanges14[0][0][2]
Classification14 = ''
if (H14 > -1 and H14 < 16 and S14 > 49 and S14 < 256 and V14 > -1 and V14 <␣
 ↪256):
        Classification14 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H14 > -1 and H14 < 101 and S14 > 0 and S14 < 31 and V14 > -1 and V14 <␣
 ↪180):
        Classification14 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification14 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(14)
start_point1 = (1088, 2354)
end_point1 = (1220, 2417)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification14, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges14[0][0])
Classi.append(Classification14)
# 15 ##########################################################################
print()
print("Sample Ground Truth 15")
pixArray15 = []
for i in range(7540+1, 7614):
        for j in range(5352+1, 5425):
                pix = image[i,j]
                pixArray15.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df15 = pd.DataFrame(pixArray15, columns=['Red','Green','Blue'])
df15.loc['mean'] = df15.mean()
#print(df4.tail())

arrMean15 = [[[]]]
arrMean15[0][0].append(df15.loc['mean'][0])
arrMean15[0][0].append(df15.loc['mean'][1])
arrMean15[0][0].append(df15.loc['mean'][2])

arrMeanNP15 = np.array(arrMean15)
col15 = np.uint8(arrMeanNP15)
print("Mean Segment RGB = ", col15)
colHSV15 = cv2.cvtColor(col15, cv2.COLOR_RGB2HSV)
```

```
HSVRanges15 = colHSV15.tolist()
print ("Mean Segment HSV = ", HSVRanges15)
H15 = HSVRanges15[0][0][0]
S15 = HSVRanges15[0][0][1]
V15 = HSVRanges15[0][0][2]
Classification15 = ''
if (H15 > -1 and H15 < 16 and S15 > 49 and S15 < 256 and V15 > -1 and V15 <␣
 ↪256):
        Classification15 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H15 > -1 and H15 < 101 and S15 > 0 and S15 < 31 and V15 > -1 and V15 <␣
 ↪180):
        Classification15 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification15 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(15)
start_point1 = (5352, 7540)
end_point1 = (5425, 7614)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification15, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges15[0][0])
Classi.append(Classification15)

# 16 ##############################################################################
print()
print("Sample Ground Truth 16")
pixArray16 = []
for i in range(6792+1, 6823):
        for j in range(873+1, 980):
                pix = image[i,j]
                pixArray16.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df16 = pd.DataFrame(pixArray16, columns=['Red','Green','Blue'])
df16.loc['mean'] = df16.mean()
#print(df4.tail())

arrMean16 = [[[]]]
arrMean16[0][0].append(df16.loc['mean'][0])
arrMean16[0][0].append(df16.loc['mean'][1])
arrMean16[0][0].append(df16.loc['mean'][2])
```

```
arrMeanNP16 = np.array(arrMean16)
col16 = np.uint8(arrMeanNP16)
print("Mean Segment RGB = ", col16)
colHSV16 = cv2.cvtColor(col16, cv2.COLOR_RGB2HSV)
HSVRanges16 = colHSV16.tolist()
print ("Mean Segment HSV = ", HSVRanges16)
H16 = HSVRanges16[0][0][0]
S16 = HSVRanges16[0][0][1]
V16 = HSVRanges16[0][0][2]
Classification16 = ''
if (H16 > -1 and H16 < 16 and S16 > 49 and S16 < 256 and V16 > -1 and V16 <␣
 ↪256):
        Classification16 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H16 > -1 and H16 < 101 and S16 > 0 and S16 < 31 and V16 > -1 and V16 <␣
 ↪180):
        Classification16 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification16 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(16)
start_point1 = (873, 6792)
end_point1 = (980, 6823)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification16, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges16[0][0])
Classi.append(Classification16)

# 17 ##############################################################################
print()
print("Sample Ground Truth 17")
pixArray17 = []
for i in range(753+1, 821):
        for j in range(7120+1, 7216):
                pix = image[i,j]
                pixArray17.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df17 = pd.DataFrame(pixArray17, columns=['Red','Green','Blue'])
df17.loc['mean'] = df17.mean()
#print(df4.tail())

arrMean17 = [[[]]]
```

```
arrMean17[0][0].append(df17.loc['mean'][0])
arrMean17[0][0].append(df17.loc['mean'][1])
arrMean17[0][0].append(df17.loc['mean'][2])

arrMeanNP17 = np.array(arrMean17)
col17 = np.uint8(arrMeanNP17)
print("Mean Segment RGB = ", col17)
colHSV17 = cv2.cvtColor(col17, cv2.COLOR_RGB2HSV)
HSVRanges17 = colHSV17.tolist()
print ("Mean Segment HSV = ", HSVRanges17)
H17 = HSVRanges17[0][0][0]
S17 = HSVRanges17[0][0][1]
V17 = HSVRanges17[0][0][2]
Classification17 = ''
if (H17 > -1 and H17 < 16 and S17 > 49 and S17 < 256 and V17 > -1 and V17 <␣
 ↪256):
        Classification17 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H17 > -1 and H17 < 101 and S17 > 0 and S17 < 31 and V17 > -1 and V17 <␣
 ↪180):
        Classification17 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification17 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(17)
start_point1 = (7120, 753)
end_point1 = (7216, 821)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification17, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges17[0][0])
Classi.append(Classification17)

# 18 ##############################################################################
print()
print("Sample Ground Truth 18")
pixArray18 = []
for i in range(1029+1, 1093):
        for j in range(3583+1, 3700):
                pix = image[i,j]
                pixArray18.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df18 = pd.DataFrame(pixArray18, columns=['Red','Green','Blue'])
```

```
df18.loc['mean'] = df18.mean()
#print(df4.tail())

arrMean18 = [[[]]]
arrMean18[0][0].append(df18.loc['mean'][0])
arrMean18[0][0].append(df18.loc['mean'][1])
arrMean18[0][0].append(df18.loc['mean'][2])

arrMeanNP18 = np.array(arrMean18)
col18 = np.uint8(arrMeanNP18)
print("Mean Segment RGB = ", col18)
colHSV18 = cv2.cvtColor(col18, cv2.COLOR_RGB2HSV)
HSVRanges18 = colHSV18.tolist()
print ("Mean Segment HSV = ", HSVRanges18)
H18 = HSVRanges18[0][0][0]
S18 = HSVRanges18[0][0][1]
V18 = HSVRanges18[0][0][2]
Classification18 = ''
if (H18 > -1 and H18 < 16 and S18 > 49 and S18 < 256 and V18 > -1 and V18 <␣
 ↪256):
        Classification18 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H18 > -1 and H18 < 101 and S18 > 0 and S18 < 31 and V18 > -1 and V18 <␣
 ↪180):
        Classification18 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification18 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(18)
start_point1 = (3583, 1029)
end_point1 = (3700, 1093)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification18, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges18[0][0])
Classi.append(Classification18)

# 19 ##############################################################################
print()
print("Sample Ground Truth 19")
pixArray19 = []
for i in range(2967+1, 3028):
        for j in range(4030+1, 4132):
                pix = image[i,j]
                pixArray19.append(pix.tolist())
```

```
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df19 = pd.DataFrame(pixArray19, columns=['Red','Green','Blue'])
df19.loc['mean'] = df19.mean()
#print(df4.tail())

arrMean19 = [[[]]]
arrMean19[0][0].append(df19.loc['mean'][0])
arrMean19[0][0].append(df19.loc['mean'][1])
arrMean19[0][0].append(df19.loc['mean'][2])

arrMeanNP19 = np.array(arrMean19)
col19 = np.uint8(arrMeanNP19)
print("Mean Segment RGB = ", col19)
colHSV19 = cv2.cvtColor(col19, cv2.COLOR_RGB2HSV)
HSVRanges19 = colHSV19.tolist()
print ("Mean Segment HSV = ", HSVRanges19)
H19 = HSVRanges19[0][0][0]
S19 = HSVRanges19[0][0][1]
V19 = HSVRanges19[0][0][2]
Classification19 = ''
if (H19 > -1 and H19 < 16 and S19 > 49 and S19 < 256 and V19 > -1 and V19 <␣
 ↪256):
        Classification19 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H19 > -1 and H19 < 101 and S19 > 0 and S19 < 31 and V19 > -1 and V19 <␣
 ↪180):
        Classification19 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification19 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(19)
start_point1 = (4030, 2967)
end_point1 = (4132, 3028)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification19, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges19[0][0])
Classi.append(Classification19)
# 20 ##########################################################################
print()

print("Sample Ground Truth 20")
pixArray20 = []
```

```
for i in range(7897+1, 8104):
        for j in range(3283+1, 3712):
                pix = image[i,j]
                pixArray20.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df20 = pd.DataFrame(pixArray20, columns=['Red','Green','Blue'])
df20.loc['mean'] = df20.mean()
#print(df4.tail())

arrMean20 = [[[]]]
arrMean20[0][0].append(df20.loc['mean'][0])
arrMean20[0][0].append(df20.loc['mean'][1])
arrMean20[0][0].append(df20.loc['mean'][2])

arrMeanNP20 = np.array(arrMean20)
col20 = np.uint8(arrMeanNP20)
print("Mean Segment RGB = ", col20)
colHSV20 = cv2.cvtColor(col20, cv2.COLOR_RGB2HSV)
HSVRanges20 = colHSV20.tolist()
print ("Mean Segment HSV = ", HSVRanges20)
H20 = HSVRanges20[0][0][0]
S20 = HSVRanges20[0][0][1]
V20 = HSVRanges20[0][0][2]
Classification20 = ''
if (H20 > -1 and H20 < 16 and S20 > 49 and S20 < 256 and V20 > -1 and V20 <␣
 ↪256):
        Classification20 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H20 > -1 and H20 < 101 and S20 > 0 and S20 < 31 and V20 > -1 and V20 <␣
 ↪180):
        Classification20 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification20 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(20)
start_point1 = (3283, 7897)
end_point1 = (3712, 8104)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification20, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges20[0][0])
Classi.append(Classification20)
```

```
# 21 ##############################################################################
print()
print("Sample Ground Truth 21")
pixArray21 = []
for i in range(4912+1, 5060):
        for j in range(6756+1, 6828):
                pix = image[i,j]
                pixArray21.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df21 = pd.DataFrame(pixArray21, columns=['Red','Green','Blue'])
df21.loc['mean'] = df21.mean()
#print(df4.tail())

arrMean21 = [[[]]]
arrMean21[0][0].append(df21.loc['mean'][0])
arrMean21[0][0].append(df21.loc['mean'][1])
arrMean21[0][0].append(df21.loc['mean'][2])

arrMeanNP21 = np.array(arrMean21)
col21 = np.uint8(arrMeanNP21)
print("Mean Segment RGB = ", col21)
colHSV21 = cv2.cvtColor(col21, cv2.COLOR_RGB2HSV)
HSVRanges21 = colHSV21.tolist()
print ("Mean Segment HSV = ", HSVRanges21)
H21 = HSVRanges21[0][0][0]
S21 = HSVRanges21[0][0][1]
V21 = HSVRanges21[0][0][2]
Classification21 = ''
if (H21 > -1 and H21 < 16 and S21 > 49 and S21 < 256 and V21 > -1 and V21 <␣
 ↪256):
        Classification21 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H21 > -1 and H21 < 101 and S21 > 0 and S21 < 31 and V21 > -1 and V21 <␣
 ↪180):
        Classification21 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification21 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(21)
start_point1 = (6756, 4912)
end_point1 = (6828, 5060)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification21, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
```

```
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges21[0][0])
Classi.append(Classification21)

# 22 ##############################################################################
print()
print("Sample Ground Truth 22")
pixArray22 = []
for i in range(6444+1, 6538):
        for j in range(4797+1,4833):
                pix = image[i,j]
                pixArray22.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df22 = pd.DataFrame(pixArray22, columns=['Red','Green','Blue'])
df22.loc['mean'] = df22.mean()
#print(df4.tail())

arrMean22 = [[[]]]
arrMean22[0][0].append(df22.loc['mean'][0])
arrMean22[0][0].append(df22.loc['mean'][1])
arrMean22[0][0].append(df22.loc['mean'][2])

arrMeanNP22 = np.array(arrMean22)
col22 = np.uint8(arrMeanNP22)
print("Mean Segment RGB = ", col22)
colHSV22 = cv2.cvtColor(col22, cv2.COLOR_RGB2HSV)
HSVRanges22 = colHSV22.tolist()
print ("Mean Segment HSV = ", HSVRanges22)
H22 = HSVRanges22[0][0][0]
S22 = HSVRanges22[0][0][1]
V22 = HSVRanges22[0][0][2]
Classification22 = ''
if (H22 > -1 and H22 < 16 and S22 > 49 and S22 < 256 and V22 > -1 and V22 <␣
 ↪256):
        Classification22 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H22 > -1 and H22 < 101 and S22 > 0 and S22 < 31 and V22 > -1 and V22 <␣
 ↪180):
        Classification22 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification22 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(22)
start_point1 = (4797, 6444)
```

```
end_point1 = (4833, 6538)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification22, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges22[0][0])
Classi.append(Classification22)

# 23 ##############################################################################
print()
print("Sample Ground Truth 23")
pixArray23 = []
for i in range(7226+1, 7460):
        for j in range(226+1,336):
                pix = image[i,j]
                pixArray23.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df23 = pd.DataFrame(pixArray23, columns=['Red','Green','Blue'])
df23.loc['mean'] = df23.mean()
#print(df4.tail())

arrMean23 = [[[]]]
arrMean23[0][0].append(df23.loc['mean'][0])
arrMean23[0][0].append(df23.loc['mean'][1])
arrMean23[0][0].append(df23.loc['mean'][2])

arrMeanNP23 = np.array(arrMean23)
col23 = np.uint8(arrMeanNP23)
print("Mean Segment RGB = ", col23)
colHSV23 = cv2.cvtColor(col23, cv2.COLOR_RGB2HSV)
HSVRanges23 = colHSV23.tolist()
print ("Mean Segment HSV = ", HSVRanges23)
H23 = HSVRanges23[0][0][0]
S23 = HSVRanges23[0][0][1]
V23 = HSVRanges23[0][0][2]
Classification23 = ''
if (H23 > -1 and H23 < 16 and S23 > 49 and S23 < 256 and V23 > -1 and V23 <␣
 ↪256):
        Classification23 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H23 > -1 and H23 < 101 and S23 > 0 and S23 < 31 and V23 > -1 and V23 <␣
 ↪180):
        Classification23 = 'Tarred Road'
        print("Segment is a tarred road")
else:
```

```
        Classification23 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(23)
start_point1 = (226, 7226)
end_point1 = (336, 7460)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification23, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges23[0][0])
Classi.append(Classification23)
# 24 ##############################################################################
print()
print("Sample Ground Truth 24")
pixArray24 = []
for i in range(5392+1, 5439):
        for j in range(4297+1,4437):
                pix = image[i,j]
                pixArray24.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df24 = pd.DataFrame(pixArray24, columns=['Red','Green','Blue'])
df24.loc['mean'] = df24.mean()
#print(df4.tail())

arrMean24 = [[[]]]
arrMean24[0][0].append(df24.loc['mean'][0])
arrMean24[0][0].append(df24.loc['mean'][1])
arrMean24[0][0].append(df24.loc['mean'][2])

arrMeanNP24 = np.array(arrMean24)
col24 = np.uint8(arrMeanNP24)
print("Mean Segment RGB = ", col24)
colHSV24 = cv2.cvtColor(col24, cv2.COLOR_RGB2HSV)
HSVRanges24 = colHSV24.tolist()
print ("Mean Segment HSV = ", HSVRanges24)
H24 = HSVRanges24[0][0][0]
S24 = HSVRanges24[0][0][1]
V24 = HSVRanges24[0][0][2]
Classification24 = ''
if (H24 > -1 and H24 < 16 and S24 > 49 and S24 < 256 and V24 > -1 and V24 <␣
↪256):
        Classification24 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H24 > -1 and H24 < 101 and S24 > 0 and S24 < 31 and V24 > -1 and V24 <␣
↪180):
```

```
        Classification24 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification24 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(24)
start_point1 = (4297, 5392)
end_point1 = (4437, 5439)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification24, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges24[0][0])
Classi.append(Classification24)

# 25 ##############################################################################
print()
print("Sample Ground Truth 25")
pixArray25 = []
for i in range(4750+1, 4932):
        for j in range(6643+1,6747):
                pix = image[i,j]
                pixArray25.append(pix.tolist())
                #print(image[i,j])
                image[i][j] = [1, 100, 1]

df25 = pd.DataFrame(pixArray25, columns=['Red','Green','Blue'])
df25.loc['mean'] = df25.mean()
#print(df4.tail())

arrMean25 = [[[]]]
arrMean25[0][0].append(df25.loc['mean'][0])
arrMean25[0][0].append(df25.loc['mean'][1])
arrMean25[0][0].append(df25.loc['mean'][2])

arrMeanNP25 = np.array(arrMean25)
col25 = np.uint8(arrMeanNP25)
print("Mean Segment RGB = ", col25)
colHSV25 = cv2.cvtColor(col25, cv2.COLOR_RGB2HSV)
HSVRanges25 = colHSV25.tolist()
print ("Mean Segment HSV = ", HSVRanges25)
H25 = HSVRanges25[0][0][0]
S25 = HSVRanges25[0][0][1]
V25 = HSVRanges25[0][0][2]
Classification25 = ''
if (H25 > -1 and H25 < 16 and S25 > 49 and S25 < 256 and V25 > -1 and V25 <␣
 ↪256):
```

```
        Classification25 = 'Red/Orange/Brown roof'
        print("Segment is a red/orange/brown roof")
elif (H24 > -1 and H25 < 101 and S25 > 0 and S25 < 31 and V25 > -1 and V25 <␣
 ↪180):
        Classification25 = 'Tarred Road'
        print("Segment is a tarred road")
else:
        Classification25 = 'Not classified'
        print("Segment is non-classified")
segmentNo.append(25)
start_point1 = (6643, 4750)
end_point1 = (6747, 4932)
cv2.rectangle(image, start_point1, end_point1, color, thickness)
cv2.putText(image, Classification25, start_point1, font, 2, color, 5)
xcoord.append(start_point1[0])
ycoord.append(start_point1[1])
avgHSV.append(HSVRanges25[0][0])
Classi.append(Classification25)

#plt.imshow(image)
#plt.title("Sample Ground Truth Segments")
#plt.show()

cv2.imwrite("Predicted_Roads.tif", image)
print()
data = {"segmentNo":segmentNo, "X-coord": xcoord, "Y-coord": ycoord,␣
 ↪"AverageHSV":avgHSV, "Class":Classi}
groundtruths = pd.DataFrame(data)
print(groundtruths)
print()
print("These are the number of objects predicted to be under each class:")
print(groundtruths['Class'].value_counts())
```

```
Sample Ground Truth 1
Mean Segment RGB =  [[[ 0 17  0]]]
Mean Segment HSV =  [[[60, 255, 17]]]
Segment is non-classified

Sample Ground Truth 2
Mean Segment RGB =  [[[131 144 121]]]
Mean Segment HSV =  [[[60, 255, 17]]]
Segment is non-classified

Sample Ground Truth 3
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified
```

```
Sample Ground Truth 4
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 5
Mean Segment RGB =  [[[19 18 17]]]
Mean Segment HSV =  [[[15, 27, 19]]]
Segment is a tarred road

Sample Ground Truth 6
Mean Segment RGB =  [[[63 65 62]]]
Mean Segment HSV =  [[[50, 12, 65]]]
Segment is a tarred road

Sample Ground Truth 7
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 8
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 9
Mean Segment RGB =  [[[ 0 33  0]]]
Mean Segment HSV =  [[[60, 255, 33]]]
Segment is non-classified


Sample Ground Truth 10
Mean Segment RGB =  [[[ 0 44  0]]]
Mean Segment HSV =  [[[60, 255, 44]]]
Segment is non-classified

Sample Ground Truth 11
Mean Segment RGB =  [[[105 131  96]]]
Mean Segment HSV =  [[[52, 68, 131]]]
Segment is non-classified

Sample Ground Truth 12
Mean Segment RGB =  [[[  0 138   0]]]
Mean Segment HSV =  [[[60, 255, 138]]]
Segment is non-classified

Sample Ground Truth 13
```

```
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 14
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 15
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 16
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 17
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 18
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 19
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 20
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 21
Mean Segment RGB =  [[[133 128 120]]]
Mean Segment HSV =  [[[18, 25, 133]]]
Segment is a tarred road

Sample Ground Truth 22
Mean Segment RGB =  [[[151 147 139]]]
Mean Segment HSV =  [[[20, 20, 151]]]
Segment is a tarred road
```

```
Sample Ground Truth 23
Mean Segment RGB =  [[[103 114  95]]]
Mean Segment HSV =  [[[47, 42, 114]]]
Segment is non-classified

Sample Ground Truth 24
Mean Segment RGB =  [[[0 0 0]]]
Mean Segment HSV =  [[[0, 0, 0]]]
Segment is non-classified

Sample Ground Truth 25
Mean Segment RGB =  [[[28 28 26]]]
Mean Segment HSV =  [[[30, 18, 28]]]
Segment is a tarred road
```

	segmentNo	X-coord	Y-coord	AverageHSV	Class
0	1	2259	2255	[60, 255, 17]	Not classified
1	2	5000	2337	[47, 41, 144]	Not classified
2	3	2993	3001	[0, 0, 0]	Not classified
3	4	1203	4047	[0, 0, 0]	Not classified
4	5	4156	4907	[15, 27, 19]	Tarred Road
5	6	6740	7133	[50, 12, 65]	Tarred Road
6	7	2376	5731	[0, 0, 0]	Not classified
7	8	5932	5216	[0, 0, 0]	Not classified
8	9	697	4415	[60, 255, 33]	Not classified
9	10	4525	784	[60, 255, 44]	Not classified
10	11	3550	2094	[52, 68, 131]	Not classified
11	12	2966	6942	[60, 255, 138]	Not classified
12	13	2131	567	[0, 0, 0]	Not classified
13	14	1088	2354	[0, 0, 0]	Not classified
14	15	5352	7540	[0, 0, 0]	Not classified
15	16	873	6792	[0, 0, 0]	Not classified
16	17	7120	753	[0, 0, 0]	Not classified
17	18	3583	1029	[0, 0, 0]	Not classified
18	19	4030	2967	[0, 0, 0]	Not classified
19	20	3283	7897	[0, 0, 0]	Not classified
20	21	6756	4912	[18, 25, 133]	Tarred Road
21	22	4797	6444	[20, 20, 151]	Tarred Road
22	23	226	7226	[47, 42, 114]	Not classified
23	24	4297	5392	[0, 0, 0]	Not classified
24	25	6643	4750	[30, 18, 28]	Tarred Road

```
These are the number of objects predicted to be under each class:
Not classified    20
Tarred Road        5
Name: Class, dtype: int64
```

[ ]:

3 References

Detect Objects of Similar Color using OpenCV in Python. Available: https://techvidvan.com/tutoria objects-of-similar-color-using-opencv-in-python/ [29 April 2022].

Understanding and Building an Object Detection Model from Scratch in Python.2018. Available: https://www.analyticsvidhya.com/blog/2018/06/understanding-building-object-detection-model-python/ [28 April 2022].

Kumar, A. 2018. Solar Panel Installations in India using Satellite Imagery. Available: https://akash29.medium.com/rooftop-assessment-for-solar-installations-using-satellite-imagery-d8f8ee368dd0 [28 April 2022].