

#License Plate Recognition

#1. Data Loading, Pre-processing, data analysis, and understanding

```
import pandas as pd
import cv2
import os
from google.colab.patches import cv2_imshow # Import cv2_imshow for Colab

# Load CSV files
detection_df = pd.read_csv('/content/Licplatesdetection_train.csv')
recognition_df =
pd.read_csv('/content/Licplatesrecognition_train.csv')

# Load images for detection and recognition
detection_images_path =
'/content/sample_data/Licplatesdetection_train'
recognition_images_path =
'/content/sample_data/Licplatesrecognition_train'

# Example of loading a detection image and corresponding annotation
for i, row in detection_df.iterrows():
    img_path = os.path.join(detection_images_path, row['img_id'])
    img = cv2.imread(img_path)

    if img is None:
        print(f"Error: Image at path {img_path} could not be loaded.")
        continue # Skip this iteration if the image can't be loaded

    # Get bounding box coordinates and crop the license plate
    ymin, xmin, ymax, xmax = int(row['ymin']), int(row['xmin']),
    int(row['ymax']), int(row['xmax'])
    license_plate = img[ymin:ymax, xmin:xmax] # Cropping license plate region

    # Display the cropped license plate using cv2_imshow
    cv2_imshow(license_plate)
```



تونس 9189 137



تونس 237 121



تونس 8196 178

تونس 3782 159

تونس 4328 92



تونس 2600 117



تونس 8557

تونس 9189 137

تونس 830 147

تونس 9185 141

تونس 1383 166

تونس 5411 107

تونس 9922 170

تونس 153 152

تونس 8788 114

تونس 8467 177

تونس 9920 62

تونس 3734 96

152 تونس 4908

تونس 5411 107

تونس 2785 139

تونس 5823 107

تونس 7725 125

تونس 2738 172

تونس 3734 96

تونس 3222 76

تونس 5075 192

تونس 9604 138

تونس 2131 106

تونس 6719 59

77 تونس 4378

تونس 9658 60

تونس 6346 76

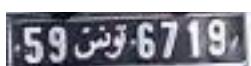
تونس 6789 105

تونس 7795 95

تونس 7487 140

تونس 3827 85

تونس 2131 106









تونس 3496 181

تونس 461 128

تونس 2131 106

تونس 159 179

تونس 5823 107

تونس 2131 106

تونس 2827 191

تونس 6361 59

تونس 6472 188

171337026

181 تونس 7519

149 تونس 8284

87 تونس 9294

98 تونس 528

117 تونس 3941

110 تونس 533

178 تونس 9010

111 تونس 5779

64 تونس 7992

تونس 668 195

تونس 5039 120

تونس 6434 112



تونس 5018 118

تونس 3458 94

تونس 6375 170

تونس 5255 155

تونس 5976 171

تونس 1040 62

تونس 8168 76



تونس 4533 132

تونس 9996 113

تونس 177 0467

تونس 8201 183

تونس 2904 57

تونس 7729 78

تونس 3782 159

تونس 557 85

تونس 5783 107

تونس 7729 78

تونس 5776 179

تونس 3019 74

تونس 1251 76

تونس 5474 194

تونس 2131 106



تونس 4908 152

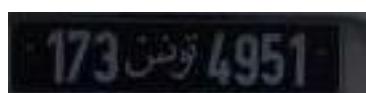
تونس 179 191

تونس 72 6300

تونس 8974 94

تونس 489 104

تونس 4924 105



133 تونس 6719

140 تونس 5408

149 تونس 7951

156 تونس 311

109 تونس 5513

78 تونس 7729

106 تونس 4788

93 تونس 3095

77 تونس 5883

179 تونس 5776

108 تونس 6472

153 تونس 6228

112 تونس 6434



تونس 4908 147

تونس 4837 95

64 تونس 4672

تونس 4001 81

تونس 5072 33

تونس 5373 191

تونس 2131 106

تونس 8269 152

تونس 8974 94

تونس 9365

تونس 5129

تونس 761

تونس 7121

تونس 7486

تونس 5976

تونس 6825

تونس 7121

تونس 8546

تونس 8576 · 112

تونس 5513 · 109

تونس 6375 · 170

تونس 2131 · 106

تونس 7121 · 179

تونس 9979 · 162

تونس 6062 · 124

تونس 8825 · 92

تونس 4573 · 61

تونس 7121 · 179

165 تونس 895

133 تونس 6719

53 تونس 291

77 تونس 4378

131 تونس 8840

159 تونس 7948

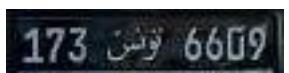
191 تونس 2827

170 تونس 6375

198 تونس 3095

110 تونس 4884

179 تونس 7121





173 تونس 6609

93 تونس 5599

130 تونس 8988

1735 تونس 182

150 تونس 8546

95 تونس 4837

170 تونس 6375



77 تونس 286

140 تونس 566

تونس 8729 81

تونس 8115 31

تونس 1341 99

تونس 9566 163

تونس 2765 125

تونس 5373 191

تونس 1546 82

تونس 6832 157

تونس 2131 106

تونس 6734 175

تونس 3759 68



تونس 2328
109

تونس 8546
150

تونس 286
77

تونس 239
114

تونس 8377
114

تونس 4837
95

تونس 3782
159

تونس 0341
68

تونس 2904
57

تونس 5075
192

تونس 2847
116

تونس 9587
77

155 تونس 4578

(141099185)

تونس 6472 188

تونس 1546 82

تونس 9146 133

تونس 8925 92

تونس 2823 135

تونس 7944 175

تونس 4950 / 80

تونس 489 - 104

تونس 1856 - 139

تونس 9979 - 162

تونس 9587 - 77

تونس 8105 - 131

تونس 2381 - 187

تونس 5018 118

تونس 7260 132

تونس 6516 156

تونس 745 149

تونس 2759 182

تونس 4924 105

تونس 6825 148

تونس 6537

تونس 7813 187

تونس 3734 96

تونس 237 121

تونس 4591 93

تونس 6375 170

[193 تونس 1374]

[128 تونس 1739]



[137 تونس 5115]

[74 تونس 5911]

[159 تونس 7948]

[60 تونس 9658]

[182 تونس 1735]

[149 تونس 7951]

[37 تونس 3507]

[175 تونس 7944]

[63 تونس 6751]



112 تونس 5380

195 تونس 668

99 تونس 1341

149 تونس 8284

165 تونس 895



172 تونس 4715





128 461



تونس 2131 · 106 ·

· 207 · تونس 168 ·

· 3230 · تونس 65 ·

تونس 5281 · 104 ·

تونس 1094 · 143 ·

تونس 1641 · 160 ·

تونس 8557 · 96 ·

تونس 2637 · 130 ·

تونس 7533 · 135 ·

تونس 1040 · 62 ·



تونس 8576 · 112

تونس 3469 · 149

تونس 8597 · 153

تونس 8840 · 131

تونس 9793 · 160

تونس 6789 · 105

تونس 5682 · 93

تونس 1901 · 40



118 تونس 5018



163 تونس 9566

63 تونس 366

57 تونس 4911

120 تونس 5039



93 تونس 140

تونس 6554
182

تونس 2955
160

تونس 4221
114

تونس 8729
81

تونس 4951
173

تونس 5526
58

تونس 9345
100

تونس 3230
65

تونس 2736
106

تونس 8974
94

تونس 8840 131

تونس 8769 77

تونس 2955 160

تونس 652 70

تونس 207 168

تونس 701 82

تونس 6823 77

تونس 179 191

تونس 9294 87

تونس 8888 58

تونس 3849 133

تونس 6657 118

تونس 4573 61

تونس 132 3446

تونس 5526

تونس 4573

تونس 6123

تونس 6898

تونس 6346

تونس 9010

تونس 7637

تونس 5018

تونس 1006

تونس 179 191

تونس 4425 128

تونس 1646 62

تونس 1451 156

تونس 7260 132

تونس 4062 176

تونس 5779 111

تونس 9920 62

تونس 489 104

تونس 4425 128

تونس 6823 77





194 تونس 5474

ن.ت 102098

تونس 4591 93

تونس 7645 170



تونس 2400 86

تونس 5378 125

تونس 8917 120

تونس 294 105

تونس 3647 163

تونس 5697 66

تونس 2589 79

تونس 3297 149

تونس 7878 165

تونس 2501 82

تونس 2759 182

تونس 8766 104

تونس 7304 114

تونس 4187 89

تونس 883 140

تونس 876 112

تونس 8892 90

تونس 9858 154

تونس 573 94

تونس 6431 175

تونس 668 195

114 تونس 7304

104 تونس 8766

181 تونس 2562

124 تونس 6418

106 تونس 5997

175 تونس 6431

127 تونس 9733

134 تونس 3970

90 تونس 9848

148 تونس 2903

111 تونس 1930

136 تونس 7602

107 تونس 5823

122 تونس 7105

تونس 9747 117

تونس 6088 125

تونس 3015 161

تونس 5906 135

تونس 5864 86

تونس 4230 126

تونس 472 147

تونس 412 158

تونس 3593 194

تونس 895 165

تونس 2499 137

تونس 8796 157

150 تونس 8996

4019
تونس 73

103 تونس 1451

129 تونس 2976

182 تونس 1735

165 تونس 6025

90 تونس 9848

129 تونس 9084

137 تونس 9983

84 تونس 2486

102 تونس 6927

164 تونس 9653

129 تونس 230

تونس 8703
116

تونس 6603
141

تونس 41
105

تونس 5301
82

تونس 4932
10233

تونس 5823
107

تونس 7121
179

تونس 4932
148

تونس 3550
101

تونس 3222
153

تونس 775
156

تونس 6744
161

تونس 4018
75

تونس 3404
170

تونس 5823
107

تونس 7497
88

تونس 8381
140



تونس 6543 184

تونس 8892 90

تونس 1383 166

تونس 2663 107

تونس 161 74

تونس 8894 159

تونس 1524 71

تونس 6302 172

تونس 7367 132

تونس 6646 65

تونس 573 94
WWW.POLO-CLASSIC.COM

تونس 7524 156

تونس 870 131

تونس 9996 113

تونس 5697 66

تونس 7304 95

تونس 3698 111

تونس 8703 116

تونس 103679

تونس 7292 71

تونس 5906 135

تونس 2310 99

تونس 1523 177

تونس 1310 41



تونس 6944 43

تونس 1524 71

تونس 7332 154

تونس 5898 39

155 تونس 9278

تونس 5333 100

65 تونس 6646

تونس 2903 148

تونس 449 84

تونس 6434 112

تونس 5979 68

تونس 3626 148

تونس 3690 181

تونس 8018 123



تونس 668 195

تونس 4392 54

تونس 4187 89

تونس 6285 45

تونس 841 30

تونس 3408 128

تونس 6810 107

تونس 2325 156

تونس 7645 170

تونس 3230 65

تونس 6404 84

تونس 8151 47

6994
تونس 52

77 تونس 4428

167 تونس 5598

106 تونس 1334

73 تونس 7689

133 تونس 5882

122 تونس 7016

96 تونس 3734

85 تونس 8265

148 تونس 2903

تونس 9687 136

تونس 188 8416

تونس 9128 145

تونس 9960 161

تونس 1066 146

تونس 3297 149

تونس 6651 70

تونس 322 164

تونس 7413 142

تونس 5599 93

تونس 4000 156

تونس 945 155

تونس 37 18 153

تونس 2340 130

تونس 6646 65

تونس 193

تونس 9593 152

تونس 1169 62

تونس 1334 106

تونس 8467 122

تونس 7944 175

تونس 2625 162

تونس 1022 181

تونس 7606 94

تونس 181 157

تونس 6603 141

تونس 1812 83

تونس 6707 130

تونس 4724 144

تونس 735 194

تونس 86 9785

تونس 8974 94

تونس 5823 107

تونس 1493 91

تونس 7105 122

تونس 9685 169

تونس 5419 119

تونس 7367 132

تونس 8677 188

تونس 8764 128

تونس 4636 90

تونس 3690 181

تونس 3458 94

تونس 8409 138

تونس 237 121

تونس 4350 52

تونس 9314 172

تونس 7119 165

تونس 66 8043

تونس 1884 104

تونس 8159 137

تونس 2340 130

تونس 6994 52

تونس 159 179

تونس 5178 98

تونس 870 131

تونس 1310 41

111 تونس 1930

133 تونس 1130

تونس 5194 56

تونس 1451 103

103679

تونس 7596 107

تونس 6996 48

تونس 159 179

تونس 1524 71

تونس 7821 115

تونس 6585 109

تونس 8312 163

تونس 2612 195

تونس 1735 182

تونس 1238 78

[114 تونس 7055]

[117 تونس 3989]

[121 تونس 6559]

[68 تونس 3759]

[191 تونس 9338]

[111 تونس 4151]

[138 تونس 2415]

[70 تونس 296]

[88 تونس 6610]

[83 تونس 1812]

[176 تونس 755]

[163 تونس 3168]

تونس 6064
115

تونس 5686
170

تونس 1451
103

تونس 2600
117

تونس 4413
134

تونس 9653
164

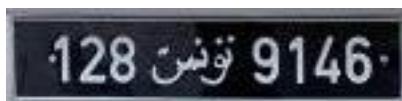
تونس 8107
68

تونس 7035
77

تونس 5882
93

تونس 3527
153

تونس 3403
99



تونس 5898 39

101 تونس 3978

تونس 5289 62

تونس 6227 105

تونس 4719 63

تونس 4640 120

تونس 5816 133

تونس 7987 117

تونس 3076 42

تونس 6532 71

تونس 3437 125

[153 تونس 3296]

[171 تونس 9707]

[172 تونس 9314]

[100 تونس 8109]

[9785
86 تونس]

[114 تونس 239]

[136 تونس 2146]

[194 تونس 558]

[170 تونس 7769]

[175 تونس 4465]

[82 تونس 2501]

[79 تونس 2589]

[109 تونس 6585]

[117 تونس 1938]

[117 تونس 3941]

تونس 1524 71

تونس 2333 154

تونس 956 188

تونس 9433 115

تونس 8703 116

تونس 2777 176

تونس 3293 78

تونس 4579 196

تونس 5368 145

تونس 2094 111

تونس 8377 114

· 135 تونس 4716 ·

· 108 تونس 5794 ·

· 92 تونس 589 ·

· 156 تونس 775 ·

· 71 تونس 7292 ·

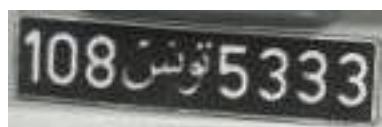
· 185 تونس 834 ·

· 78 تونس 140 ·

· 122 تونس 7105 ·

· 96 تونس 4579 ·

· 160 تونس 6356 ·



7951 تونس 149

7178 تونس 115

8018 تونس 128

1769 تونس 138

413 تونس 190

4579 تونس 96

4034 تونس 142

9955 تونس 86

5686 تونس 170

2347 تونس 162

9240 تونس 140

8084 تونس 171

6105 تونس 152

155 تونس 5239

168 تونس 1951

128 تونس 6283

74 تونس 366

169 تونس 7127

117 تونس 6491

86 تونس 4855

142 تونس 7100

132 تونس 6885

121 تونس 4909

167 تونس 2217

167 تونس 5102

تونس 2661 64

تونس 613 190

تونس 3437 125

تونس 2562 181

تونس 4228 188

تونس 2998 91

تونس 6075 69

تونس 1216 155

تونس 7972 108

تونس 3782 159

تونس 2527 58

تونس 6569 62

تونس 13 141

تونس 8151 47

تونس 3168 163

تونس 6108 162

تونس 6234 84

تونس 1077 130

تونس 755 176

تونس 589 92

تونس 59 127

تونس 5419 119

تونس 9733 127

تونس 5794 108

تونس 7706 72

· 147 تونس 6538 ·

· 75 تونس 4018 ·

· 109 تونس 658 ·

· 132 تونس 7367 ·

· 145 تونس 4948 ·

· 192 تونس 6756 ·

· 70 تونس 652 ·

· 142 تونس 8217 ·

· 128 تونس 9146 ·

· 109 تونس 6585 ·

· 95 تونس 3967 ·

· 122 تونس 7105 ·

تونس 9884 117

تونس 7279 159

تونس 4640 120

تونس 8531 83

تونس 7602 136

تونس 2047 116

تونس 5826 178

تونس 7279 159

تونس 9454 178

2002 3

تونس 2415 138

تونس 7122 89

تونس 9630 118

تونس 8086 128

تونس 6320 75

تونس 3169 87

تونس 237 121

تونس 6646 65



تونس 3970 134

تونس 2903 148

تونس 8869 139

تونس 2225 87



تونس 9146 128

تونس 7506 117

تونس 1006 191

تونس 4843 94

تونس 1234 119

تونس 2658 95

تونس 4280 78

تونس 1396 131

تونس 116 131

تونس 6707 130

تونس 3973 129

تونس 6926 119

تونس 564 86

تونس 165 895

تونس 194 3593

تونس 137 8453

تونس 108 4002

تونس 148 2903

تونس 163 8312

تونس 194 3593

تونس 105 294

تونس 119 9133

تونس 175 4637

تونس 100 8343

تونس 156 6516

تونس 156 864

تونس 153 3847

تونس 121 4099

تونس 5239 176

تونس 8723 53

تونس 4562 91

تونس 8467 122

تونس 9170 87

تونس 536 158

تونس 2087 139

تونس 9393 83

تونس 895 165





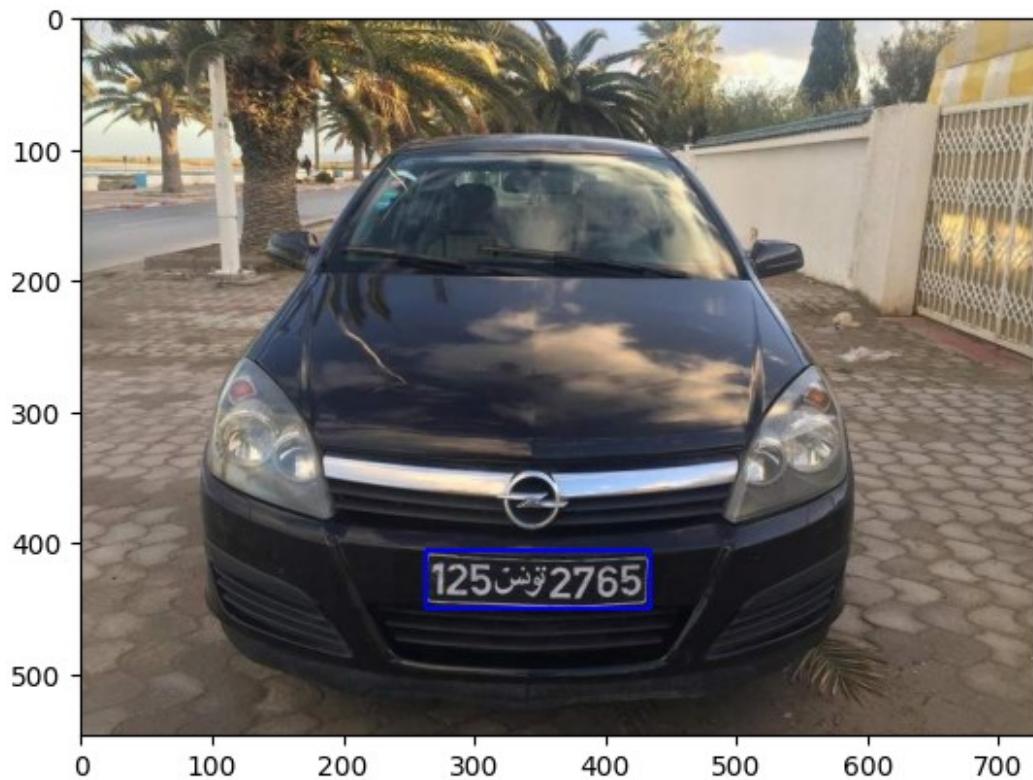
#2. Data exploration - EDA Exploratory Data Analysis

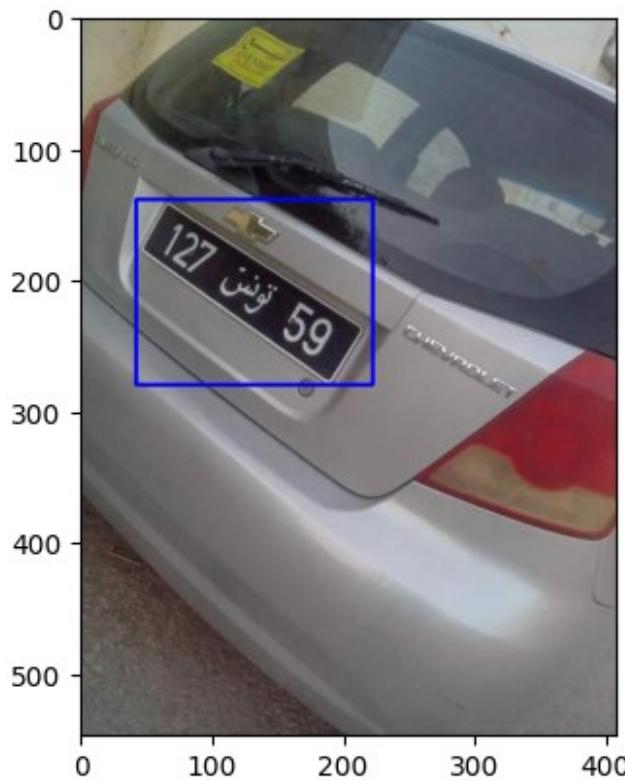
```
import matplotlib.pyplot as plt
import seaborn as sns

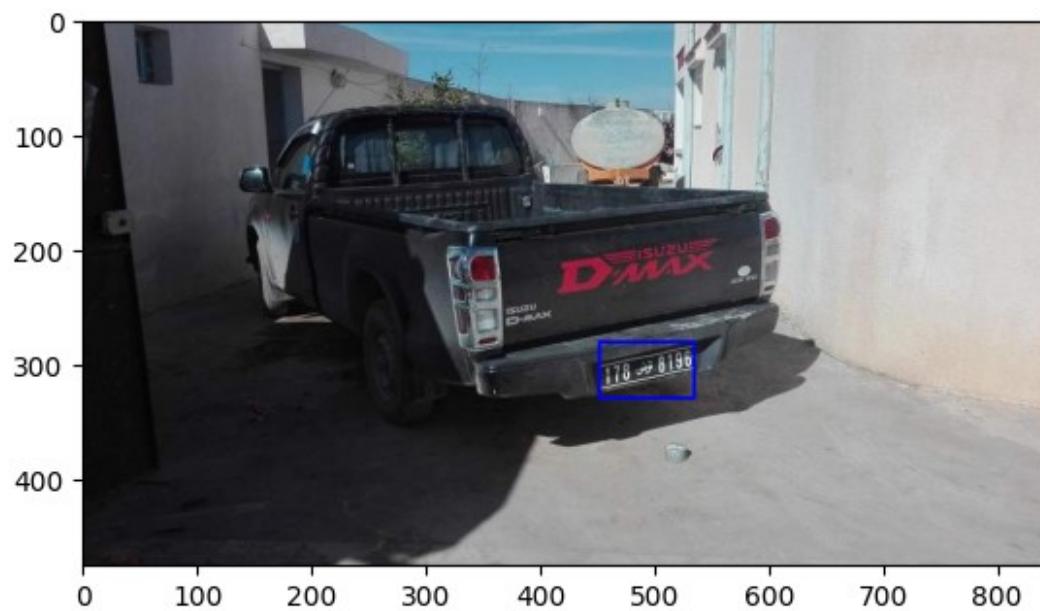
# Visualize some of the bounding boxes from detection dataset
def plot_bounding_boxes(df, img_folder):
    for i in range(10): # Display 5 samples
        img_path = os.path.join(img_folder, df.iloc[i]['img_id'])
        img = cv2.imread(img_path)
        ymin, xmin, ymax, xmax = df.iloc[i]['ymin'], df.iloc[i]
        ['xmin'], df.iloc[i]['ymax'], df.iloc[i]['xmax']
        cv2.rectangle(img, (xmin, ymin), (xmax, ymax), (255, 0, 0), 2)
        plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        plt.show()

plot_bounding_boxes(detection_df, detection_images_path)
```











This visualization helps to inspect the quality of the detected bounding boxes, ensuring that they correctly encompass the license plates. It allows for a quick visual verification of the detection performance and can help identify any issues with misaligned or incorrectly sized bounding boxes.

```
import random

def plot_random_bounding_boxes(df, img_folder, num_samples=5):
    # Exclude the first 5 rows and select random samples
    available_indices = df.index[5:] # Skip the first 5 indices
    random_indices = random.sample(list(available_indices),
num_samples) # Randomly select 5 indices

    for i in random_indices:
        img_path = os.path.join(img_folder, df.iloc[i]['img_id'])
        img = cv2.imread(img_path)

        if img is not None:
            ymin, xmin, ymax, xmax = (
                int(df.iloc[i]['ymin']),
                int(df.iloc[i]['xmin']),
                int(df.iloc[i]['ymax']),
                int(df.iloc[i]['xmax']),
            )
            cv2.rectangle(img, (xmin, ymin), (xmax, ymax), (255, 0,
0), 2)
            plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
            plt.axis('off') # Turn off axis numbers and ticks
            plt.show()

# Call the function to plot random bounding boxes
plot_random_bounding_boxes(detection_df, detection_images_path)
```







The above one is the same but with 5 random objects from the dataset.

```
# detection_df
# recognition_df
# test_images_path

# Display the first few rows of each dataset
print("Detection data Sample:")
print(detection_df.head())

print("\nRecognition Data Sample:")
print(recognition_df.head())

Detection data Sample:
    img_id  ymin  xmin  ymax  xmax  box_width  box_height
0    1.jpg   276    94   326   169        75          50
1   10.jpg   311   395   344   444        49          33
2  100.jpg   406   263   450   434       171          44
3  101.jpg   283   363   315   494       131          32
4  102.jpg   139    42   280   222       180         141

Recognition Data Sample:
    img_id      text
0    0.jpg  117T3989
1    1.jpg  128T8086
2   10.jpg   94T3458
```

```
3 100.jpg 133T6719
4 101.jpg 68T5979
```

The look into the datasets.

```
# Display the shape and data types
print("Detection data Sample:")
print(detection_df.info())

print("\nRecognition Data Sample:")
print(recognition_df.info())

Detection data Sample:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 900 entries, 0 to 899
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  --  
 0   img_id      900 non-null    object 
 1   ymin        900 non-null    int64  
 2   xmin        900 non-null    int64  
 3   ymax        900 non-null    int64  
 4   xmax        900 non-null    int64  
 5   box_width   900 non-null    int64  
 6   box_height  900 non-null    int64  
dtypes: int64(6), object(1)
memory usage: 49.3+ KB
None

Recognition Data Sample:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 900 entries, 0 to 899
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  --  
 0   img_id   900 non-null    object 
 1   text     900 non-null    object 
dtypes: object(2)
memory usage: 14.2+ KB
None
```

A brief info fo the datasets given.

```
print("Detection data Sample:")
print(detection_df.isnull().sum())

print("\nRecognition Data Sample:")
print(recognition_df.isnull().sum())
```

```
Detection data Sample:
```

```
img_id      0  
ymin       0  
xmin       0  
ymax       0  
xmax       0  
box_width   0  
box_height  0  
dtype: int64
```

```
Recognition Data Sample:
```

```
img_id      0  
text        0  
dtype: int64
```

Checking for the null values.

```
print("Detection data Sample:")  
print(detection_df.describe())
```

```
print("\nRecognition Data Sample:")  
print(recognition_df.describe())
```

```
Detection data Sample:
```

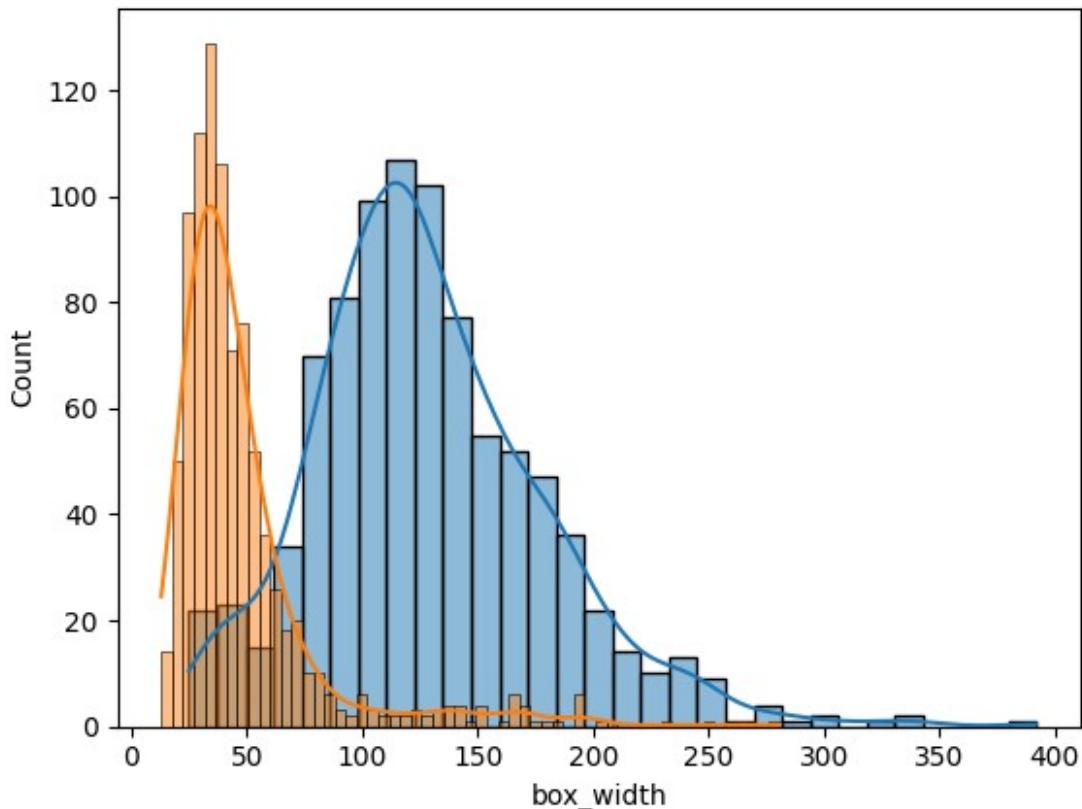
	ymin	xmin	ymax	xmax	box_width
box_height					
count	900.000000	900.000000	900.000000	900.000000	900.000000
900.000000					
mean	308.596667	240.044444	356.741111	368.750000	128.705556
48.144444					
std	75.768988	142.368243	71.955977	149.029253	50.294405
33.908094					
min	14.000000	1.000000	121.000000	84.000000	25.000000
13.000000					
25%	263.000000	129.000000	313.000000	245.750000	96.000000
30.000000					
50%	314.000000	212.000000	359.000000	338.000000	122.000000
38.500000					
75%	363.000000	320.250000	410.000000	485.250000	157.000000
53.000000					
max	525.000000	698.000000	547.000000	823.000000	392.000000
281.000000					

```
Recognition Data Sample:
```

	img_id	text
count	900	900
unique	900	596
top	0.jpg	106T2131
freq	1	10

The description of the dataset.

```
# Analyze distribution of bounding box sizes
detection_df['box_width'] = detection_df['xmax'] -
detection_df['xmin']
detection_df['box_height'] = detection_df['ymax'] -
detection_df['ymin']
sns.histplot(detection_df['box_width'], kde=True)
sns.histplot(detection_df['box_height'], kde=True)
plt.show()
```



Bounding Box Width and Height Calculation: It computes the width and height of each bounding box by subtracting the minimum coordinates from the maximum coordinates (box_width and box_height).

Distribution Visualization: Two histograms are generated using Seaborn (sns.histplot), one for the widths and one for the heights. Each histogram includes a Kernel Density Estimate (KDE) overlay for a smoother representation of the data distribution.

```
!pip install seaborn
import matplotlib.pyplot as plt
import seaborn as sns # Import seaborn for plotting
```

```
Requirement already satisfied: seaborn in
/usr/local/lib/python3.10/dist-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in
/usr/local/lib/python3.10/dist-packages (from seaborn) (1.26.4)
Requirement already satisfied: pandas>=1.2 in
/usr/local/lib/python3.10/dist-packages (from seaborn) (2.2.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in
/usr/local/lib/python3.10/dist-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.0)
Requirement already satisfied: cybler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.7)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.1)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn)
(2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn)
(2024.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)

# Create new columns for bounding box dimensions
detection_df['box_width'] = detection_df['xmax'] -
detection_df['xmin']
detection_df['box_height'] = detection_df['ymax'] -
detection_df['ymin']

# Visualizing the distribution of bounding box sizes
plt.figure(figsize=(12, 5))
```

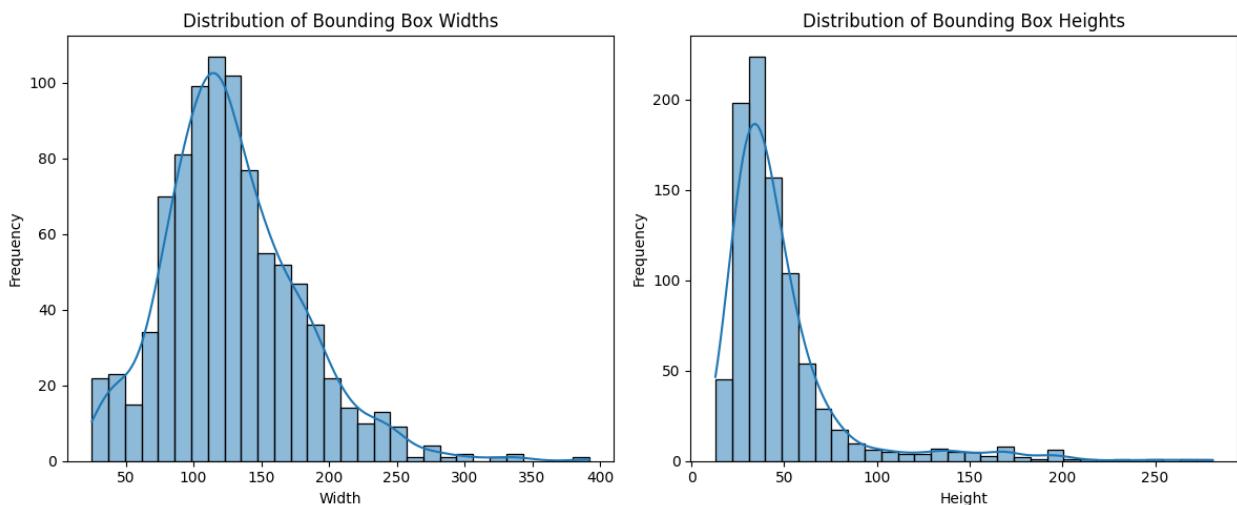
```

plt.subplot(1, 2, 1)
sns.histplot(detection_df['box_width'], bins=30, kde=True)
plt.title('Distribution of Bounding Box Widths')
plt.xlabel('Width')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
sns.histplot(detection_df['box_height'], bins=30, kde=True)
plt.title('Distribution of Bounding Box Heights')
plt.xlabel('Height')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()

```



New Columns: It creates `box_width` and `box_height` columns in the DataFrame by calculating the differences between the corresponding coordinates of the bounding boxes which are the number plates itself.

Purpose: This analysis helps identify common dimensions, detect outliers, and assess the effectiveness of the detection model, guiding potential optimizations.

```

def plot_more_bounding_boxes(df, img_folder, num_samples=5):
    random_indices = random.sample(range(len(df)), num_samples) # Randomly select samples

    for i in random_indices:
        img_path = os.path.join(img_folder, df.iloc[i]['img_id'])
        img = cv2.imread(img_path)

        if img is not None:
            ymin, xmin, ymax, xmax = (
                int(df.iloc[i]['ymin']),

```

```

        int(df.iloc[i]['xmin']),
        int(df.iloc[i]['ymax']),
        int(df.iloc[i]['xmax']),
    )
    cv2.rectangle(img, (xmin, ymin), (xmax, ymax), (255, 0,
0), 2)
    plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    plt.axis('off') # Turn off axis numbers and ticks
    plt.show()

# Call the function to visualize more samples
plot_more_bounding_boxes(detection_df, detection_images_path,
num_samples=5)

```

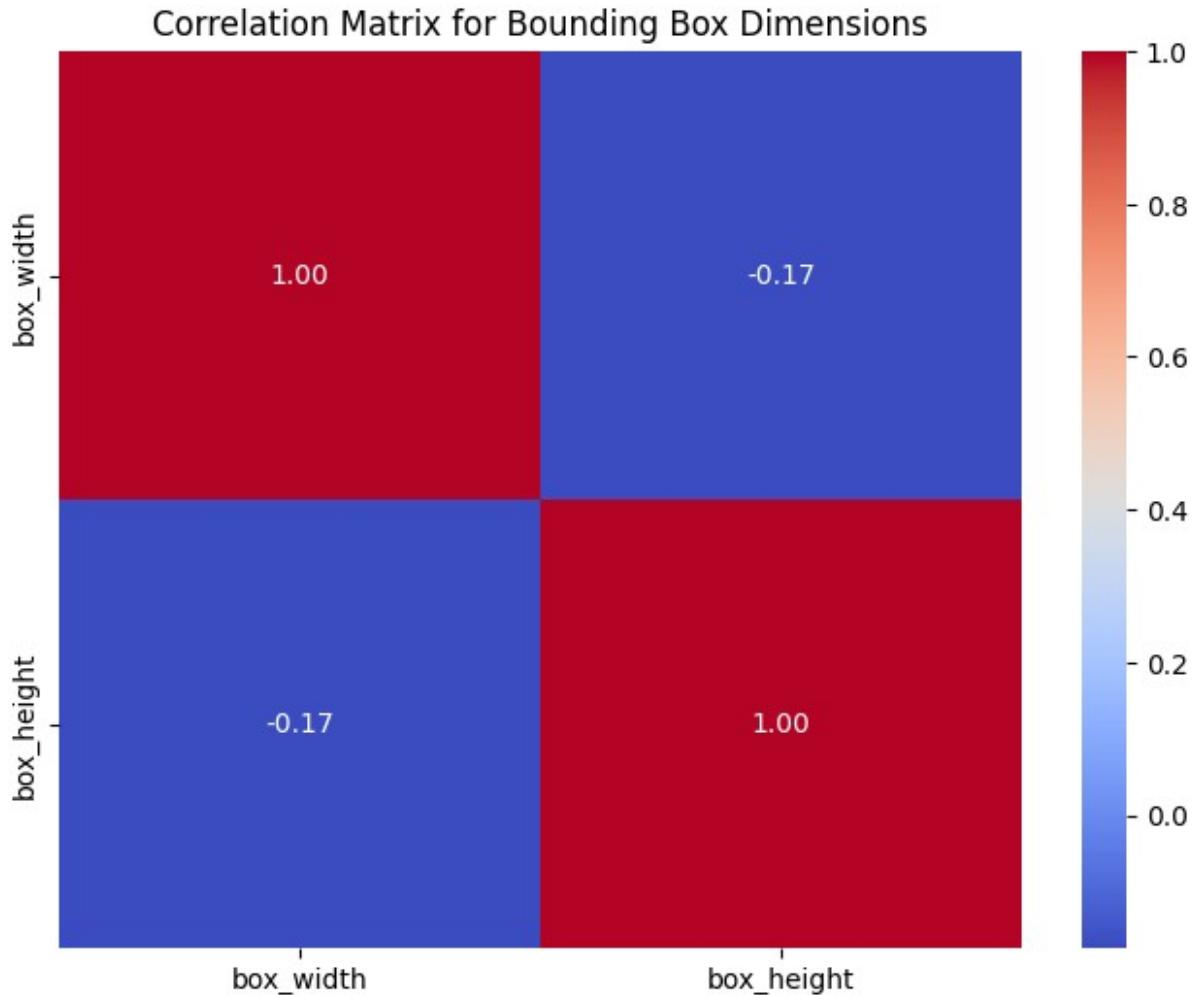






Again any random 5 samples.

```
# Correlation heatmap for bounding box dimensions
plt.figure(figsize=(8, 6))
correlation_matrix = detection_df[['box_width', 'box_height']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt='.2f')
plt.title('Correlation Matrix for Bounding Box Dimensions')
plt.show()
```



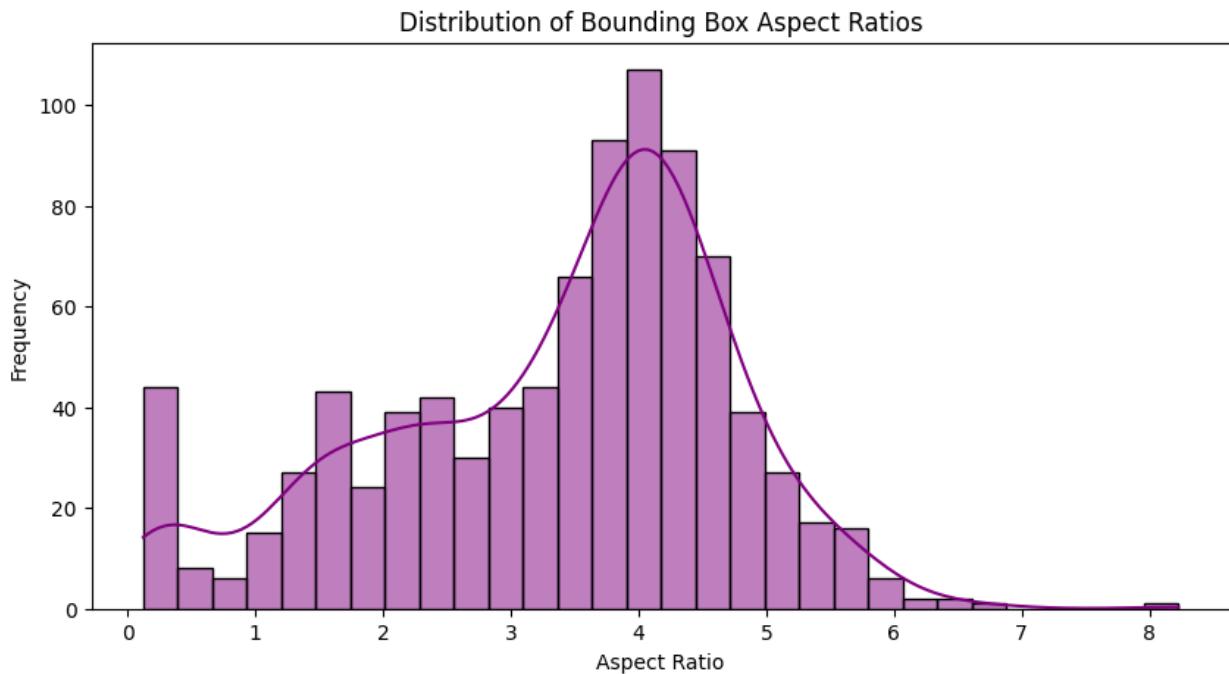
This is a correlation matrix, and it calculates the correlation coefficients between the box_width and box_height columns of the detection_df DataFrame. This matrix shows how strongly these dimensions are related to each other.

The Purpose of this would be, analysis that helps to understand the relationship between width and height of the bounding boxes, indicating whether changes in one dimension are associated with changes in the other. A strong correlation could imply a consistent aspect ratio in the bounding boxes.

```
# Calculate aspect ratio
detection_df['aspect_ratio'] = detection_df['box_width'] /
detection_df['box_height']

# Visualize aspect ratio distribution
plt.figure(figsize=(10, 5))
sns.histplot(detection_df['aspect_ratio'], bins=30, kde=True,
color='purple')
plt.title('Distribution of Bounding Box Aspect Ratios')
plt.xlabel('Aspect Ratio')
```

```
plt.ylabel('Frequency')
plt.show()
```



There is a new column, `aspect_ratio`, represents the relationship between the width and height of the bounding boxes.

This analysis helps to understand how the dimensions of the bounding boxes vary in terms of their shape. A consistent aspect ratio across bounding boxes may indicate uniformity in the size and shape of the detected objects, which in this case are likely license plates.

```
# Function to get image dimensions
def get_image_dimensions(img_folder):
    widths, heights = [], []
    for img_file in os.listdir(img_folder):
        img_path = os.path.join(img_folder, img_file)
        img = cv2.imread(img_path)
        if img is not None:
            heights.append(img.shape[0])
            widths.append(img.shape[1])
    return widths, heights

detection_widths, detection_heights =
get_image_dimensions(detection_images_path)
recognition_widths, recognition_heights =
get_image_dimensions(recognition_images_path)

# Plot the distribution of image dimensions
plt.figure(figsize=(12, 5))
```

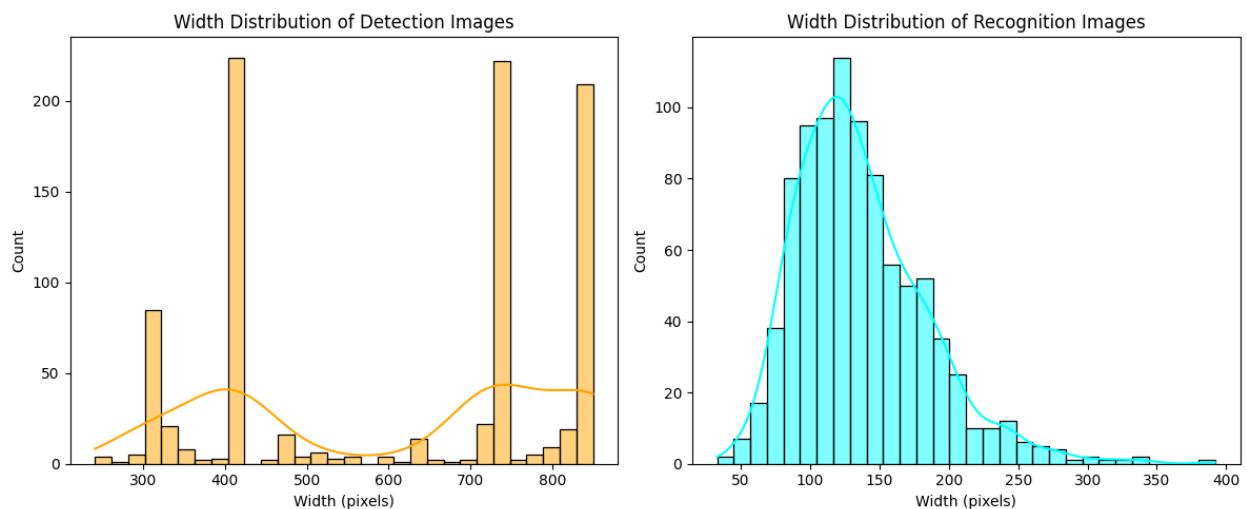
```

plt.subplot(1, 2, 1)
sns.histplot(detection_widths, bins=30, kde=True, color='orange')
plt.title('Width Distribution of Detection Images')
plt.xlabel('Width (pixels)')

plt.subplot(1, 2, 2)
sns.histplot(recognition_widths, bins=30, kde=True, color='cyan')
plt.title('Width Distribution of Recognition Images')
plt.xlabel('Width (pixels)')

plt.tight_layout()
plt.show()

```

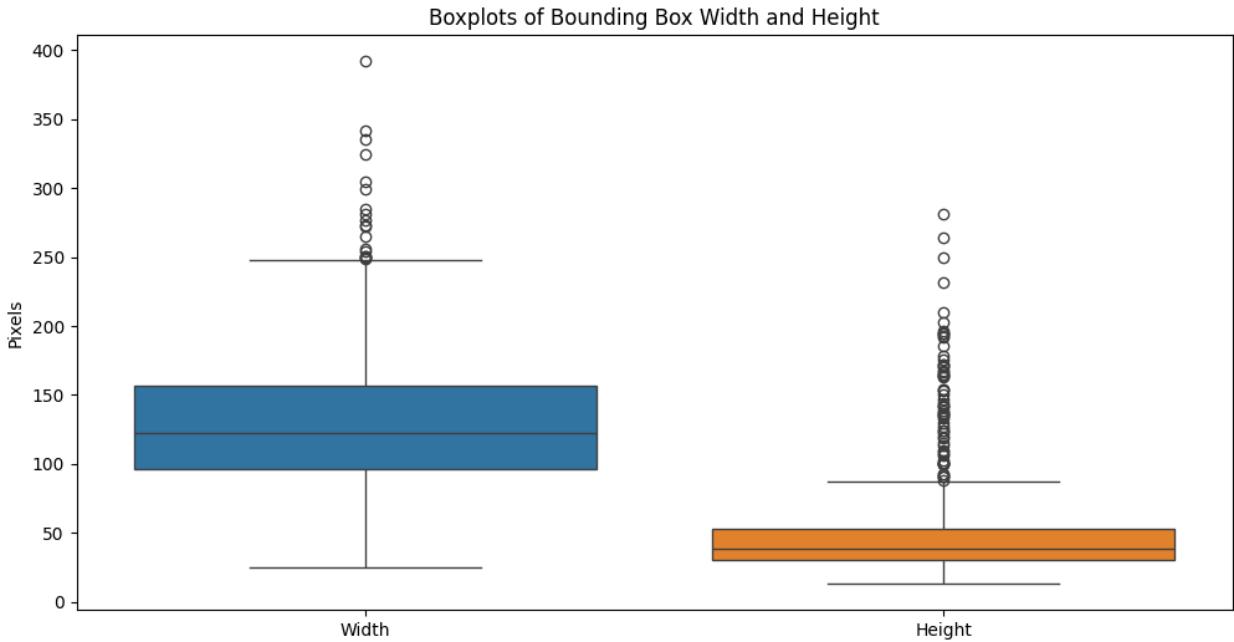


This basically shows us the graph of how the width(in terms of pixels) differs in the detection dataset and the recognition dataset and this helps identify the range and frequency of image sizes, providing insights into the variability of image dimensions within the datasets.

```

plt.figure(figsize=(12, 6))
sns.boxplot(data=detection_df[['box_width', 'box_height']])
plt.title('Boxplots of Bounding Box Width and Height')
plt.ylabel('Pixels')
plt.xticks([0, 1], ['Width', 'Height'])
plt.show()

```



The most efficient way to showcase the Width and Height ranges in the dataset. Width -> 100 - 120

Height -> 30 - 50

```
import numpy as np

def compute_image_brightness(img_folder):
    brightness = []
    for img_file in os.listdir(img_folder):
        img_path = os.path.join(img_folder, img_file)
        img = cv2.imread(img_path)
        if img is not None:
            brightness.append(np.mean(cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY)))
    return brightness

detection_brightness = compute_image_brightness(detection_images_path)
recognition_brightness =
compute_image_brightness(recognition_images_path)

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.histplot(detection_brightness, bins=30, kde=True, color='red')
plt.title('Brightness Distribution of Detection Images')
plt.xlabel('Brightness')

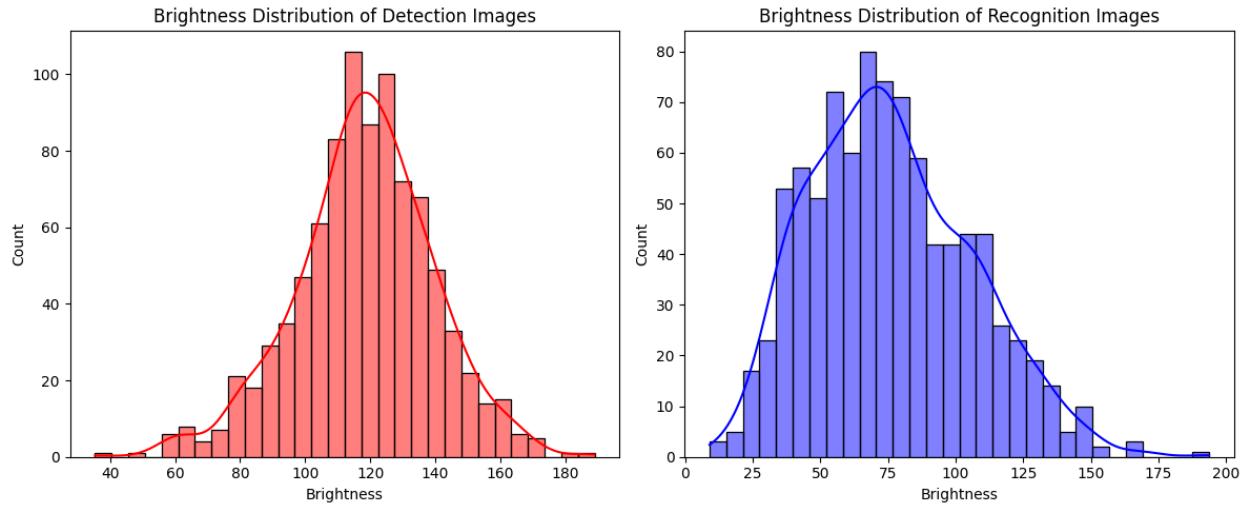
plt.subplot(1, 2, 2)
sns.histplot(recognition_brightness, bins=30, kde=True, color='blue')
plt.title('Brightness Distribution of Recognition Images')
```

```

plt.xlabel('Brightness')

plt.tight_layout()
plt.show()

```



Brightness Calculation: For each image, it converts the image to grayscale and calculates the average brightness value, storing these values in a list.

Visualization: It then creates histograms for both datasets:

Detection Images: Brightness distribution is plotted in red. **Recognition Images:** Brightness distribution is plotted in blue.

```

def plot_sample_images_with_crops(df, img_folder, num_samples=5):
    random_indices = random.sample(range(len(df)), num_samples) # Randomly select samples

    for i in random_indices:
        img_path = os.path.join(img_folder, df.iloc[i]['img_id'])
        img = cv2.imread(img_path)
        ymin, xmin, ymax, xmax = df.iloc[i]['ymin'], df.iloc[i]['xmin'], df.iloc[i]['ymax'], df.iloc[i]['xmax']
        license_plate_crop = img[ymin:ymax, xmin:xmax] # Crop the license plate region

        plt.figure(figsize=(10, 5))
        plt.subplot(1, 2, 1)
        plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        plt.title('Original Image')
        plt.axis('off')

        plt.subplot(1, 2, 2)
        plt.imshow(cv2.cvtColor(license_plate_crop,
cv2.COLOR_BGR2RGB))

```

```
plt.title('Cropped License Plate')
plt.axis('off')

plt.show()

plot_sample_images_with_crops(detection_df, detection_images_path,
num_samples=5)
```

Original Image



Cropped License Plate



Original Image



Cropped License Plate



Cropped License Plate



Original Image



Original Image



Cropped License Plate





This does crop out the number plates out of the main image i.e., fetching data to recognize from the detection step.

It creates a side-by-side plot for each sample.

- The left side displays the original image.
- The right side shows the cropped license plate region.
- Each image is displayed without axes for better visual clarity.

```
# Function to calculate image resolutions
def get_image_resolutions(img_folder):
    resolutions = []
    for img_file in os.listdir(img_folder):
        img_path = os.path.join(img_folder, img_file)
        img = cv2.imread(img_path)
        if img is not None:
            resolutions.append(img.shape[0] * img.shape[1]) # Height
* Width
    return resolutions

detection_resolutions = get_image_resolutions(detection_images_path)
recognition_resolutions =
get_image_resolutions(recognition_images_path)

plt.figure(figsize=(12, 5))
sns.histplot(detection_resolutions, bins=30, kde=True, color='green')
plt.title('Resolution Distribution of Detection Images')
plt.xlabel('Resolution (pixels)')
plt.ylabel('Frequency')
plt.show()
```

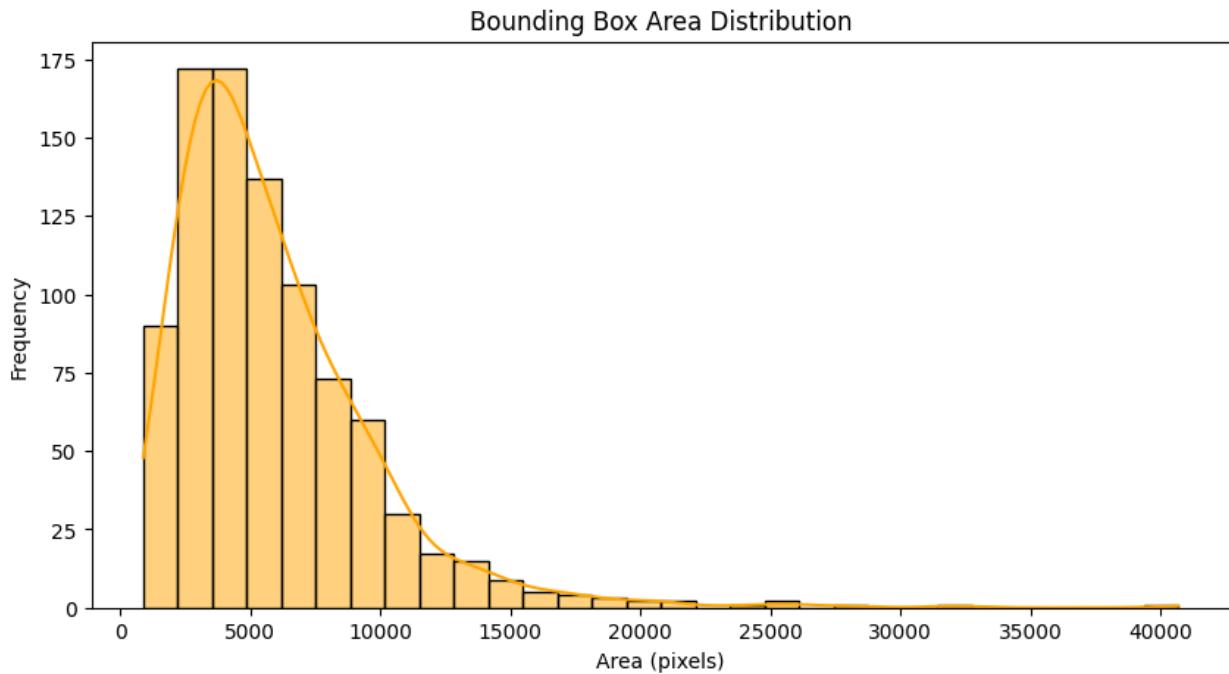


The frequency(number of images) of the dataset distributed based on the clarity/Resolution i.e., the pixel number

The function `get_image_resolutions` iterates over all image files in the specified folder. For each image, it reads the image and computes its resolution by multiplying the height (`img.shape[0]`) by the width (`img.shape[1]`). The computed resolutions are stored in a list.

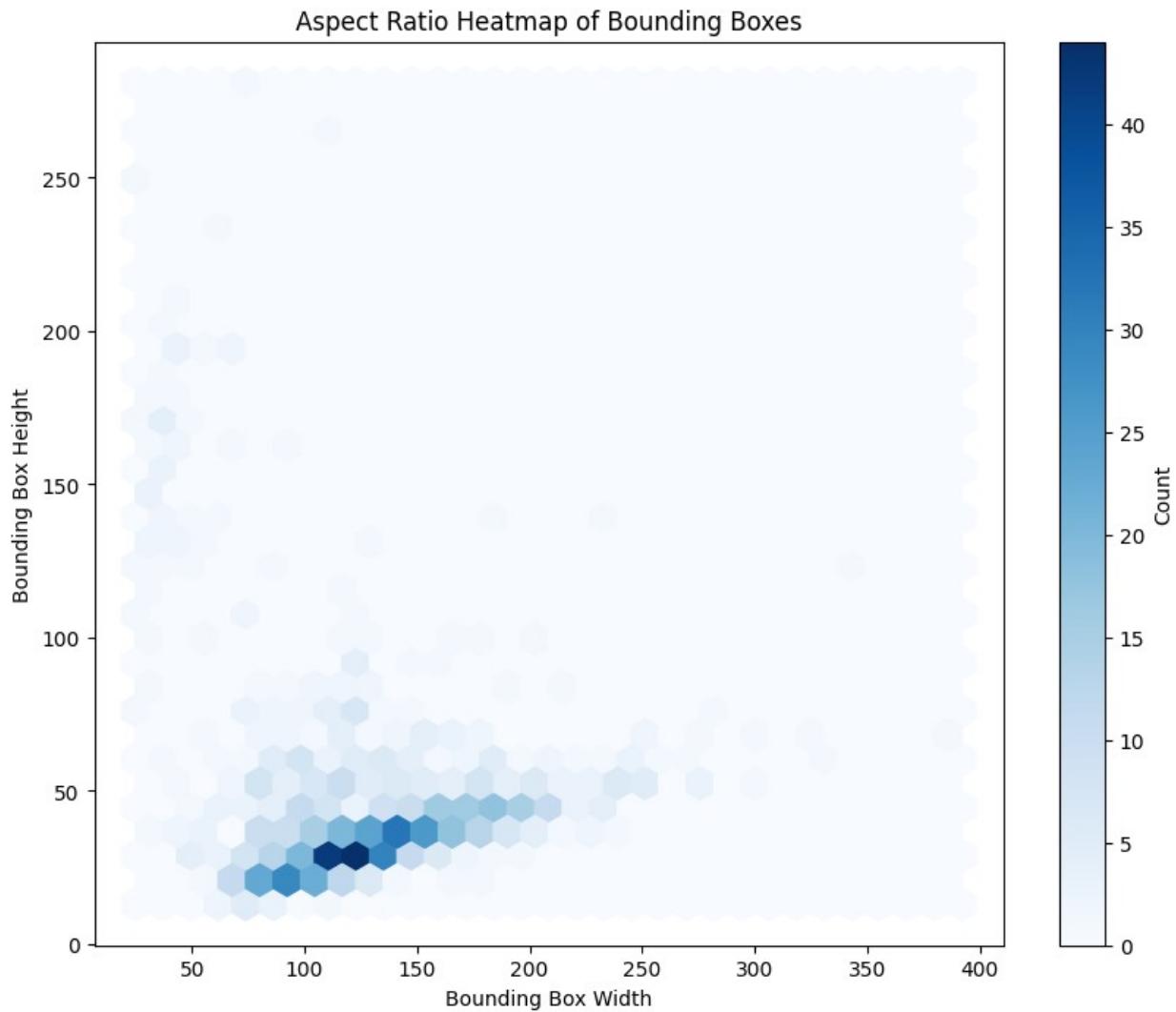
```
# Calculate bounding box area
detection_df['box_area'] = (detection_df['xmax'] -
detection_df['xmin']) * (detection_df['ymax'] - detection_df['ymin'])

# Visualize bounding box area distribution
plt.figure(figsize=(10, 5))
sns.histplot(detection_df['box_area'], bins=30, kde=True,
color='orange')
plt.title('Bounding Box Area Distribution')
plt.xlabel('Area (pixels)')
plt.ylabel('Frequency')
plt.show()
```



More depth about the resolution as this visualization helps in understanding the distribution of bounding box sizes, indicating the commonality of certain area sizes and any potential outliers in the dataset.

```
plt.figure(figsize=(10, 8))
plt.hexbin(detection_df['box_width'], detection_df['box_height'],
gridsize=30, cmap='Blues')
plt.colorbar(label='Count')
plt.title('Aspect Ratio Heatmap of Bounding Boxes')
plt.xlabel('Bounding Box Width')
plt.ylabel('Bounding Box Height')
plt.show()
```



#3. Model building

```
!sudo apt install tesseract-ocr
!pip install pytesseract

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  tesseract-ocr-eng tesseract-ocr-osd
The following NEW packages will be installed:
  tesseract-ocr tesseract-ocr-eng tesseract-ocr-osd
0 upgraded, 3 newly installed, 0 to remove and 49 not upgraded.
Need to get 4,816 kB of archives.
```

```
After this operation, 15.6 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-
ocr-eng all 1:4.00~git30-7274cfa-1.1 [1,591 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-
ocr-osd all 1:4.00~git30-7274cfa-1.1 [2,990 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tesseract-
ocr amd64 4.1.1-2.1build1 [236 kB]
Fetched 4,816 kB in 2s (2,035 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog
based frontend cannot be used. at
/usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78, <>> line 3.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package tesseract-ocr-eng.
(Reading database ... 123622 files and directories currently
installed.)
Preparing to unpack .../tesseract-ocr-eng_1%3a4.00~git30-7274cfa-
1.1_all.deb ...
Unpacking tesseract-ocr-eng (1:4.00~git30-7274cfa-1.1) ...
Selecting previously unselected package tesseract-ocr-osd.
Preparing to unpack .../tesseract-ocr-osd_1%3a4.00~git30-7274cfa-
1.1_all.deb ...
Unpacking tesseract-ocr-osd (1:4.00~git30-7274cfa-1.1) ...
Selecting previously unselected package tesseract-ocr.
Preparing to unpack .../tesseract-ocr_4.1.1-2.1build1_amd64.deb ...
Unpacking tesseract-ocr (4.1.1-2.1build1) ...
Setting up tesseract-ocr-eng (1:4.00~git30-7274cfa-1.1) ...
Setting up tesseract-ocr-osd (1:4.00~git30-7274cfa-1.1) ...
Setting up tesseract-ocr (4.1.1-2.1build1) ...
Processing triggers for man-db (2.10.2-1) ...
Collecting pytesseract
  Downloading pytesseract-0.3.13-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: packaging>=21.3 in
/usr/local/lib/python3.10/dist-packages (from pytesseract) (24.1)
Requirement already satisfied: Pillow>=8.0.0 in
/usr/local/lib/python3.10/dist-packages (from pytesseract) (10.4.0)
Downloading pytesseract-0.3.13-py3-none-any.whl (14 kB)
Installing collected packages: pytesseract
Successfully installed pytesseract-0.3.13

!git clone https://github.com/ultralytics/yolov5.git
%cd yolov5
!pip install -r requirements.txt

Cloning into 'yolov5'...
remote: Enumerating objects: 17022, done.
remote: Counting objects: 100%
```

```
(217/217), done.ote: Compressing objects: 100% (146/146), done.ote:  
Total 17022 (delta 122), reused 131 (delta 71), pack-reused 16805  
(from 1)ent already satisfied: gitpython>=3.1.30 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 5)) (3.1.43)  
Requirement already satisfied: matplotlib>=3.3 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 6)) (3.7.1)  
Requirement already satisfied: numpy>=1.23.5 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 7)) (1.26.4)  
Requirement already satisfied: opencv-python>=4.1.1 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 8)) (4.10.0.84)  
Requirement already satisfied: pillow>=10.3.0 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 9)) (10.4.0)  
Requirement already satisfied: psutil in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 10)) (5.9.5)  
Requirement already satisfied: PyYAML>=5.3.1 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 11)) (6.0.2)  
Requirement already satisfied: requests>=2.32.2 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 12)) (2.32.3)  
Requirement already satisfied: scipy>=1.4.1 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 13)) (1.13.1)  
Collecting thop>=0.1.1 (from -r requirements.txt (line 14))  
  Downloading thop-0.1.1.post2209072238-py3-none-any.whl.metadata (2.7 kB)  
Requirement already satisfied: torch>=1.8.0 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 15)) (2.4.1+cu121)  
Requirement already satisfied: torchvision>=0.9.0 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 16)) (0.19.1+cu121)  
Requirement already satisfied: tqdm>=4.66.3 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 17)) (4.66.5)  
Collecting ultralytics>=8.2.34 (from -r requirements.txt (line 18))  
  Downloading ultralytics-8.3.20-py3-none-any.whl.metadata (34 kB)  
Requirement already satisfied: pandas>=1.1.4 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 27)) (2.2.2)  
Requirement already satisfied: seaborn>=0.11.0 in  
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt  
(line 28)) (0.13.2)
```

```
Requirement already satisfied: setuptools>=70.0.0 in
/usr/local/lib/python3.10/dist-packages (from -r requirements.txt
(line 42)) (75.1.0)
Requirement already satisfied: gitdb<5,>=4.0.1 in
/usr/local/lib/python3.10/dist-packages (from gitpython>=3.1.30->-r
requirements.txt (line 5)) (4.0.11)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r
requirements.txt (line 6)) (1.3.0)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r
requirements.txt (line 6)) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r
requirements.txt (line 6)) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r
requirements.txt (line 6)) (1.4.7)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r
requirements.txt (line 6)) (24.1)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r
requirements.txt (line 6)) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r
requirements.txt (line 6)) (2.8.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->-r
requirements.txt (line 12)) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->-r
requirements.txt (line 12)) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->-r
requirements.txt (line 12)) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->-r
requirements.txt (line 12)) (2024.8.30)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r
requirements.txt (line 15)) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r
requirements.txt (line 15)) (4.12.2)
Requirement already satisfied: sympy in
/usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r
requirements.txt (line 15)) (1.13.3)
Requirement already satisfied: networkx in
```

```
/usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r
requirements.txt (line 15)) (3.4.1)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r
requirements.txt (line 15)) (3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r
requirements.txt (line 15)) (2024.6.1)
Requirement already satisfied: py-cpuinfo in
/usr/local/lib/python3.10/dist-packages (from ultralytics>=8.2.34->-r
requirements.txt (line 18)) (9.0.0)
Collecting ultralytics-thop>=2.0.0 (from ultralytics>=8.2.34->-r
requirements.txt (line 18))
  Downloading ultralytics_thop-2.0.9-py3-none-any.whl.metadata (9.3
kB)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas>=1.1.4->-r
requirements.txt (line 27)) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas>=1.1.4->-r
requirements.txt (line 27)) (2024.2)
Requirement already satisfied: smmap<6,>=3.0.1 in
/usr/local/lib/python3.10/dist-packages (from gitdb<5,>=4.0.1-
>gitpython>=3.1.30->-r requirements.txt (line 5)) (5.0.1)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib>=3.3->-r requirements.txt (line 6)) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.8.0->-r
requirements.txt (line 15)) (3.0.2)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.8.0->-r
requirements.txt (line 15)) (1.3.0)
Downloading thop-0.1.1.post2209072238-py3-none-any.whl (15 kB)
Downloading ultralytics-8.3.20-py3-none-any.whl (876 kB)
----- 876.6/876.6 kB 15.3 MB/s eta
0:00:00
```

#4.Result and Accuracy of the character recognition from the license plate

```
!pip install easyocr

Collecting easyocr
  Downloading easyocr-1.7.2-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: torch in
/usr/local/lib/python3.10/dist-packages (from easyocr) (2.4.1+cu121)
Requirement already satisfied: torchvision>=0.5 in
/usr/local/lib/python3.10/dist-packages (from easyocr) (0.19.1+cu121)
Requirement already satisfied: opencv-python-headless in
```

```
/usr/local/lib/python3.10/dist-packages (from easyocr) (4.10.0.84)
Requirement already satisfied: scipy in
/usr/local/lib/python3.10/dist-packages (from easyocr) (1.13.1)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from easyocr) (1.26.4)
Requirement already satisfied: Pillow in
/usr/local/lib/python3.10/dist-packages (from easyocr) (10.4.0)
Requirement already satisfied: scikit-image in
/usr/local/lib/python3.10/dist-packages (from easyocr) (0.24.0)
Collecting python-bidi (from easyocr)
  Downloading python_bidi-0.6.3-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.9 kB)
Requirement already satisfied: PyYAML in
/usr/local/lib/python3.10/dist-packages (from easyocr) (6.0.2)
Requirement already satisfied: Shapely in
/usr/local/lib/python3.10/dist-packages (from easyocr) (2.0.6)
Collecting pyclipper (from easyocr)
  Downloading pyclipper-1.3.0.post6-cp310-cp310-
manylinux_2_12_x86_64.manylinux2010_x86_64.whl.metadata (9.0 kB)
Collecting ninja (from easyocr)
  Downloading ninja-1.11.1.1-py2.py3-none-
manylinux1_x86_64.manylinux_2_5_x86_64.whl.metadata (5.3 kB)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (4.12.2)
Requirement already satisfied: sympy in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (1.13.3)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.4.1)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr)
(2024.6.1)
Requirement already satisfied: imageio>=2.33 in
/usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr)
(2.35.1)
Requirement already satisfied: tifffile>=2022.8.12 in
/usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr)
(2024.9.20)
Requirement already satisfied: packaging>=21 in
/usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr)
(24.1)
Requirement already satisfied: lazy-loader>=0.4 in
/usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr)
(0.4)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch->easyocr)
```

```

(3.0.2)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch->easyocr)
(1.3.0)
Downloading easyocr-1.7.2-py3-none-any.whl (2.9 MB)
  2.9/2.9 MB 21.0 MB/s eta
0:00:00
anylinux1_x86_64.manylinux_2_5_x86_64.whl (307 kB)
  307.2/307.2 kB 17.6 MB/s eta
0:00:00
anylinux_2_12_x86_64.manylinux2010_x86_64.whl (912 kB)
  912.2/912.2 kB 36.8 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (286 kB)
  286.8/286.8 kB 13.8 MB/s eta
0:00:00

# Install EasyOCR if not already installed
!pip install easyocr

# Import necessary libraries
import cv2
import os
import pandas as pd
import easyocr
import numpy as np

# Initialize EasyOCR reader for English
reader = easyocr.Reader(['en'])

# Function to recognize text from the license plate
def recognize_license_plate_text(license_plate_img):
    # Preprocess the image: convert to grayscale and apply
    # thresholding
    gray_img = cv2.cvtColor(license_plate_img, cv2.COLOR_BGR2GRAY)
    _, thresh_img = cv2.threshold(gray_img, 0, 255, cv2.THRESH_BINARY +
+ cv2.THRESH_OTSU)

    # Use EasyOCR to recognize text
    result = reader.readtext(thresh_img)

    # Extract recognized text
    recognized_text = ''.join([text[1] for text in result])
    return recognized_text.strip()

# Function to create one-hot encoding for the recognized digits (0-9)
def create_digit_one_hot_encoding(recognized_text):
    one_hot = [0] * 10  # Initialize with 10 zeros (for digits 0-9)

    for char in recognized_text:

```

```

    if char.isdigit(): # Check if the character is a digit
        one_hot[int(char)] = 1 # Set the respective index to 1

    return one_hot

# Function to process the test images in the specified folder and save
# the results
def process_test_images(img_folder, output_csv):
    results = []
    image_files = [f for f in os.listdir(img_folder) if
f.endswith('.jpg', '.png'))] # Adjust based on your image formats

    for i, img_file in enumerate(image_files):
        img_path = os.path.join(img_folder, img_file)
        img = cv2.imread(img_path)

        if img is None:
            print(f"Error: Image at path {img_path} could not be
loaded.")
            continue

        # Assuming the license plate is the whole image, if you have
        # bounding box logic apply it here
        license_plate_img = img # Change this if you have cropping
        logic

        # Recognize the text from the license plate
        recognized_text =
recognize_license_plate_text(license_plate_img)

        # Create one-hot encoding for the recognized digits
        one_hot_encoding =
create_digit_one_hot_encoding(recognized_text)

        # Prepare the row to append to the results (id + one-hot
encoding)
        img_id = f"img_{i+901}_{1}" # For example: img_901_1,
img_901_2, etc.
        result_row = [img_id] + one_hot_encoding
        results.append(result_row)

        # Create DataFrame and save as CSV
        columns = ['id'] + [str(i) for i in range(10)] # 'id', '0',
'1', ..., '9'
        results_df = pd.DataFrame(results, columns=columns)

        # Save to CSV
        results_df.to_csv(output_csv, index=False)
        print(f"Results saved to {output_csv}")

```

```
# Example usage
test_images_path = '/content/sample_data/test' # Adjust to your test
# images folder
output_csv_path = '/content/SUB.csv' # Adjust as needed

process_test_images(test_images_path, output_csv_path)

Requirement already satisfied: easyocr in
/usr/local/lib/python3.10/dist-packages (1.7.2)
Requirement already satisfied: torch in
/usr/local/lib/python3.10/dist-packages (from easyocr) (2.4.1+cu121)
Requirement already satisfied: torchvision>=0.5 in
/usr/local/lib/python3.10/dist-packages (from easyocr) (0.19.1+cu121)
Requirement already satisfied: opencv-python-headless in
/usr/local/lib/python3.10/dist-packages (from easyocr) (4.10.0.84)
Requirement already satisfied: scipy in
/usr/local/lib/python3.10/dist-packages (from easyocr) (1.13.1)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from easyocr) (1.26.4)
Requirement already satisfied: Pillow in
/usr/local/lib/python3.10/dist-packages (from easyocr) (10.4.0)
Requirement already satisfied: scikit-image in
/usr/local/lib/python3.10/dist-packages (from easyocr) (0.24.0)
Requirement already satisfied: python-bidi in
/usr/local/lib/python3.10/dist-packages (from easyocr) (0.6.3)
Requirement already satisfied: PyYAML in
/usr/local/lib/python3.10/dist-packages (from easyocr) (6.0.2)
Requirement already satisfied: Shapely in
/usr/local/lib/python3.10/dist-packages (from easyocr) (2.0.6)
Requirement already satisfied: pyclipper in
/usr/local/lib/python3.10/dist-packages (from easyocr) (1.3.0.post6)
Requirement already satisfied: ninja in
/usr/local/lib/python3.10/dist-packages (from easyocr) (1.11.1.1)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (4.12.2)
Requirement already satisfied: sympy in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (1.13.3)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.4.1)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch->easyocr)
(2024.6.1)
Requirement already satisfied: imageio>=2.33 in
/usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr)
(2.35.1)
Requirement already satisfied: tifffile>=2022.8.12 in
```

```
/usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr)
(2024.9.20)
Requirement already satisfied: packaging>=21 in
/usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr)
(24.1)
Requirement already satisfied: lazy-loader>=0.4 in
/usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr)
(0.4)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch->easyocr)
(3.0.2)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch->easyocr)
(1.3.0)

WARNING:easyocr.easyocr:Neither CUDA nor MPS are available -
defaulting to CPU. Note: This module is much faster with a GPU.
WARNING:easyocr.easyocr:Downloading detection model, please wait. This
may take several minutes depending upon your network connection.

Progress: |██████████| 100.0%
Complete

WARNING:easyocr.easyocr:Downloading recognition model, please wait.
This may take several minutes depending upon your network connection.

Progress: |██████████| 100.0%
CompleteResults saved to /content/SUB.csv
```