# Gate License detector

# Team Members:

**Khaled Mostafa**            **sec:1 BN:21**

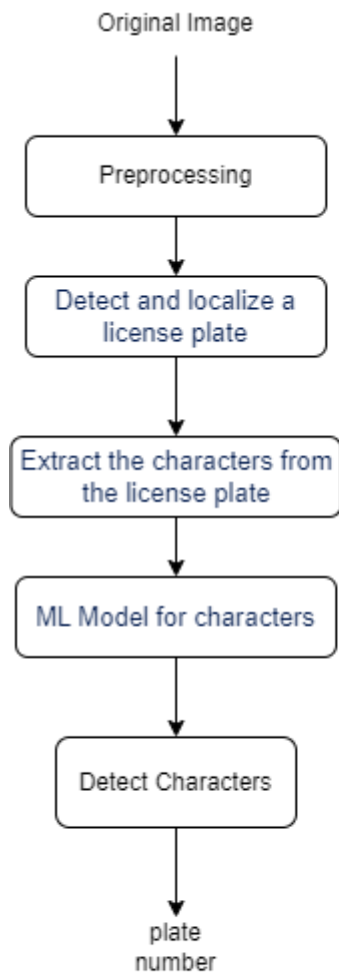**Abdel-Hamid Emad**          **sec:1 BN:34**

**Abdel-Rahman Ashraf**       **sec:1 BN:35**

**Karim Mohamed**            **sec:2 BN:8**

# Proposal:

Detect Plate number of the given car

# Block Diagram:



Original Image
↓
Preprocessing
↓
Detect and localize a license plate
↓
Extract the characters from the license plate
↓
ML Model for characters
↓
Detect Characters
↓
plate number



**PreProcessing**
1. remove noise from image
2. Thresholding
3. erosion
4. canny for edge detection

**Detect and localize a license plate**

Detecting the edges of the image

Find contours.

Finding the contour with four sides

Cropping the rectangular part identified as license plate

Drawing the selected contour on the original image

**Extract the characters from the license plate**

Extracting text from the image of the cropped license plate

(OCR)

to recognize the extracted characters

# Used Algorithms:

1. Histogram of gradient (HoG)
2. Support Vector Machine (SVM)
3. Check Area and ratio between the width and height to detect contour of plate

# Accuracy:

1. Accuracy of model:

```
Accuracy=   0.9998591549295774
Precision=   0.9998598962194217
Recall=   0.9998591549295774
F1 Score=   0.9998591781023665
```

| Classification Report= | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 217 |
| 1 | 1.00 | 1.00 | 1.00 | 203 |
| 2 | 1.00 | 1.00 | 1.00 | 206 |
| 3 | 1.00 | 1.00 | 1.00 | 216 |
| 4 | 1.00 | 1.00 | 1.00 | 208 |
| 5 | 1.00 | 1.00 | 1.00 | 198 |
| 6 | 1.00 | 1.00 | 1.00 | 208 |
| 7 | 1.00 | 1.00 | 1.00 | 202 |
| 8 | 1.00 | 1.00 | 1.00 | 222 |
| 9 | 1.00 | 1.00 | 1.00 | 207 |
| A | 1.00 | 1.00 | 1.00 | 212 |
| B | 1.00 | 1.00 | 1.00 | 201 |
| C | 1.00 | 1.00 | 1.00 | 219 |
| D | 1.00 | 1.00 | 1.00 | 199 |
| E | 1.00 | 1.00 | 1.00 | 191 |
| F | 1.00 | 1.00 | 1.00 | 209 |
| G | 1.00 | 1.00 | 1.00 | 210 |
| H | 1.00 | 1.00 | 1.00 | 199 |
| I | 1.00 | 1.00 | 1.00 | 207 |
| J | 1.00 | 1.00 | 1.00 | 191 |
| K | 1.00 | 1.00 | 1.00 | 186 |
| L | 1.00 | 1.00 | 1.00 | 211 |
| M | 1.00 | 1.00 | 1.00 | 201 |
| N | 1.00 | 1.00 | 1.00 | 197 |
| P | 1.00 | 1.00 | 1.00 | 200 |
| Q | 0.99 | 1.00 | 1.00 | 189 |
| R | 1.00 | 1.00 | 1.00 | 192 |
| S | 1.00 | 1.00 | 1.00 | 209 |
| T | 1.00 | 1.00 | 1.00 | 191 |
| U | 1.00 | 1.00 | 1.00 | 194 |
| V | 1.00 | 1.00 | 1.00 | 210 |
| W | 1.00 | 1.00 | 1.00 | 202 |
| X | 1.00 | 1.00 | 1.00 | 204 |
| Y | 1.00 | 1.00 | 1.00 | 212 |
| Z | 1.00 | 1.00 | 1.00 | 177 |

2. Accuracy of Project:
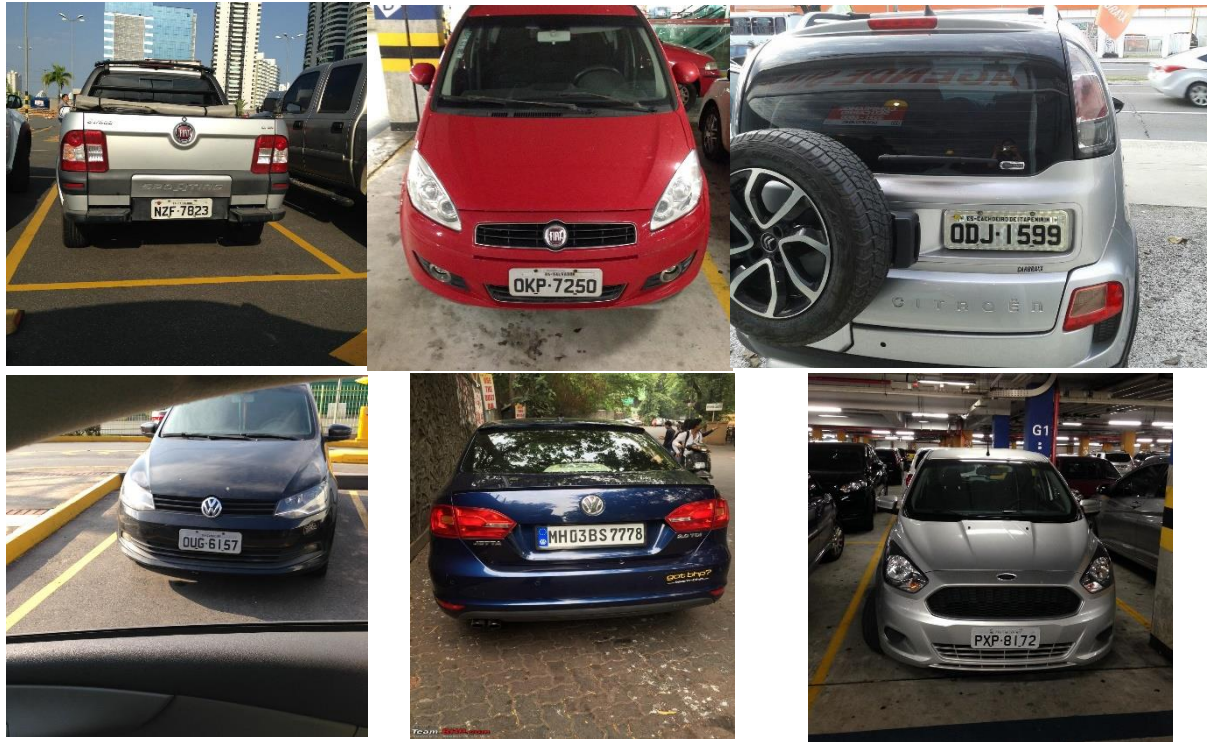
```
    print(getAccuracy(True_license,predict_licence))
 ✓  0.4s
```

89.7590361445783

```
    print(getFullAccuracy(True_license,predict_licence))
 ✓  0.3s
```

56.52173913043478

# Workload:

All participate in the first module which detect contours and separate characters

All participate in the second which detect the characters (Machine learning)

# Cars Variety:



# Letters Variety:



# Analysis:

## Points of Strengths:

1. Can detect different orientation of car plate
2. Can detect different orientation of characters
3. Can detect car plate with various illumination

## Points of Weakness:

1. White cars
2. Some letters like W & V


## References:

OpenCV: OpenCV-Python Tutorials

sklearn.svm.SVC — scikit-learn 1.2.0 documentation