DIGITAL IMAGE PROCESSING - FINAL PROJECT

Vehicle number plate detection and number recognition using YOLOv3 and CNN

Presented By
Monalisa Bakshi -MTech SP - 21697
Nikhil Bisen -MTech AI - 21698

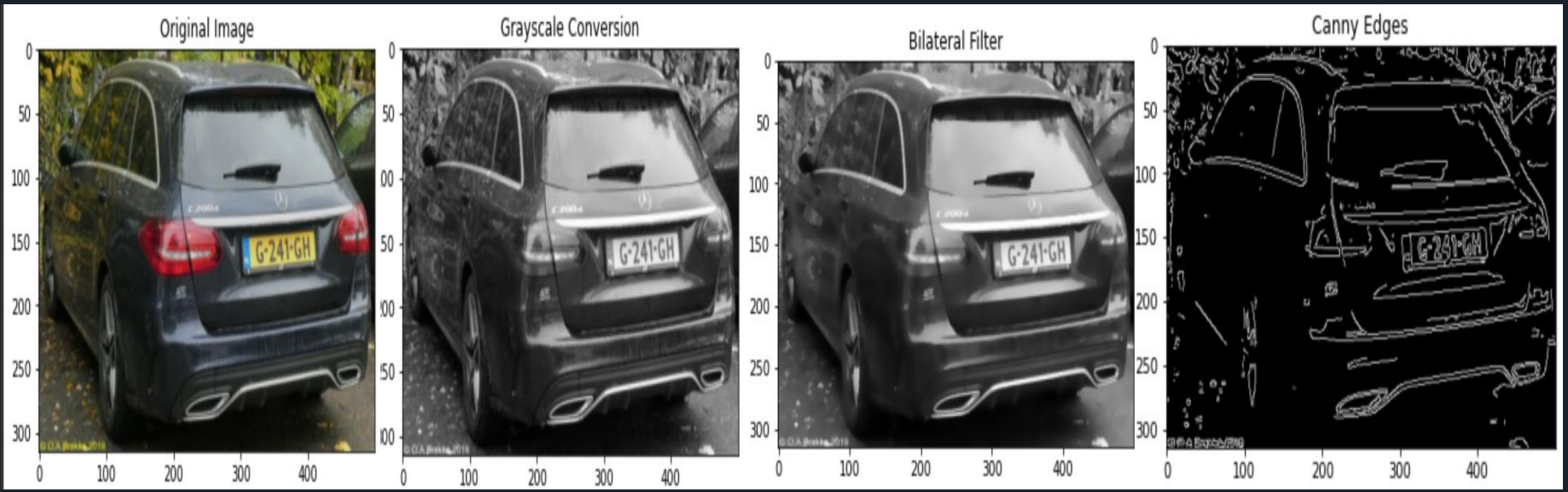
Introduction and Project Flow

Vehicle Number Plate Detector can detect the license number of Car from the images obtained from surveillance camera . It can be used at tolls for recording the license number and finding the stolen cars etc . Below is the flow of project which contain 3 major steps :

01	Detecting and Extracting Number Plate from overall Image	• We have done this with 2 different method • First by Finding Contours that resembles the number plate • Second by You Only Look Once(YOLO) algorithm .
02	Separating the Individual Character .	• Individual Character needs to be separated and segregated. • We will use character segmentation technique like Finding rectangular Contours.
03	Recognizing the Separated Characters .	• Segmented Characters will be recognized using deep learning classifier. • We will make use off 4 layer -Convolutional Neural Network to do this .

Object detection using finding contours(Preprocessing)

Input image → Gray scale image → Noise elimination → Canny edge detection

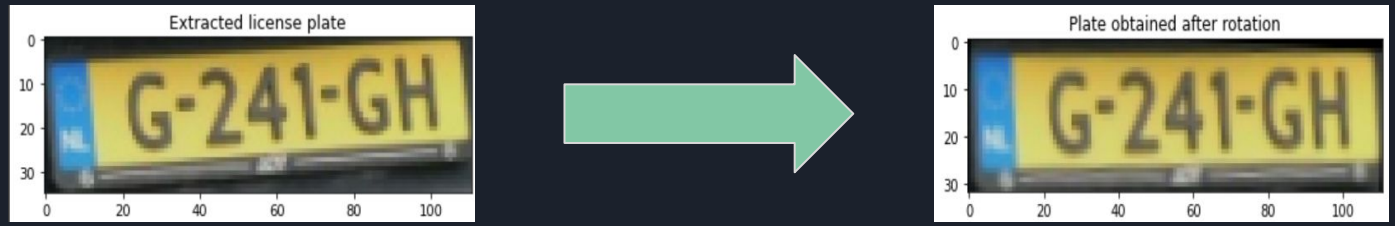


- Each contours are approximated to form a Polygon .
- Contours with favourable condition are detected using cv2 library

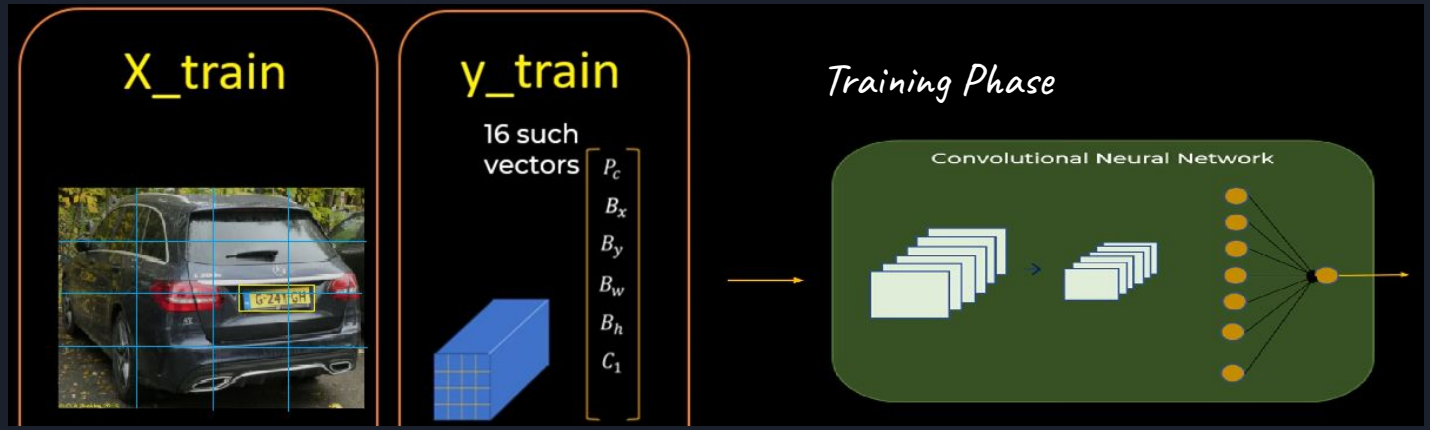
Canditions:

1. Area>30
2. Quadrilateral in shape

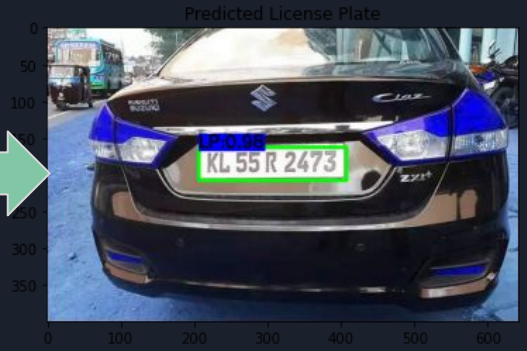
- Rotation(if needed)



Object Detection Using YOLOv3

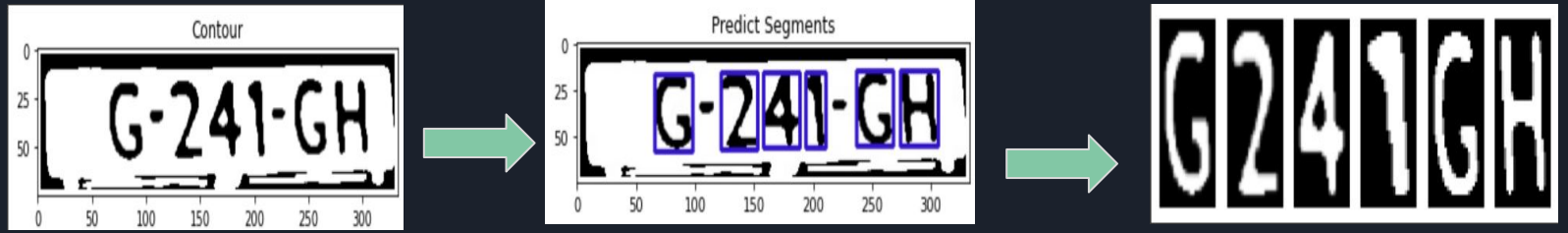


- We will use Pre-Trained Model with YOLOv3 weights for vehicle plate detection . We have got this from Kaggle .
- Final Result :



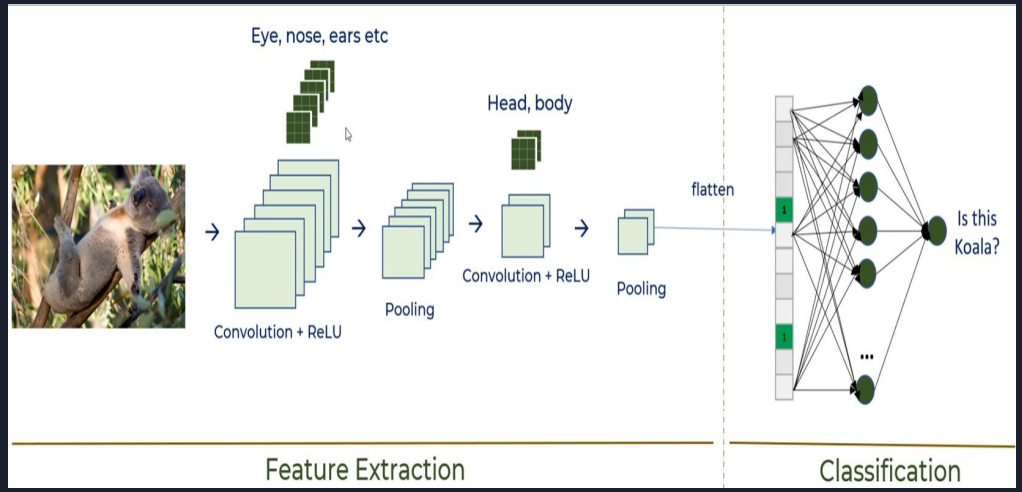
Character Segmentation

- Pre Processing - Resizing, Grayscale conversion. Binarization, Eroding, Dilate
- Contour detection
- Finding bounding rectangle of each character by dimension comparison.



Character Recognition using CNN

Working of CNN



CNN Model Summary that we use for Character Recognition:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 16)        23248
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 32)        131104
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 28, 28, 64)        131136
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 64)        65600
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 7, 7, 64)          0
_________________________________________________________________
dropout (Dropout)            (None, 7, 7, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 3136)              0
_________________________________________________________________
dense (Dense)                (None, 128)               401536
_________________________________________________________________
dense_1 (Dense)              (None, 36)                4644
=================================================================
Total params: 757,268
Trainable params: 757,268
Non-trainable params: 0
_________________________________________________________________
```

Hyper Parameter Tuning :

For dropout rate = 0.2

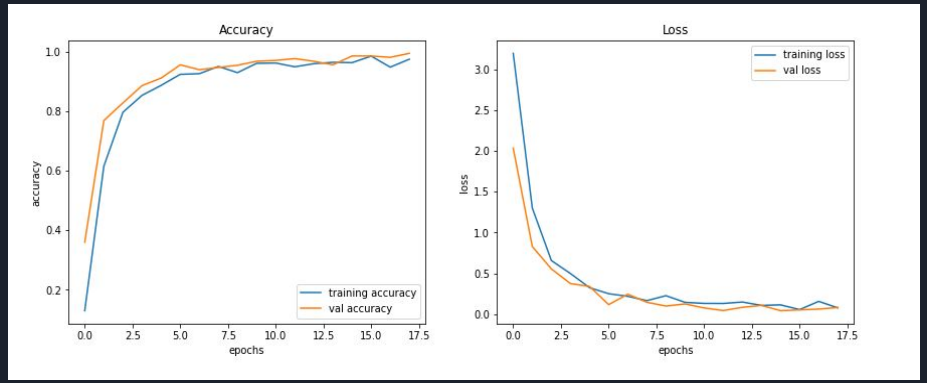
	adam	sgd	rmsprop
Lr=0.001	0.8925	0.9743	0.9285
lr=0.0005	0.9506	0.9754	0.9598
Lr=0.0001	0.9726	0.8321	0.9794

For dropout rate = 0.4

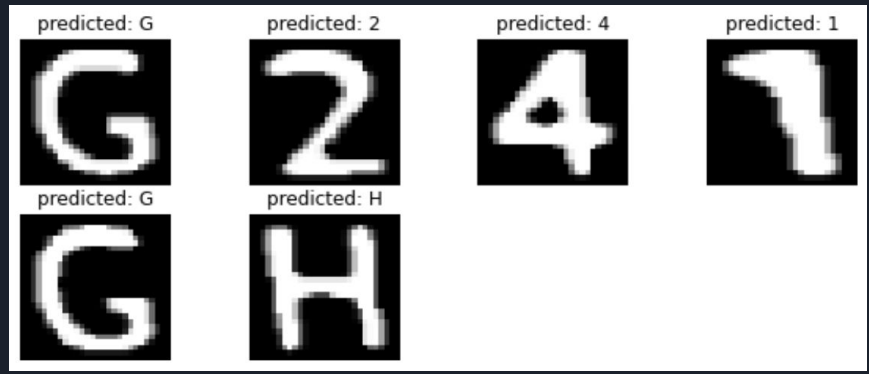
	adam	sgd	rmsprop
Lr=0.001	0.2944	0.9656	0.8888
lr=0.0005	0.9441	0.9479	0.9322
Lr=0.0001	0.9812	0.6574	0.9618

Loss function : sparse_categorical_crossentropy and optimal dropout rate = 0.4 , optimal learning rate = 0.0001 , and optimizer = adam

Training Accuracy and Loss vs Epochs :



Predicting the Characters :-



Conclusions:

1) Accuracy for Number Plate Detection using Finding Contour comes out to be around 63 % .
2) Accuracy for Number Plate Detection using YOLOv3 comes out to be around 72 % .
3) Contour Method may return any rectangular objects found in an image , hence its accuracy is less compared to YOLO v3
4) It's important to do preprocessing before every step in order to get better results .
5) Hyper parameter tuning in CNN helps us to get the optimal parameter and thus increase model efficiency .

References

1)Pre-Trained CNN model with YOLOv3 weights for license plate detection(from Kaggle)https://www.kaggle.com/datasets/achrafkhazri/yolo-weights-for-licence-plate-detector

2)Setiyono, Budi & Amini, Dyah & Sulistyaningrum, Dwi. (2021). Number plate recognition on vehicle using YOLO - Darknet. Journal of Physics: Conference Series. 1821. 012049. 10.1088/1742-6596/1821/1/012049.

3)Shrutika Saunshi, Vishal Sahani, Juhi Patil, Abhishek Yadav, Dr. Sheetal Rathi, "License Plate Recognition Using Convolutional Neural Network" in IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661,p-ISSN: 2278-8727, PP 28-33

THANK YOU