VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

IMAGE PROCESSING
CSE4019

AUTOMATIC NUMBER PLATE RECOGNIZATION SYSTEM

PROJECT COMPONENT
REVIEW 3

UNDER THE GUIDENCE OF
PROF. SWATI J.N.

BY DIVYANAND 18BCE0902
PARTH MEHRA 18BCE0504

VIDEO LINK:

https://drive.google.com/drive/u/0/folders/139wC2fDbTgRNOhCqR8A_NK_XNsbVt0XO

GITHUB LINK:

https://github.com/gouravdeepu/automatic-number-plate

ABSTRACT

Recognizing vehicle’s license plate is necessary because the number of vehicles is increasing and it goes beyond human’s ability to complete this task. Automatic Number Plate Recognition is a mass surveillance method that performs optical character recognition on images to read the license plates on vehicles. It is an image-processing technology used to identify vehicles by their license plates.

License Plate Recognition is a computer system that recognizes any digital image automatically on the number plate. This system includes various operations such as taking pictures, localizing the number pad, truncating characters and OCR (Optical Character Recognition) from alphanumeric characters. The main idea of this system is to design and develop effective image processing techniques and algorithms to localize the license plate in the captured image, to divide the characters from that number plate and to identify each character of the segment by using the Open Computer Vision Library. Many applications can be implemented by using this system, such as security, highway speed detection, violation of light, identification of handwritten text, discovery of stolen cars, automatic fee collection systems.

The aim of this project is to design and analyse the License Plate Identification program mediated through Digital Images or Automatic Number Plate Recognition (ANPR), especially by using desktop peripheral. In doing so, license plates attached, especially, on cars will be the test subject of this research. It will be able to necessarily recognize or identify the license plates installed on the vehicles by using their digital images.

The fundamental goal of the ANPR program itself is actually to utilize digital image identification system in order to identify every single vehicle. From the result of our experimentation, the ANPR is able to detect and translate the license plates into a form of text in a very minimum time. Series of analysis that the ANPR program situates involves; analysis on the ratio of the license plate, experimentation on the distance of license plate detection process, and overall system examination. The result retrieved from the conducted analysis can be considered as a part of the ANPR system.

INTRODUCTION

People from different countries interact in a multicultural environment to develop solutions to never-ending problems for men. The Open Source section is a one of the outstanding contribution in the scientific world is Python. Computer vision in the Intel's research has been producing a fruit called Open Computer Vision (Open CV), which can support the development of computer vision.

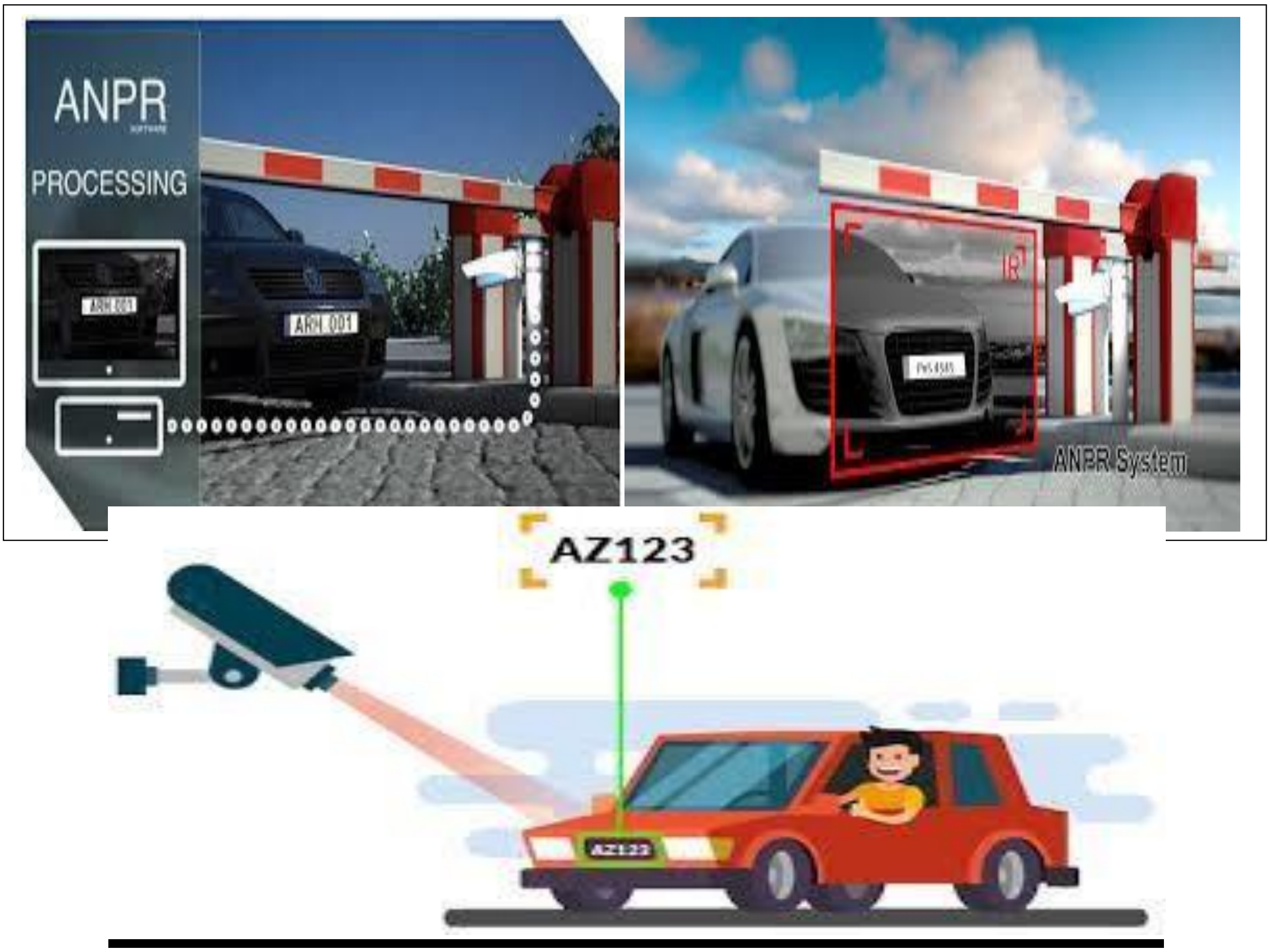
At present, the use of vehicles is increasing throughout the country. All of these vehicles have a unique vehicle identification number as their main identifier. The ID is actually in the license number that refers to a legal license to participate in the public movement. Each vehicle in the world must have its own number plate that must be installed on its body (at least on the back). They need to identify the vehicles are increasing in parallel with the number of vehicles. This identification system helps with safety, automatic switching systems, highway speed detection, light detection, stolen vehicle detection, and human and non-human loss collection systems. The auto license plate recognizing system replaces the manual license plate number writing process in the computer system.

In order to obtain an appropriate personal recognition, the license plate identification technique consists of three main topics. They are, find the location of the panel of digital images, segmentation the characters from the pictures of the panel and the visual character recognition. The most dominant and basic step is to determine the exact location of the number plate in the captured image. The localization of a license plate has been recognized either by structural analysis and colour analysis method. In the License panel area, unwanted spots are removed by parsing the connected component.

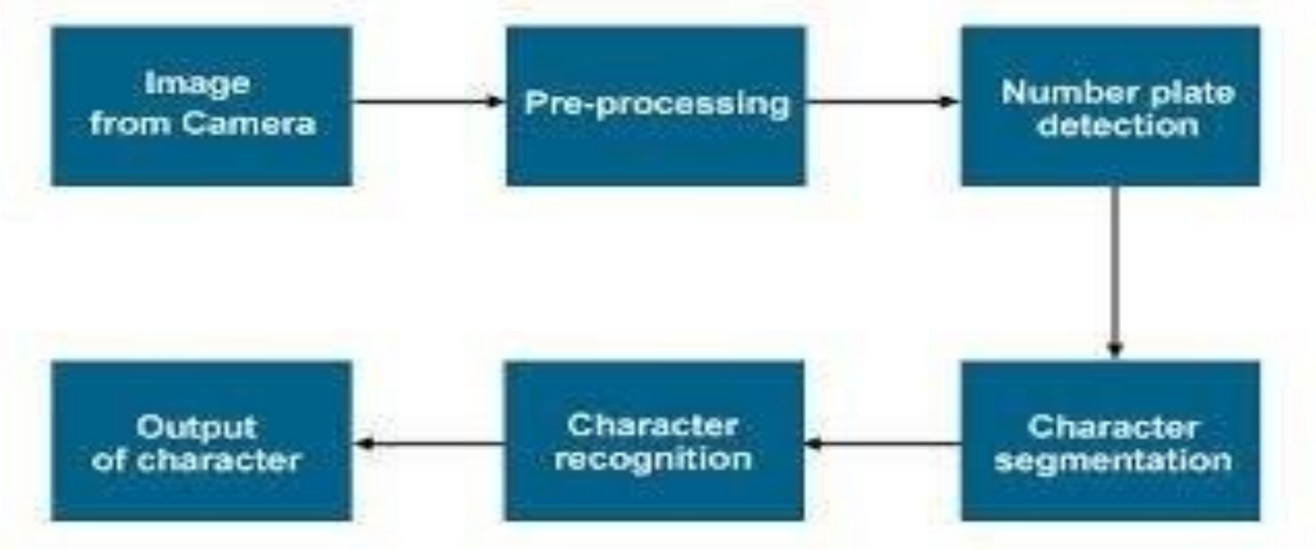
ANPR is a collective control system that captures the vehicle image and identifies the license number. Some ANPR system applications are automatic traffic control and tracking system, highway toll collection / automatic parking systems, petrol station automation, flight time monitoring. These systems automate the process of identifying vehicle license number, making it fast, cost effective.

It is a technology that uses optical character recognition on images to read vehicle registration plates to create vehicle location data. It can use existing closed-circuit television, road-rule enforcement cameras, or cameras specifically designed for the task. It is deterrent for serial traffic offenders. The software helps with reactive security as well, which includes inspections, forensics, investigations, and legal proceedings. It seems like a necessity for ensuring our safety on the road. It could also be useful when it comes to detecting stolen vehicles. LPR system may be installed as a part of traffic monitoring systems working together with traffic light in order to identify the car that break traffic rules or detect prohibited devices.In many cases, LPR system is useful for recognizing vehicles with harmful behaviours, for example, riding against the traffic direction, riding over speed limit, and not wearing helmet.

Nowadays, road safety enforcement is getting high attention as an important issue in order to reduce the possibility of accidence from irresponsible drivers. So, we decided to make a project by using deep learning approach to recognize license plates. ANPR tends to be extremely challenging subfield of computer vision, due to the vast diversity and assortment of license plate types across states and countries.



Steps involved in this project will be -:



PROBLEM STATEMENT

- Recognizing vehicle’s license plate is necessary for various purpose for security , rule violation , recognization of lost or stolen vehicle etc etc but as the number of vehicles is increasing it goes beyond human’s ability to complete this task.

BASE PAPER IDENTIFICATION

https://ieeexplore.ieee.org/document/9027882 (Base Paper)
https://ieeexplore.ieee.org/document/9088975
https://ieeexplore.ieee.org/document/9092977

LITERATURE SURVEY

Searching for license plate recognition is still a challenge. It involves three major steps. They specify number pad space, character segmentation, and character recognition. Each step suggested different ways to improve efficiency.

One of these methods used the adaptive threshold to highlight the characters and suppress the background. In order to remove unwanted image spaces, a component algorithm is first applied to the converted binary image from the original panel. A special algorithm called Image Scissoring is used to divide the Optical Character Recognition engine called tesseract, which returns ASCII to the license number. The entire system has been implemented using open CV.

Another method is to deploy the forward background feed method for character classification. The neural network is developed by using the backward- propagation algorithm. Normalization, scale and edge detection are included in the steps of the pre-processing. The horizontal and vertical graph and component survey are able to address the problem of character fragmentation.

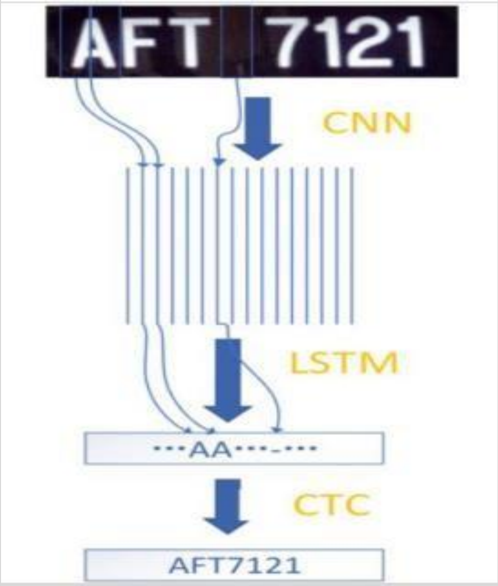
Another way in which character areas are selected is through binarization, connected component analysis. The Point Analysis method removes unwanted points and combines split points and split points. This unit achieves a 97.2% accuracy rate in character segmentation. The reliability of the recognition was 90.9%.

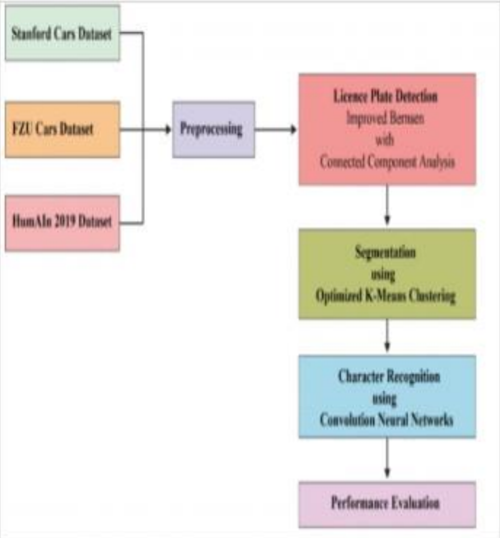
Offers an approach that relies on effective morphological operation and the detection method of Sobel Edge. This approach is simplified to divide all letters and numbers used in the number pad using the surround box method. After the template is fragmented, the matching policy is used to recognize numbers and characters. This whole system was implemented using MATLAB. Provides an overview of the analysis of related components and processes, such as aspect ratio analysis and pixel count analysis.

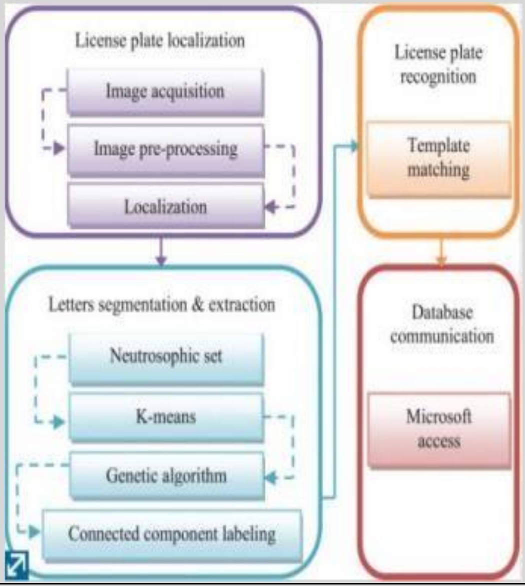
In the author studies a comparison of four algorithms that are sequentially using statistical properties, the Hough Transform and Contour algorithm, the medium transformation approach and morphological processes and their results.

The handwritten text is fragmented by the watershed algorithm. Noise removal, slope correction, budgeting and normalization were eliminated in pre-treatment. After fragmentation the process of extracting a segmented image is done by a reverse integer to convert the wavelet integer. The classification is then sorted by neuroscience.

Some other research papers that we have studied are:

Title	Methodology	Conclusion	Future Work
CNN-RNN based method for license plate recognition	In this, for the separation of foreground and background information, a Canny edge image is considered in which the edge pixels are the foreground information, and the rest pixels are background pixels. It applies K-means clustering with K=2 for this separation for classifying the private and public license plates. For the classified images, a combination of CNN and RNN, which involves BLSTM(bi-directional long-short term memory), for recognition is proposed. Three steps involved: CNN for feature extraction, LSTM for extracting sequence and connectionist temporal classification (CTC) for the final recognition. 	The proposed classification works well for images with blur, low contrast and illumination effects but fails for too low resolution and poor quality. This method is better in terms of robustness. The proposed method before and after show that classification is useful for achieving high recognition rate for a complex dataset rather than developing new method.	There is a scope for improvement and extension as this method does not classify images with very low resolution and poor quality. Also, when the images are oriented in different directions, the proposed \ method does not work well. It is a limitation of CTC.

Automatic Vehicle License Plate Recognition Using Optimal K-Means With Convolutional Neural Network for Intelligent Transportation Systems	The proposed model is OKM-CNN Model whose working principle is as shown:  Stanford Cars Dataset FZU Cars Dataset HumAIn 2019 Dataset Preprocessing Licence Plate Detection Improved Bernsen with Connected Component Analysis Segmentation using Optimized K-Means Clustering Character Recognition using Convolution Neural Networks Performance Evaluation **Framework of OKM-CNN method** At first, LP localization and recognition process takes place utilizing IBA and CCA model. Two Binary methodologies: The function of Otsu is a contingent on illumination constraints that has drastic variation. In order to overcome the irregular illumination barrier, especially for shadow images, a new binary technique such as the enhanced BA was utilized in this research. After this, the characters in the LP get segmented using OKM algorithm in which K-means clustering with KH algorithm is incorporated. At last, the CNN-based character recognition process takes place to recognize the characters present in LP.	It is implied that the OKM-CNN approach analysed the LP number on images. The presented OKMCNN model achieved the best recognition performance. It can be employed as the major element of intelligent infrastructure like toll free collection, parking management and traffic surveillance.	In future, the performance of the OKMCNN model can be enhanced to recognize multilingual LPs. Furthermore, the experimental outcome of OKM-CNN model can be improved by the inclusion of bio-inspired optimisation algorithm based parameter tuning process.
Toward an Optimized Neutrosophic K-Means with Genetic Algorithm for automatic Vehicle License Plate	In this, we consider both Arabic-Egyptian license plates and English license plates. 4 stages: 1 Detection 2 Segmentation 3 Recognition 4 Database Communication.	The proposed system offers a successful detection with an accurate recognition in both Arabic and English license plates. The proposed methodology offers a high rate of LP	The extension aims to implement the neutrosophic set according to more optimization techniques like particle

Recognition (ONKMAVLPR)	In Ist stage, detect the location of (LP) using means of edge detection & morphological operations. In 2nd stage, use optimized neutrosophic set (NS) algorithm for extracting the most salient features in (LP) images. This optimization has been accomplished using genetic algorithm. It aims to reduce indeterminacy in (LP) images. K-means algorithm has been utilized for clustering purposes, and the last step in previous stage, connected component labelling analysis (CCLA) has been applied in order to extracting characters individually. In the third stage, characters would be recognized according to the measurement of characters matching with the templates that stored in the database. Finally, store the recognized (LP) in Microsoft access database.  License plate localization Image acquisition Image pre-processing Localization Letters segmentation & extraction Neutrosophic set K-means Genetic algorithm Connected component labeling License plate recognition Template matching Database communication Microsoft access	recognition accuracy in the presence of some popular image degradation.	swarm, fuzzy tech., etc. Also, more image disruption & variation can be included to have a wide decision making criteria for the best optimizer.
Research on License Plate Recognition Algorithm Based on Deep Learning in Complex Environment.	It discusses the application of deeplearning in license plate recognition 1. Three main technical difficulties: License plate skew: Various vertical and horizontal tilt in the images result in distortion and affect the character recognition. Image noise: Sources of noise are image acquisition, transmission and compression, etc.	It presents a survey on existing license plate systems based on deep learning algorithms, and categorize the algorithms at each stage by the process. And the different license plate recognition systems	It should be concentrated on solving the three aspects of complex scenes, namely, license plate correction, denoising and high

	License plate blur: It will be time consuming to detect small target by learning its representation on multiple scales. 2. According to the process, the deep learning algorithms are classified into: Direct detection algorithms: Directly predict the location, height and width information of the license plate by feeding the image. Loss function such as Euclidean distance should be calculated for parameter gradient. Indirect detection algorithms: When the target is too small or partial shielded then it can be used. 3. License plate recognition systems are compared: Segmentation Based: Five categories: connected component analysis, projection analysis, prior knowledge of characters, character contours and their combination. Segmentation Free Based: Transform the license plate recognition problem into character sequence labelling. It utilize global information of input image.	based on deep learning are compared in term of models, datasets, precision and processing time. Some public available license plate datasets are sorted out to compare the amount of each dataset & image resolution, and explain the situation in terms of shooting angle, illumination conditions & other background complexity.	resolution representation, as well as the diversified evaluation system and the construction of a unified model to be end-to-end trained and tested.
Anonymous Vehicle Detection for Secure Campuses: A Framework for License Plate Recognition using Deep Learning	This system used 3 main steps to develop the automatic plate recognition: a) Training using Faster R-CNN: By using R-CNN they were able to work on a nearly cost-free region proposal that uses full-image Convolution features. They have used TensorFlow Object detection API to train Faster R-CNN pretrained model. b) Image processing: The number plate detected in the new image undergoes image processing so that the characters are identified properly. Gaussian smoothing filter and median Blur filter is used to blur the image and remove noise. This finally gives the image that can be used for character segmentation and recognition. c) Tesseract OCR: For extracting the number, they have used tesseract, an OCR engine.	This works with change in light intensity of surroundings on number plates. This framework can further be used for smart traffic: identifying the number plates of vehicles breaking traffic rules or for optimized parking: keeping an account of vehicles in parking.	There is a scope for improvement and extension as this method does not classify images with very low resolution and poor quality.

Indian Car Number Plate Recognition using Deep Learning	In this paper single YOLO model is used for both number plate detection and recognition. They have taken an ANPR data set of over 6500 real-world car number plate images, out of which 90% images are used for training purpose and the remaining 10% for testing. The number of filters used in this model is 126. An input image is fed into the YOLO model, then if the number plate is detected, the corresponding number plate region of interest is extracted and this image is again fed into the YOLO model to get recognized. Finally, the recognized output is sorted from left to right so that it's in the correct order as in the number plate.	This system achieve a 100 % accuracy in Number Plate Detection and 91% accuracy in number plate recognition. This presents a system, which is based on the cutting-edge object detection algorithm YOLOv3	Can't detect number plates having character "0" or "O" as they both look similar.
An Embedded Automatic License Plate Recognition System using Deep Learning	This paper proposes an embedded solution to detect and recognize Brazilian license plates using convolutional neural networks (CNN). The proposed system consists of two major phases. Firstly, the detection phase is performed receiving as input the entire image frame. The detection operation returns the detected plates as bounding boxes coordinates. The Recognition feature takes the segmented plates and performs character recognition of the whole sequence. The YOLO algorithm optimizes the processing using an image grid, which requires a single forward pass, instead of a sliding window strategy. So, in this work, we combined the YOLO module with the recognition phase. This strategy has the advantage of optimizing the computation while it only passes the small segmented plates to the recognition phase.	The proposed system has demonstrated to be robust to angle, lightning and noise variations. The system was validated using real license plate images under different environmental conditions reached a detection rate of 99.37% and an overall recognition rate of 97.00% while showing an average time of 2.70 seconds.	The proposed system only works for private vehicle license plates. The inclusion of motorcycle and special license plates, which have a different format
An efficient plate recognition system using convolution neural networks	This proposed system first recognises the vehicle using deep learning techniques and then retrieves license plate from the detected vehicle and at last used CNN to improve character recognition of blurred images. The prosed system consists of four major phases: Vehicle Detection, License Plate Localization, Character Segmentation and Character Recognition. They use coco 2017 dataset to train a YOLOv2 model.	Reduces the chances of false detection, provides 99.2% of accuracy and superiority of the proposed license plate recognition system in both accuracy and performance.	Takes time to give the output as numbers of phases are increased.

METHODOLOGY

Tools Used:

1. **Python-tesseract:** Python-tesseract is an optical character recognition tool for python. That is, it'll recognize and "read" the text embedded in images. Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. It is used as an individual script, because it can read all image types like jpeg, png, gif, bmp, tiff, etc. Additionally, if used as a script, Python-tesseract will print the recognized text rather than writing it to a file. It has ability to recognize more than 100 languages.

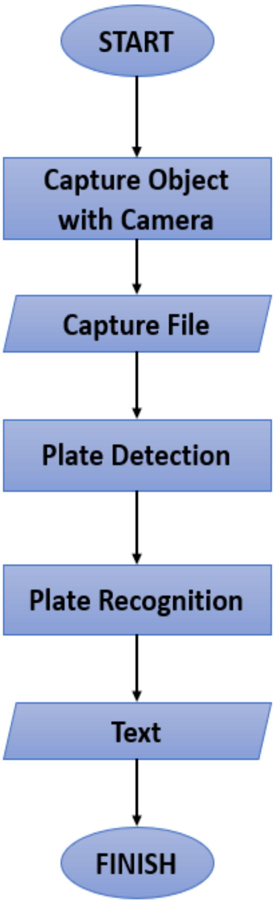
2. **OpenCV:** Open Source Computer Vision Library is a common platform and set of programming functions for real-time applications. The open CV library contains several algorithms for more than 2500 optimize algorithms. Used mostly around the world, with forty thousand people in the user group. Open CV is a multiplatform library, containing C ++, Python, and Java interfaces. Open CV is designed to achieve computational efficiency with a strong focus on real-time applications. These algorithms are often used to search and recognize faces, identify objects, recognize scenery and generate markers to overlay images using augmented reality, etc.

EXISTING METHOD:

In many countries ANPR methods have been implemented such as Australia, Korea and a few other countries. In the development of ANPR system in many countries the number plate standards are strictly implemented. These systems use standard features for license plates such as: panel dimensions, panel borders, colour and letter characters, etc., which help to easily localize the number pad and specify the car license number.

In India, plate number standards are rarely followed. There are wide variations in font types, text, size, position, and colours of number plate. In a few cases, there are other undesirable decorations on the number panel. Also, different other countries, there are no special features on Indian number panel to facilitate recognition. Thus, only manual recording systems are currently being used and ANPR has not been commercially developed in India.

PROPOSED METHOD:



Flowchart of the ANPR System

The vehicle license recognition system commonly combines 2 sub-systems:

1. **License Plate Detection:** It aims to locate the vehicle and its license plate.
2. **License Plate Recognition**: It aims to recognize the characters on the plate.

This process involving following steps:

Step 1: Detect and localize a license plate in an input image/frame.
Step 2: Extract the characters from the license plate.
Step 3: Apply some form of Optical Character Recognition (OCR) to recognize the extracted characters.

ALGORITHM PROCEDURE:

1. Begin
2. Input: Original Image
3. Output: Characters
4. LP: License Plate
5. Convert RGB to Grayscale
6. Find the edges of the image
7. Find all the contours in the image
8. Find top 30 contours in the image
9. Find the 4 corner contour in the image which represents the number plate
10. Crop that license plate contour
11. Recognize the characters using Python-tesseract
12. Print the recognized characters

STEPS INVOLVED:

1. Capture the Input Image:

The car's number pad is taken from a high resolution camera. The resolution of the number plate recognition system depends on the captured image. A better choice is an Infrared (IR) camera. The camera may be rolled and pitched with respect to the license plates. Character recognition is generally very sensitive to the skew. The readable characters can become distorted due to the obliqueness of the camera. Using a better camera with more definition and resolution will increase the success ratio of the system.

Captured Image

2. **Pre-Processing:**

Pre-processing is a set of algorithms applied to the image to improve the quality. It is an important and common phase in any computer vision system.

For the present system pre-processing involves two processes:
Resize – The image size from the camera might be large and can drive the system slow. It is to be resized to a feasible aspect ratio.
Convert Colour Space – Images captured using IR or photographic cameras will be either in raw format or encoded into some multimedia standards. Normally, these images will be in RGB mode, with three channels (viz. red, green and blue).

Number of channels defines the amount colour information available on the image. The image has to be converted to grayscale.

3. **Edge-Detection:**

Edge detection is an image processing technique for finding the boundaries of the objects within the images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction.

4. **Contour Detection:**

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same colour or intensity. The contours are a useful tool for shape analysis and object detection and recognition. In this project, we will be highlighting all the contours first, and then we will find top 30 contours among them and highlight them. After that, By applying some functions, we will find the contour having 4 edges (like a number plate) and highlight that.

5. **Segmentation:**

Segmentation is the process of cropping out the number plate contour from the whole image.

6. **Character Recognition:**

Finally, cropped image is send to a Optical Character Recognition (OCR) Engine, which returns the recognised characters of the license plate.

IMPLEMENTATION

OpenCV with Python is a deadly combination for computer vision. Here, Three packages have been imported for this. First one is cv2, which is OpenCV package. Next is imutils package, which will be used to resize the image. And last one is python-tesseract package which will convert image into the string.

Importing files

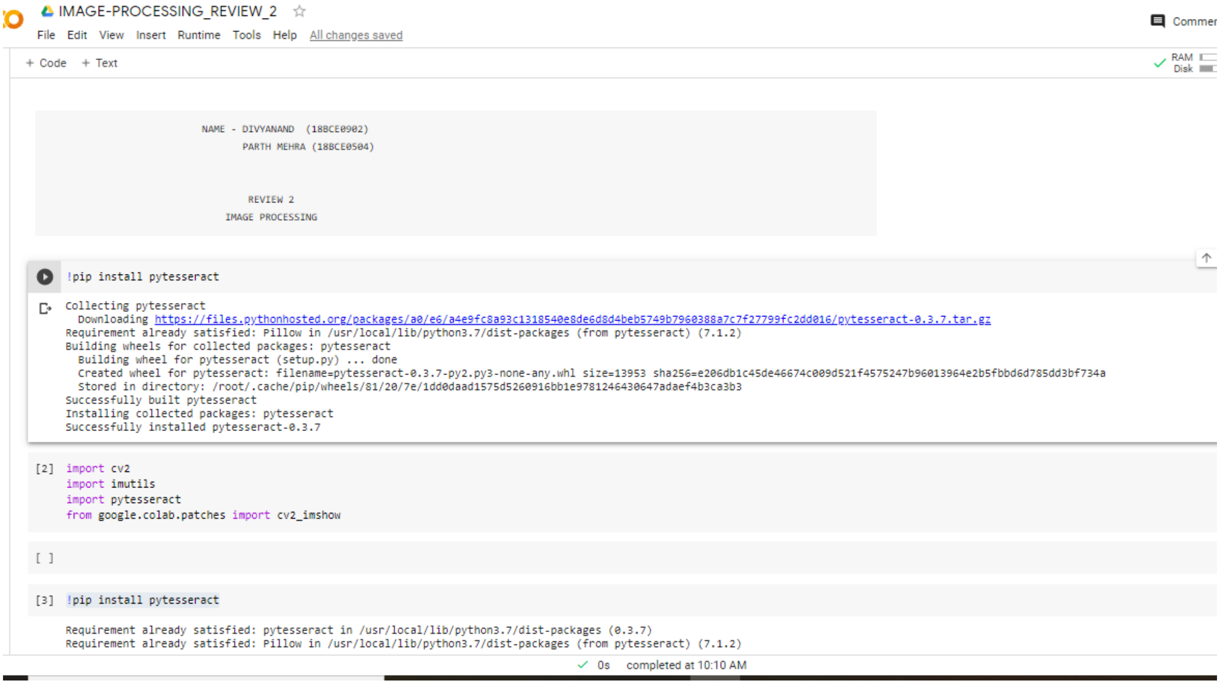


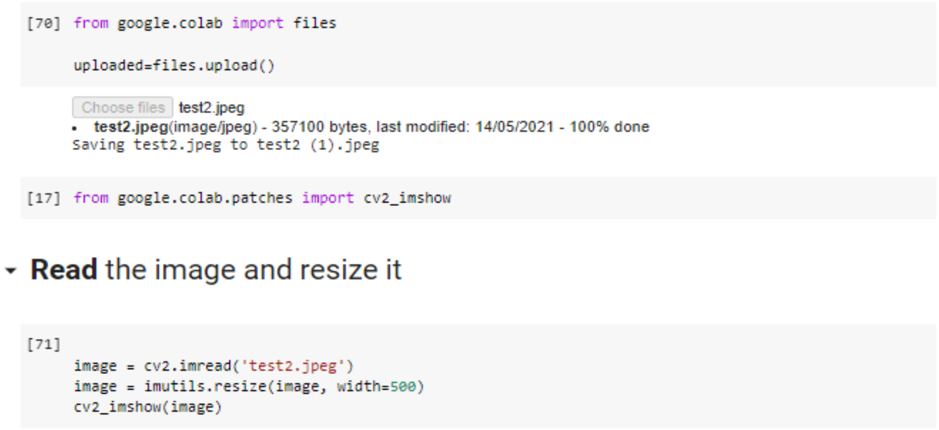
IMAGE-PROCESSING_REVIEW_2
File Edit View Insert Runtime Tools Help All changes saved
Commer
+ Code + Text
RAM
Disk
✓ 0s completed at 10:10 AM

```
NAME - DIVYANAND  (18BCE0902)
       PARTH MEHRA (18BCE0504)


        REVIEW 2
    IMAGE PROCESSING
```

```
!pip install pytesseract

Collecting pytesseract
  Downloading https://files.pythonhosted.org/packages/a0/e6/a4e9fc8a93c1318540e8de6d8d4beb5749b7960388a7c7f27799fc2dd016/pytesseract-0.3.7.tar.gz
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from pytesseract) (7.1.2)
Building wheels for collected packages: pytesseract
  Building wheel for pytesseract (setup.py) ... done
  Created wheel for pytesseract: filename=pytesseract-0.3.7-py2.py3-none-any.whl size=13953 sha256=e206db1c45de46674c009d521f4575247b96013964e2b5fbbd6d785dd3bf734a
  Stored in directory: /root/.cache/pip/wheels/81/20/7e/1dd0daad1575d5260916bb1e9781246430647adaef4b3ca3b3
Successfully built pytesseract
Installing collected packages: pytesseract
Successfully installed pytesseract-0.3.7
```

```
[2] import cv2
    import imutils
    import pytesseract
    from google.colab.patches import cv2_imshow

[ ]

[3] !pip install pytesseract

    Requirement already satisfied: pytesseract in /usr/local/lib/python3.7/dist-packages (0.3.7)
    Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from pytesseract) (7.1.2)
```

Read the image and resize it



```
[70] from google.colab import files

     uploaded=files.upload()
```

test2.jpeg

- **test2.jpeg**(image/jpeg) - 357100 bytes, last modified: 14/05/2021 - 100% done

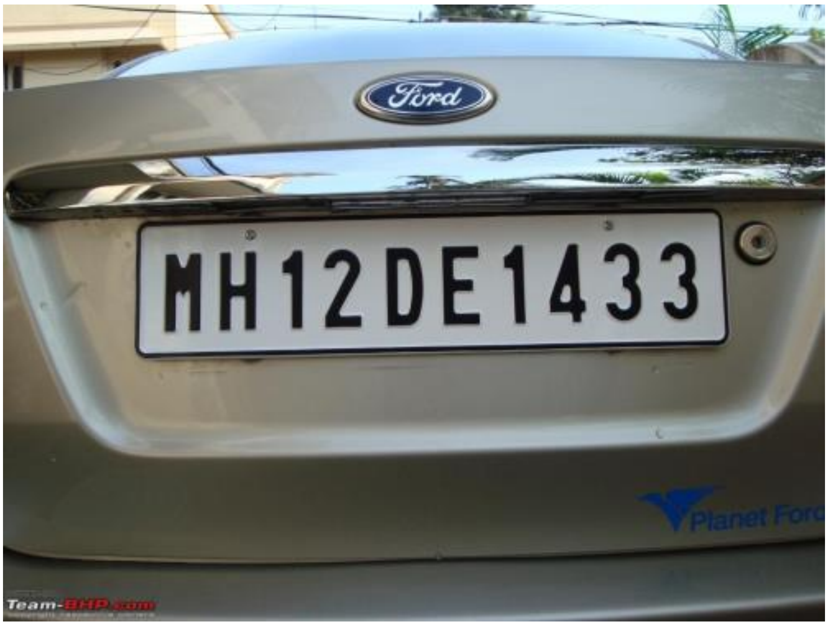
```
Saving test2.jpeg to test2 (1).jpeg
```

```
[17] from google.colab.patches import cv2_imshow
```

▾ **Read** the image and resize it

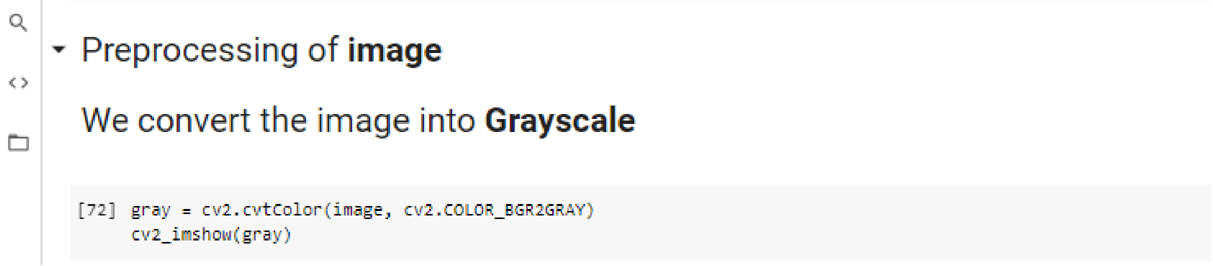
```
[71]
     image = cv2.imread('test2.jpeg')
     image = imutils.resize(image, width=500)
     cv2_imshow(image)
```

Image -:



Preprocessing of image -:

- **Turning into grayscale -:**

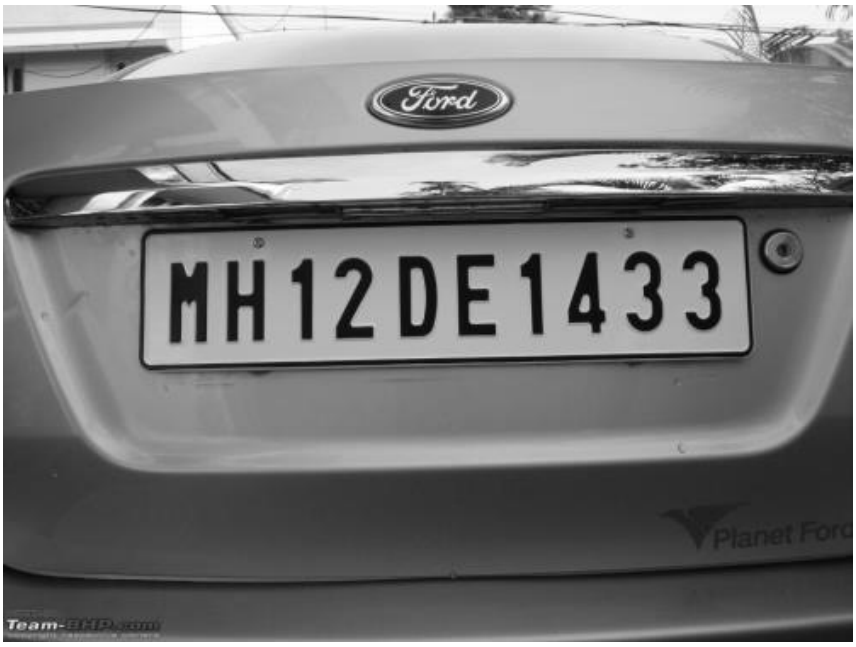


Preprocessing of **image**

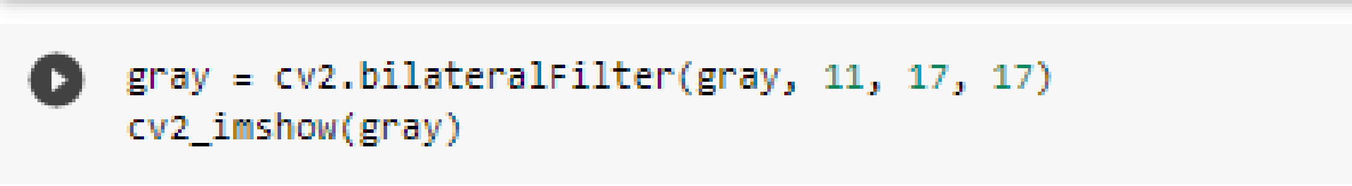
We convert the image into **Grayscale**

```
[72] gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
     cv2_imshow(gray)
```

- **Grayscale image -:**

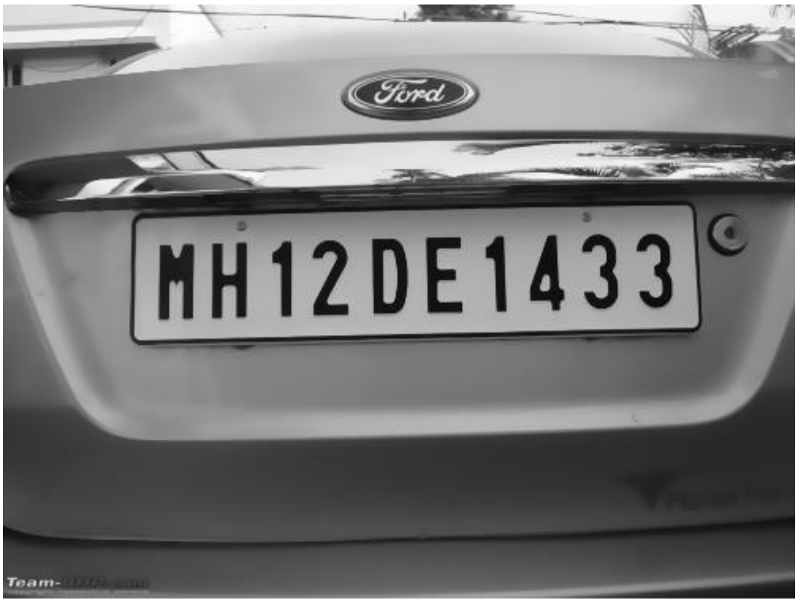


- **<u>Noise reduction is done so its easy to understand different parts of the image</u>**

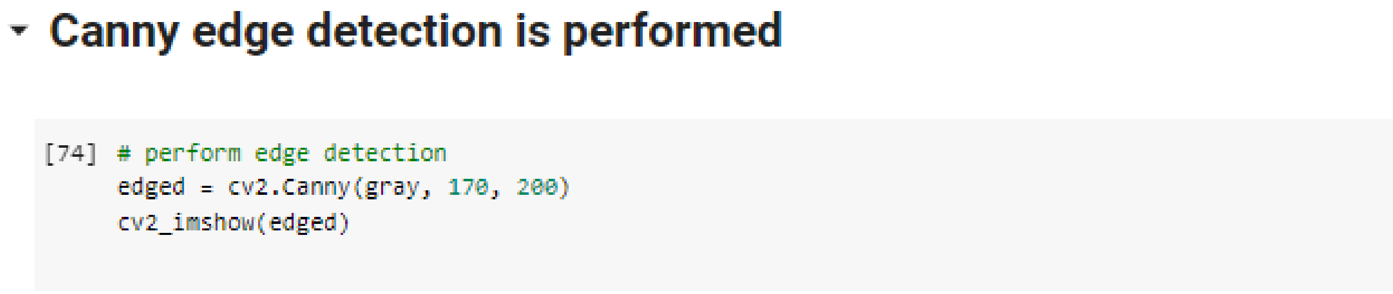


```
gray = cv2.bilateralFilter(gray, 11, 17, 17)
cv2_imshow(gray)
```

Image after reduction of noise -:



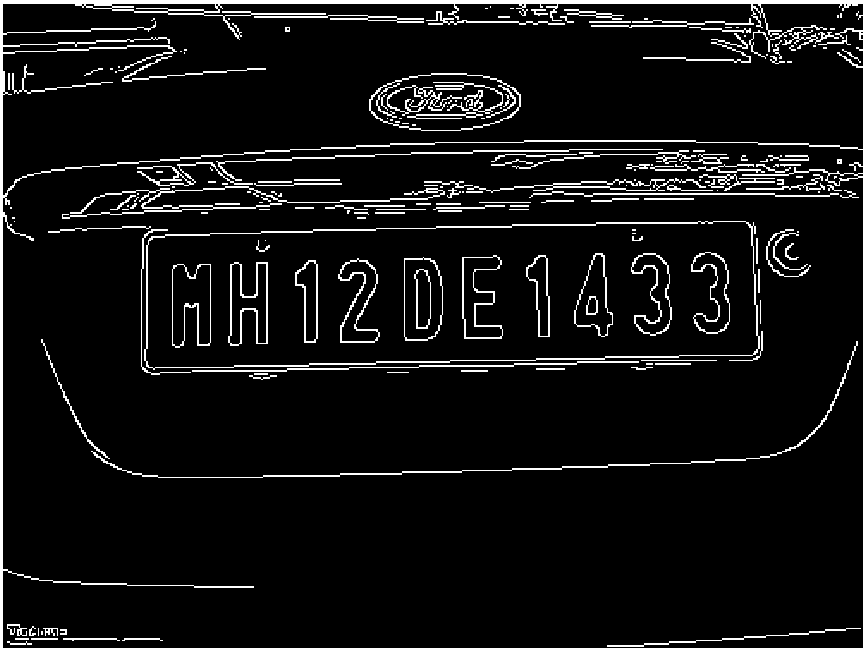
Canny edge detection -:



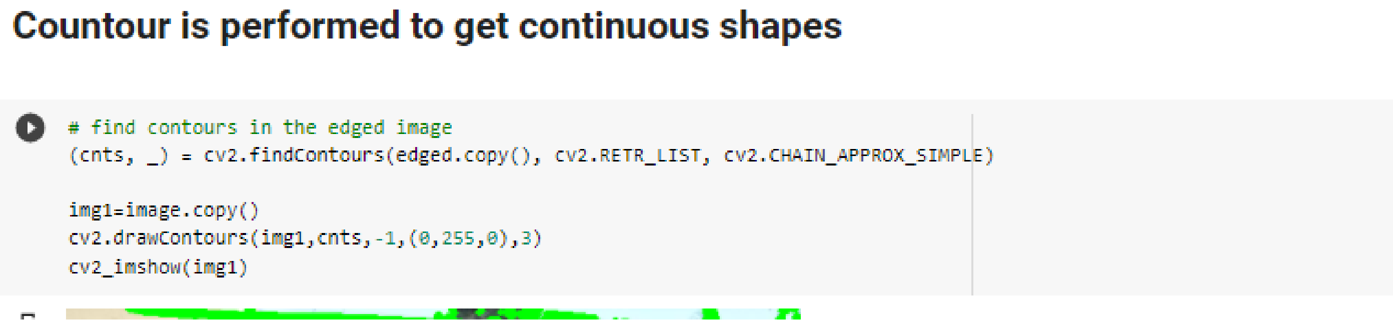
▾ Canny edge detection is performed

```
[74] # perform edge detection
     edged = cv2.Canny(gray, 170, 200)
     cv2_imshow(edged)
```

Image -:



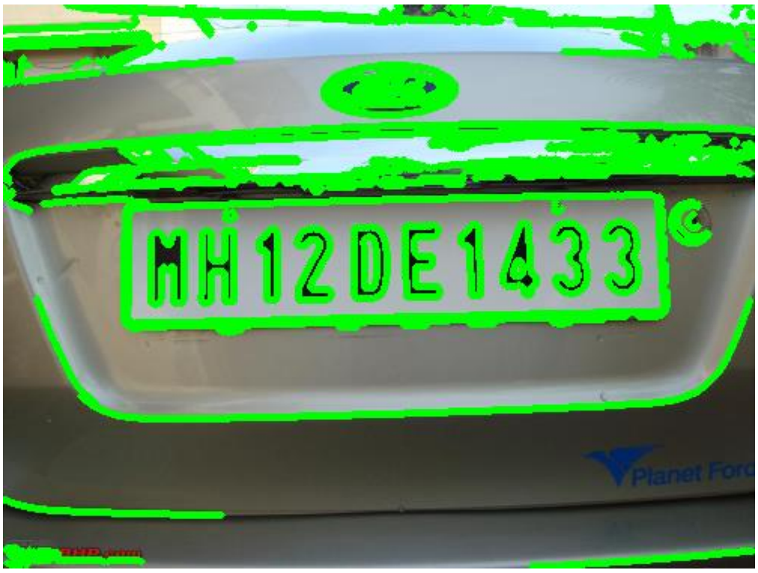
Contour detection



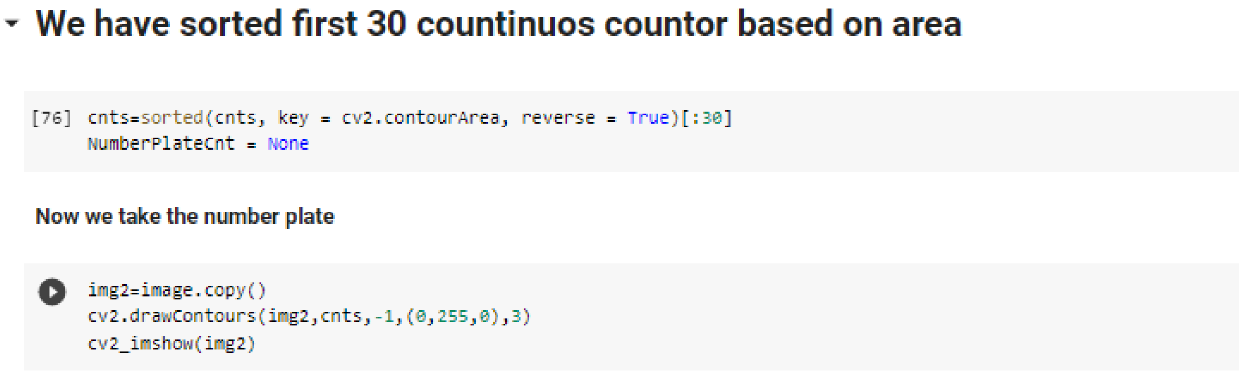
Countour is performed to get continuous shapes

```
# find contours in the edged image
(cnts, _) = cv2.findContours(edged.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)

img1=image.copy()
cv2.drawContours(img1,cnts,-1,(0,255,0),3)
cv2_imshow(img1)
```

Image :



Sorting the contours



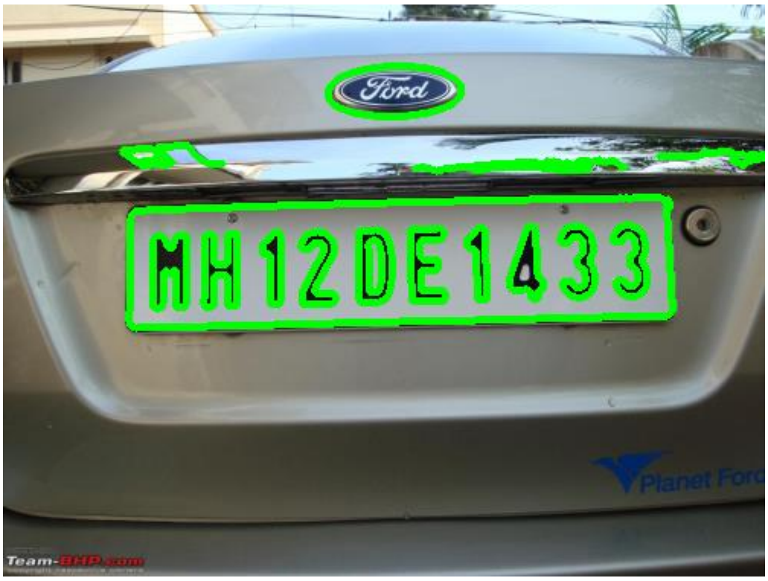
We have sorted first 30 countinuos countor based on area

```
[76] cnts=sorted(cnts, key = cv2.contourArea, reverse = True)[:30]
     NumberPlateCnt = None
```

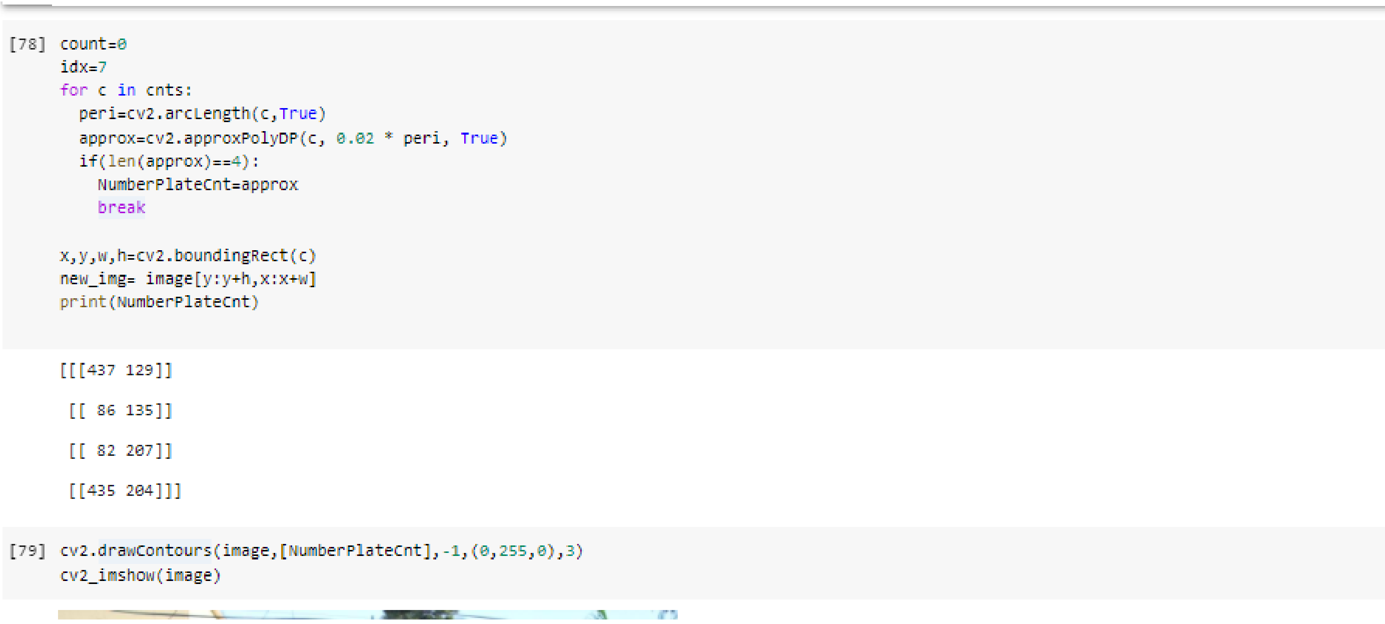
Now we take the number plate

```
img2=image.copy()
cv2.drawContours(img2,cnts,-1,(0,255,0),3)
cv2_imshow(img2)
```

Image -:



Finding the number plate -:

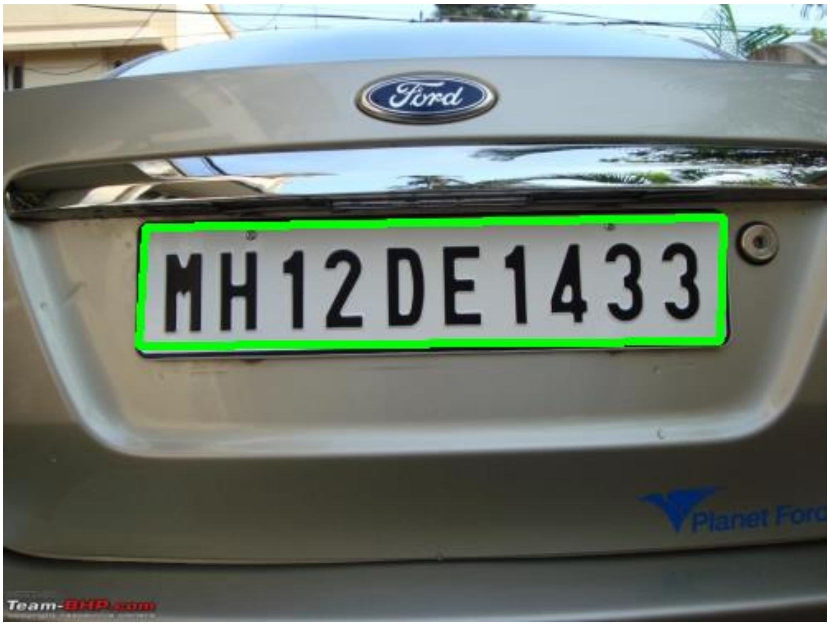


```
[78] count=0
     idx=7
     for c in cnts:
       peri=cv2.arcLength(c,True)
       approx=cv2.approxPolyDP(c, 0.02 * peri, True)
       if(len(approx)==4):
         NumberPlateCnt=approx
         break

     x,y,w,h=cv2.boundingRect(c)
     new_img= image[y:y+h,x:x+w]
     print(NumberPlateCnt)
```

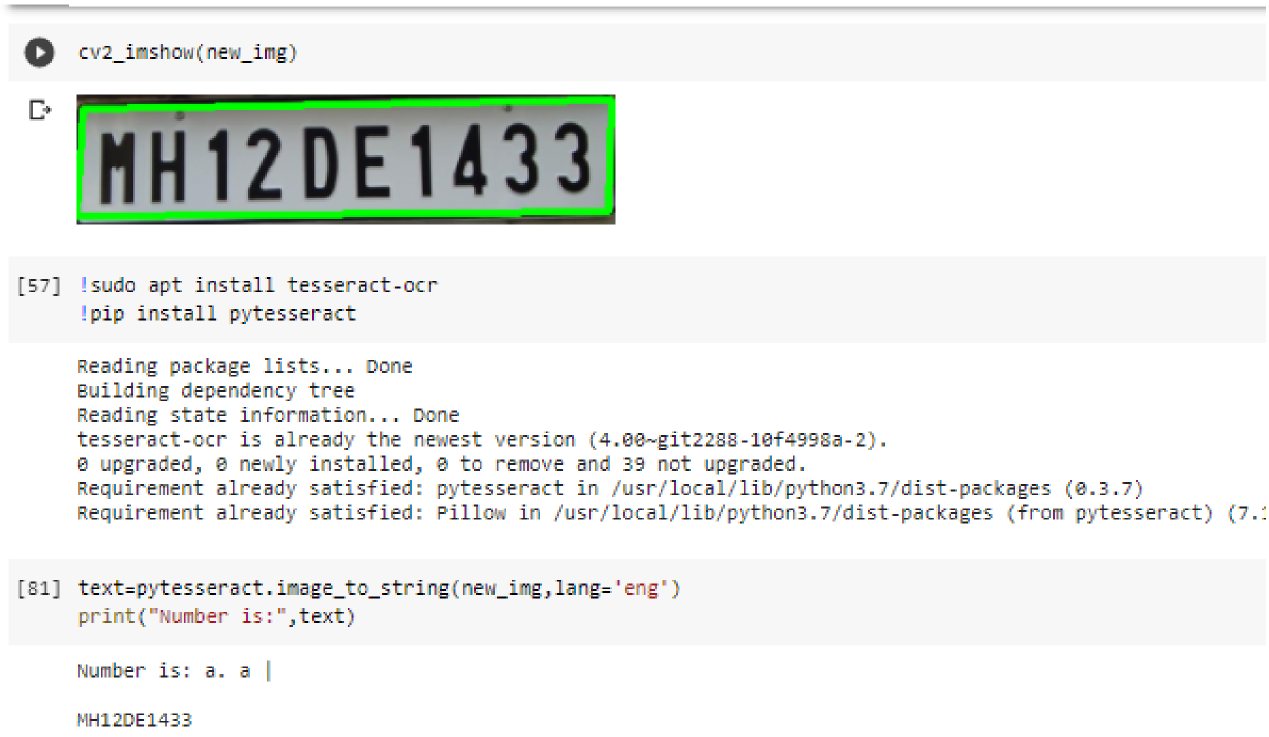
```
     [[[437 129]]

      [[ 86 135]]

      [[ 82 207]]

      [[435 204]]]
```

```
[79] cv2.drawContours(image,[NumberPlateCnt],-1,(0,255,0),3)
     cv2_imshow(image)
```

Image -:



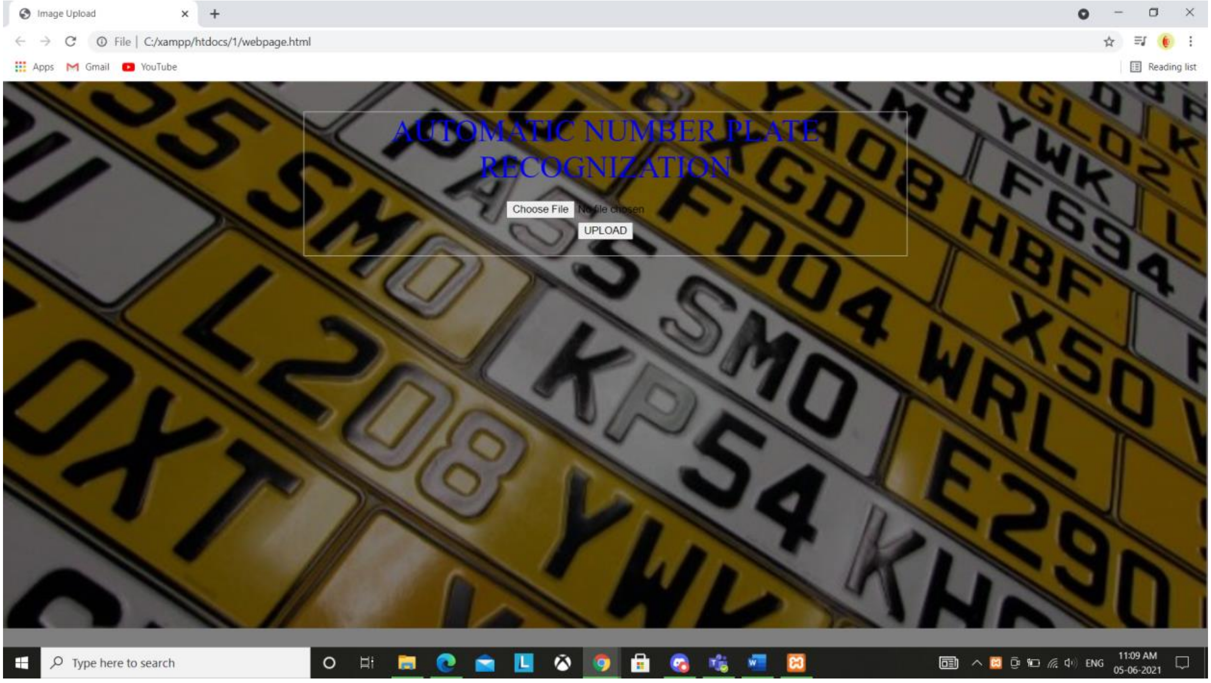
Cropping and getting image using pytesseract to get no plate



```
cv2_imshow(new_img)
```

```
[57] !sudo apt install tesseract-ocr
     !pip install pytesseract

     Reading package lists... Done
     Building dependency tree
     Reading state information... Done
     tesseract-ocr is already the newest version (4.00~git2288-10f4998a-2).
     0 upgraded, 0 newly installed, 0 to remove and 39 not upgraded.
     Requirement already satisfied: pytesseract in /usr/local/lib/python3.7/dist-packages (0.3.7)
     Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from pytesseract) (7.1

[81] text=pytesseract.image_to_string(new_img,lang='eng')
     print("Number is:",text)

     Number is: a. a |

     MH12DE1433
```

The Front end snapshot:



CONCLUSION

We have tested this with different condition and problems that occurred within the process of digital projection on the license plate as its object. Our system results show 100% accuracy for high quality image, 80% for medium quality and 45% for low quality images. This result is still lower and needs to be improved. Future extension of this work is to develop character recognition using template matching algorithm.

REFERENCES

- https://ieeexplore.ieee.org/document/8548626
- https://ieeexplore.ieee.org/document/9088975
- https://ieeexplore.ieee.org/document/9027882
- https://ieeexplore.ieee.org/document/9092977
- https://ieeexplore.ieee.org/document/8394573
- https://ieeexplore.ieee.org/document/9092977
- https://ieeexplore.ieee.org/document/8993238
- https://ieeexplore.ieee.org/document/8692040
- https://ieeexplore.ieee.org/document/8969068