Rajiv Gandhi University of Knowledge Technologies

R.K Valley, Y.S.R Kadapa (Dist)-516330

A
project report
on

Number Plate Recognition
using
Optical Character Recognition (OCR)

Submitted by

N.Suresh	R170185
M.Pavanesh	R170200
K.Brahmaiah	R170191

Under the guidance of

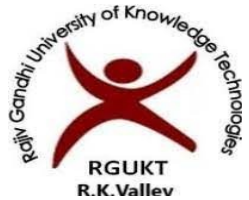
Mrs.Challa Ratnakumari
(Assistant Professor)

Department of Computer Science Engineering

This project report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Technology in software Engineering.

Sep-2022

Rajiv Gandhi University of Knowledge Technologies
IIIT,R.K.Valley,YSR Kadapa (Dist) -516330



CERTIFICATE

This is to certify that report entitled "**Number Plate Recognition using Optical Character Recognition**" Submitted by N.Suresh (R170185), M.Pavanesh (R170200), K.Brahmaiah (R170191) in partial fulfillment of the requirements of the award of bachelor of technology in computer science engineering is a bonafide work carried by them under the supervision and guidance.

The report has been not submitted previously in part or full to this or any other university or institue for the award of any degree or diploma.

GUIDE
Mrs.CHALLA RATNAKUMARI

HEAD OF THE DEPARTMENT
Mr.P HARINADHA

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and who's constant guidance and encouragement crown all the efforts success.

We would like to express my sincere gratitude to **Mrs.Challa Ratnakumari,** my project guide for valuable suggestions and keen interest throughout the progress of our project.

We are grateful to **Mr.Harinadha HOD CSE**, for providing excellent computing facilities and congenial atmosphere for progressing our project.

At the outset, we would like to thank **Rajiv Gandhi University of Knowledge Technologies (RGUKT),** for providing all the necessary resources and support for the successful completion of my course work.

DECLARATION

We hereby declare that this report entitled "**Number Plate Recognition using Optical Character Recognition**" Submitted by us under the guidance and supervision of **Mrs.Challa Ratnakumari**, is a bonafide work. We also declare that it has not been of Submitted previously in part or in full to this University or other institution for the award of any degree or diploma.

Date:- 21-09-2022

Place:-RK VALLEY

N.Suresh (R170185)

M.Pavanesh (R170200)

K.Brahmaiah (R170191)

INDEX

ABSTRACT

Optical character recognition (OCR) is the electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scenephoto or from subtitle text superimposed on an image. OCR technology plays an important role in our modern world. This technology is constantly evolving and finding its place in modern world with its wide range of applications such as language translation, hardcopy to softcopy conversion, traffic sign recognition, data entry for business documents, assistive technology for blind and visually impaired users, making searchable documents etc.

In this work the proposal is made to implement Number Plate Recognition (NPR), it is an image processing technology which uses OCR to identify the registration number of the vehicle. This NPR is aimed to reduce the human intervention in the process of extracting the registration number. NPR is very much useful in traffic regulation and provide a good assistance to traffic police in their duties. NPR provides these services by going through the following steps like preprocessing of the image, isolation of number plate, number plate extraction, extracting registration number.

INTRODUCTION

What is Number Plate Recognition?

Number plate recognition is a form of automatic vehicle identification. A number plate is the unique identification of vehicle. It is an image processing technology used to identify vehicles by their own number plates. Real time number plate recognition plays an important role in maintaining law enforcement and maintaining traffic rules. It has wide applications areas such as toll plaza, parking area, highly security areas, boarder's areas etc. Number plate recognition is designed to identify the number plate and then recognize the vehicle number plate.

What is Optical Character Recognition?

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scenephoto or from subtitle text superimposed on an image. OCR is generally an "offline" process, which analyses a static document. There are cloud based services which provide an online OCR API service. Handwriting movement analysis can be used as input to handwriting recognition. Instead of merely using the shapes of glyphs and words, this technique is able to capture motions, such as the order in which segments are drawn, the direction, and the pattern of putting the pen down and lifting it. This additional information can make the end-to-end process more accurate. This technology is also known as "on-line character recognition", "dynamic character recognition", "realtime character recognition", and "intelligent character recognition".

Applications

- Parking lot Management
- Security for Entrance
- Vehicle Tracking
- Traffic Regulation
- Tolling

LITERATURE

Optical Character Recognition (OCR) technique was used, which is a widely used technology which translates scanned images of printed text into machine encoded text. Existing projects takes the input as the clear cut image of number plate only. Here the drawback is we have to crop the number plate from the original image and we have to give that cropped image as the input to the existing systems. So this takes a lot of time and human effort to extract the registration number from the original image which contains the total view of vehicle. This existing system may lead to the error like extracting the unwanted text which is not present in the number plate but it was on the outside of the number plate of the vehicle. This means we cannot able to give the vehicles images which contains the decoratives such as stickers, names etc. Why because this system will extract those unwanted text to the result as padding so in this system we have to give input as the either the image of the number plate of the vehicle or the image of the vehicle with the number plate which does not contains any text or unwanted things outside of that number plate.

PRELIMINARIES

A bilateral filter is used for smoothening images and reducing noise while preserving the edges. Bilateral filter uses gaussian blur technique for smoothing. Bilateral filter is non-linear filter introdued to overcome the disadvantages of linear filter. Each pixel is replaced by a weighted average of its neighbors. Each neighbor is weighted by a spatial component that penalizes distant pixels and range component that penalizes pixels with a different intensity.

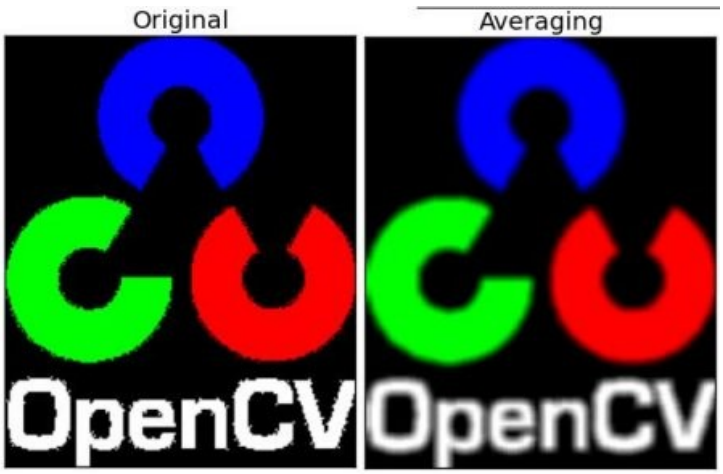


Fig.1.1

Canny edge detection algorithm is an edge detection operator that uses multi-stage algorithm to detect a wide range edges from images. Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations.

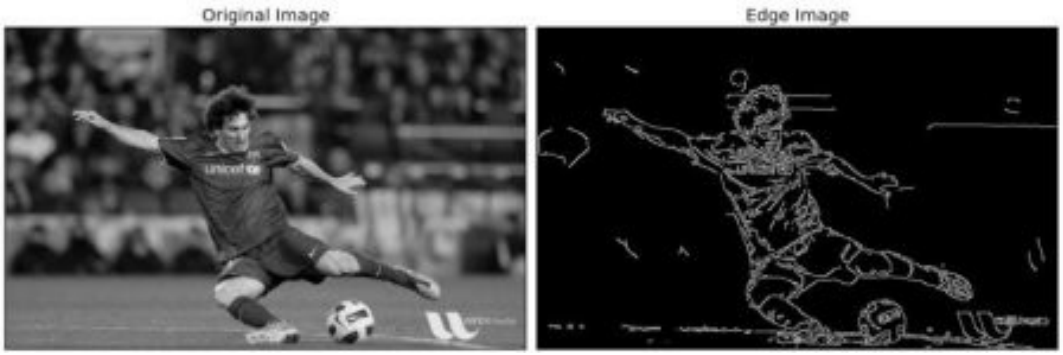


Fig.1.2

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. Finding contours is like finding white object from black background. So remember, object to be found should be white and background should be black. There are three arguments in function, first one is source image, second is contour retrieval mode, third is contour approximation method.

WORKING MODEL

Number Plate Recognition is an image-processing technology combined with machine learning used to identify registration number. This technology is used in traffic regulations to add additional good assistance to traffic police in their duties. The sole intension of this project to find the way to recognize the registration number from the given input image. Number plate recognition involves three steps. The first step (Number plate detection) is to detect the number plate from the image. We will use the contour option in OpenCV to detect for rectangular objects to find the number plate. The accuracy can be improved if we know the exact size, color and approximate location of the number plate. Normally the detection algorithm is trained based on the position of camera and type of number plate used in that particular country. In secound step (number plate extraction), Once we have detected the number plate we have to crop it out and save it as a new image. Again this can be done easily using OpenCV. In third step (Registration Number Extraction) the new image that we obtained in the previous step is sure to have some characters (numbers/alphabets) written on it. So we can perform OCR (Optical Character Recognition) on it to detect the number.

1. Number Plate Detection

Take a sample image of a car and start with detecting the License Plate on that car. Perform gray scale operation on the image. This gray scaling process is common in all image processing techniques. This speeds up other following process sine we no longer have to deal with the color details when processing an image. Every image will have useful and useless information, in this case for us only the number plate is the useful information the rest are pretty much useless. This useless information is called noise. Normally using a bilateral filter (Blurring) will remove the unwanted details from an image. We can increase/decrease the sigma color and sigma space values in order to blur out more background information, but be careful that the useful part does not get blurred.

After reducing the noise the next step is to detect the edges. There are many ways to do it, the most easy and popular way is to use the canny edge detection method from OpenCV. The syntax will be destination_image = cv2.Canny (source_image, thresholdValue 1, thresholdValue 2). The Threshold Vale 1 and Threshold Value 2 are the minimum and maximum threshold values. Only the edges that have an intensity gradient more than the minimum threshold value and less than the maximum threshold value will be displayed.

Now we can start looking for contours on our image. Once the counters have been detected we sort them from big to small and consider only the first 10 results ignoring the others. In our image the counter could be anything that has a closed surface but of all the obtained results the license plate number will also be there since it is also a closed surface.

To filter the license plate image among the obtained results, we will loop though all the results and check which has a rectangle shape contour with four sides and closed figure. Since a license plate would definitely be a rectangle four sided figure. Once we have found the right counter we save it in a variable called location.

2. Number Plate Extraction

Create a n-dimensional mask array which is of shape same as input image and which contains every index value is zero. Now draw the boundaries of the location on the mask. Now perform the bitwise_and operation between the mask and the input image. Now we will get the image which contains only number plate remaining unnecessary things are set to zero.

Now find all cordinates of the white pixels of the result image from the previous step and store every x_co-ordinates in a array name it as “x”. Then every y_co-ordinates in the array name it as “y”. Our main motive of this step is to find the first white pixel and the last white pixel. Now find the first white pixel (x1,y1) which are the mins of x-array and y-array respectively and last white pixel (x2,y2) which are the maxs of x-array and y-array respectively.

Now we have the first and last white pixel of the number plate in the image. So, Now we can crop the number plate from the image with the help of both first and last co-ordinates which denotes the exact position of the number plate and say it as cropped image.

3. Registration Number Extraction

Now set the language of reader of EasyOcr to english. Now extract the text with the help of reader from cropped image and store it as result. Finally we have extracted the registration number from the input image. If the result length is equals to none or if its prints zero it means it does not recognize anything from the image and prints nothing, if the length of result is greater than only it prints the extracted text from the number plate.

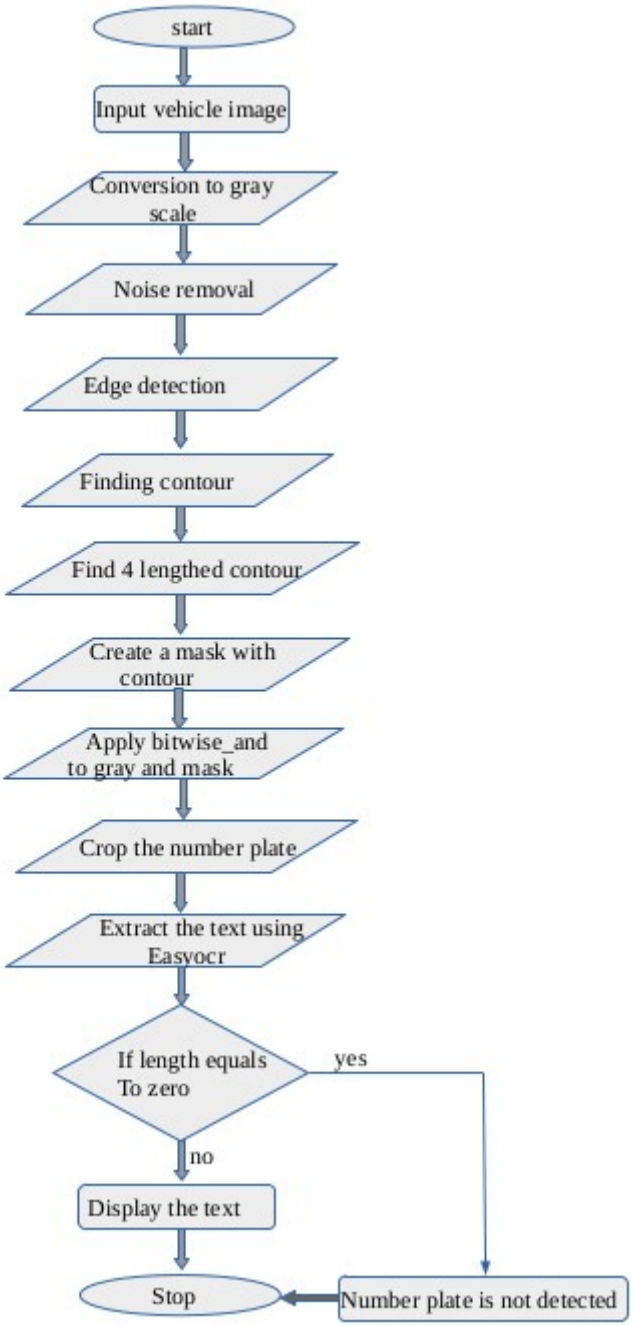


Fig-1:flow chart

PROJECT IMPLEMENTATION AND OUTPUT

The Number Plate Recognition system is implemented by using the following tools:

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It supports an object oriented programming language approach to develop applications. It is simple and easy to learn and provides lots of high level data structures.

Python is easy to learn yet powerful and versatile scripting language, which makes it is attractive for application development. Python is not intended to work in a particular area, such as web programming, that is why it is known as multipurpose programming language because it can be used with web, enterprise machine learning, image processing etc.

Easyocr

EasyOCR is a python module for extracting text from image. It is a general OCR that can read both natural scene text and dense text in document. Easyocr is an open-source and ready-to-use OCR with almost 80 supported languages. Easyocr is a Python-based library for using a ready-to-use OCR model. With this library, you don't have to worry about the preprocessing and the modeling step. In a few lines of code, you can use the OCR with greater accuracy.

Open CV

OpenCV is an open-source library for the computer vision. It provides the facility to the machine to recognize the faces or objects. OpenCV is used to read the Image from input, it includes Canny Edge Detection, Template matching, Contour, Gaussian blur and so on. OpenCV-python makes use of numpy, which is highly optimized library for numerical operations with matplot-style syntax all the OpenCV array structures are converted to and from numpy arrays.

Imutils

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV.

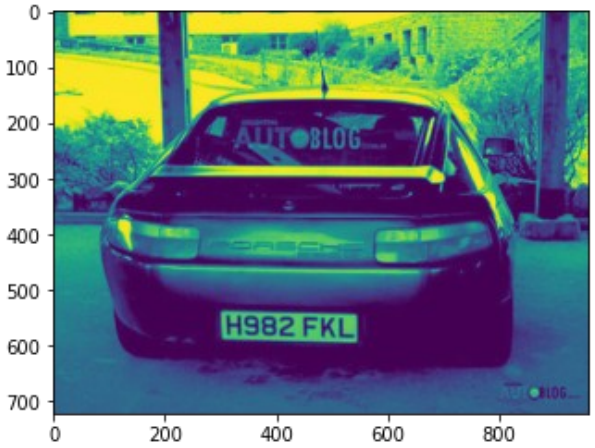
1) Installing and Import Dependencies

```
!pip install easyocr
!pip install imutils

import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

2) Read In Image,Grayscale and Blur

```
img = cv2.imread('/content/image4.jpg')
```

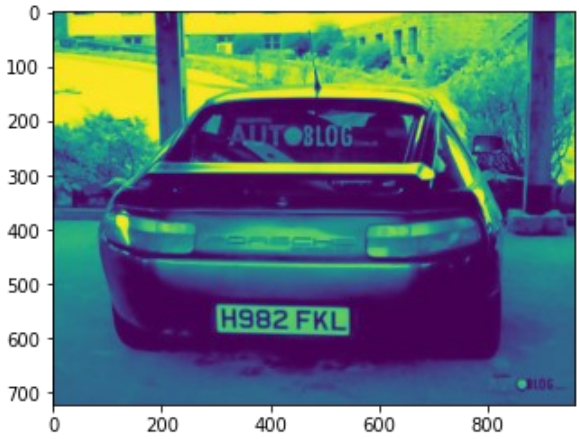

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(gray)
```



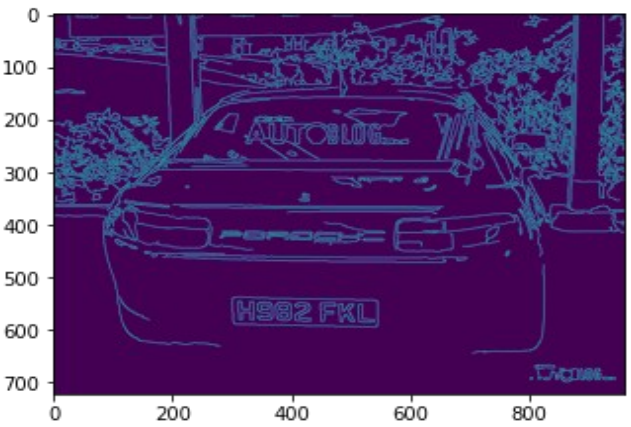
- img = cv2.imread('image4.jp'): We are taking in the image as our input.image4.jpg is the name of the image.
- gray= cv2.cvtColor(img, cv2.COLOR_BGR2GRAY): We are creating a variable gray.We are then passing our input imagecv2.cvtColor.cv2.COLOR_BGR2GRAY specifies that image should be converted to gray image.
- Plt.imshow() : We are displaying the image which is already converted to grey.

3) Reducing the noise in the greyscale image and detecting the edges of the smoothened image

```
smooth = cv2.bilateralFilter(gray, 11, 17, 17)
plt.imshow(smooth)
```




```
edged = cv2.Canny(smooth, 30, 200)
plt.imshow(edged)
```



- cv2.bilateralFilter(gray, 11, 17, 17): We are reducing the noise in the grey image hence smoothening it.
- cv2.Canny(gray, 30, 200): We are creating variable edged. We are then passing our smoothened image to cv2.canny to detect the edges in it.
- plt.imshow(edged): We are displaying the image with the smoothened image and detected edges.

4) Finding the contours from the edged image

```
keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

contours = imutils.grab_contours(keypoints)

contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
```

- findContours(): findContours function retrieve all the contours in the image that it can find .
- Cv2.RETR_TREE: It retrieves all the contours but does not create any parent-child relationship.
- Cv2.CHAIN_APPROX_SIMPLE: Removes all the redundant points on the contours detected.
- grab_contours(): We are identifying contours on our image. Input the values as they are.
- sorted(contours, key=cv2.contourArea, reverse=True)[:10]: We are sorting contours based on the minimum area 10 and ignoring the ones below that.

5) Finding the contour with four sides

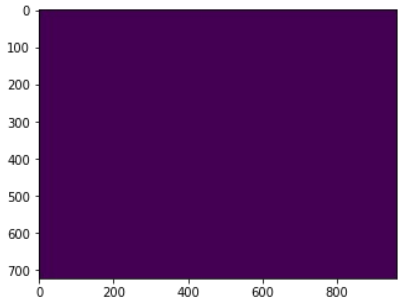
```
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break
```

- cv2.approxPolyDP(contour,10, True): *ApproxPolyDP* approximates the curve of polygon with precision.
- if len(approx) == 4:: Chooses the contours with four sides as this will probably be our number plate.

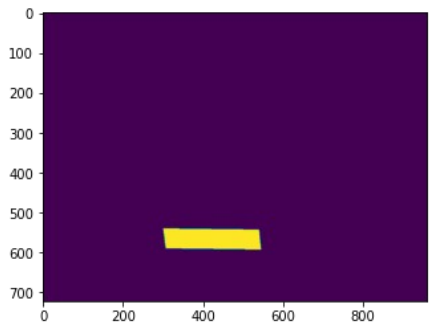
```
location
array([[[300, 540]],

       [[306, 589]],

       [[543, 592]],

       [[538, 543]]], dtype=int32)
```

6) Drawing the selected contour on the original image and applying mask

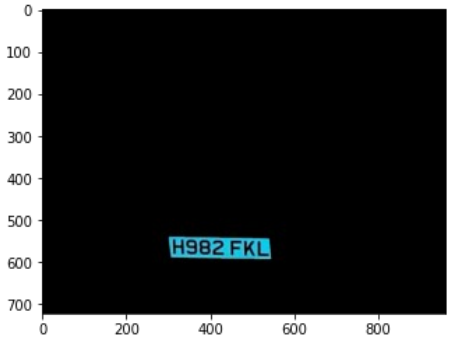
```
mask = np.zeros(gray.shape, np.uint8)
```




```
new_image = cv2.drawContours(mask, [location], 0,255, -1)
```



new_image = cv2.bitwise_and(img, img, mask=mask)



- np.zeros(gray.shape,np.init8): np.zeros() function is used to generate an array containing zeros and name it as mask.
- cv2.drawContours(image, [screenCnt], -1, (0, 255, 0), 3): This draws the contour selected to be the number plate on our original image.
- Cv2.bitwise_and(img, img, mask=mask): Bitwise operations are used in image manipulation and used for extracting contour parts in the original image. Bitwise operations helps in image masking.

7) Cropping the rectangular part identified as Number Plate

(x,y) = np.where(mask==255)

(x1, y1) = (np.min(x), np.min(y))

(x2, y2) = (np.max(x), np.max(y))

cropped_image = gray[x1:x2+1, y1:y2+1]

- x list contains x co-ordinates of all white pixels.
- Y list contains y co-ordinates of all white pixels
- (x1,y1) is the co-ordinate of the first white pixel
- (x2,y2) is the co-ordinate of the last white pixel
- with the help of both first and last co-ordinates of the white pixels we cropped the image.

8) Extracting text from the image of the cropped Number Plate

reader = easyocr.Reader(['en'])

result = reader.readtext(cropped_image)

- easyocr.Reader([‘en’]): Setting language of reader to english
- extracting the registration number from the cropped_image using reader.readtext() and storing the text in result.

9) Printing text

```
if len(result)==0:
      print("number plate is not detected")
else:
      for i in range(0,len(result)):
            print(result[i][-2])
```

10) output

```
H982 FKL
```

FUNCTIONAL TESTING

S.no	Test Case	Input	Expected Output	Actual Output	Is Actual output is same as Expected output?
1	Load an Image	.jpg image	Image to be loaded	Image is loaded	Yes
2	Gray scale conversion	Input image	Image will be converted to gray scale	Gray scale image	Yes
3	Noise removal	Gray scale image	Noise reduction will be done	Noiseless image	Yes
4	Edge detection	Noiseless image	Edges are to be detected	Edges are detected	Yes
5	Contour detection	Image with edges	Countour of the number plate	As expected	Yes
6	Number plate extraction	Image with contour of number plate	Cropped image of number plate	Cropped image is obtained	Yes
7	Registration number extraction	Cropped image	Registration number of the vehicle	Registration number is obtained	Yes
8	Load an image without number plate	Image with out number plate	Displays number plate is not detected	As expected	Yes
9	Change in Orientation	Cropped image	Registration number of the vehicle	Registration number is obtained	Yes

CONCLUSION

The Number Plate Recognition system based on Optical Character Recognition is demonstrated for the implementation of reducing the human intervention and providing good assistance in the matters of traffic regulations by following the series of image processing techniques of gray scale conversion, noise reduction, edge detection, contour detection, masking, number plate extraction and text extraction by OCR. Ever time when the traffic police takes the picture of vehicle, they can use the NPR to extract the registration number of the vehicle which will gradually reduce the human effort. Each module is functionally tested as expected and each test case is passed. This system is tested in various conditions like image without number plate, change in orientation, complete image of the vehicle. It can be easily integrated with the existing systems.

REFERENCES

1)https://www.geeksforgeeks.org/python-opencv-cv2-cvtcolor-method/

2)https://www.tutorialspoint.com/opencv/opencv_bilateral_filter.htm

3)http://www.stackoverflow.com

4)https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html

5)https://www.geeksforgeeks.org/find-and-draw-contours-using-opencv-python/

6)https://www.geeksforgeeks.org/text-detection-and-extraction-using-opencv-and-ocr/

----THANK YOU----