*License Plate Detection And Character Recognition An Application of Image Processing*

Rohan N Kalpavruksha Roshan N Kalpavruksha

*ECE                       ECE*

*PES University        PES University*

[rohankalpavruksha@yahoo.com](mailto:rohankalpavruksha@yahoo.com)          [rkalpavruksha@yahoo.com](mailto:rkalpavruksha@yahoo.com)

Guide Name- Raghavendra M J

***Abstract: This paper presents a method for license plate recognition through analysis of vehicle images. Accurate location of license plate and extracting characters of plate are implemented. License plate is detected and characters in it is detected and the result is printed both on the console and the image. At last, characters extraction is done by using open cv.. And the extraction characters are matched with the templates. Experimental results show that this approach can recognize license plates more effectively and establish a good technical base for the future advanced license plate recognition.***

***Keywords: Vehicle number detection, character recognition, python, open cv, grey scale, image processing***

I. INTRODUCTION

The use of vehicles has increased exponentially day by day and thereby violation of traffic rules, theft of vehicles, entering in restricted areas, high number of accidents lead to increase in the crime rates.

Vehicle license plate detection technique can be significantly used in the field of vehicle security and safety system.

The motto behind the project is detecting the cars number plate *automatically* and provide them with pay-slip and then open the road for that particular car. Parking authorities also use this system for allowing the vehicle to *park* in their area. In this system, firstly we *capture the image* of number plate then process it and *read* each and every character present in the number plate for their perfect recognition. The most significant phase is *OCR*, where the letterings on the image of number plate are changed into the texts which can be decoded later

Number Plate Recognition system is a *security system*. Image processing concept is used in Number Plate Recognition system. OCR (*Optical Character Recognition*) scheme is also applied in this for reading the image of vehicle number plate.In India, the number plate containing white background with black foreground color is used for private cars and for the commercial vehicles yellow is used as background and black as foreground color.

Every module plays a vital role in gaining *efficiency and accuracy*. The challenges here are font size and style variations, angle of the picture, low contrast light effect, speed of the vehicles. Machine learning algorithms are fed through *unsupervised* learning.

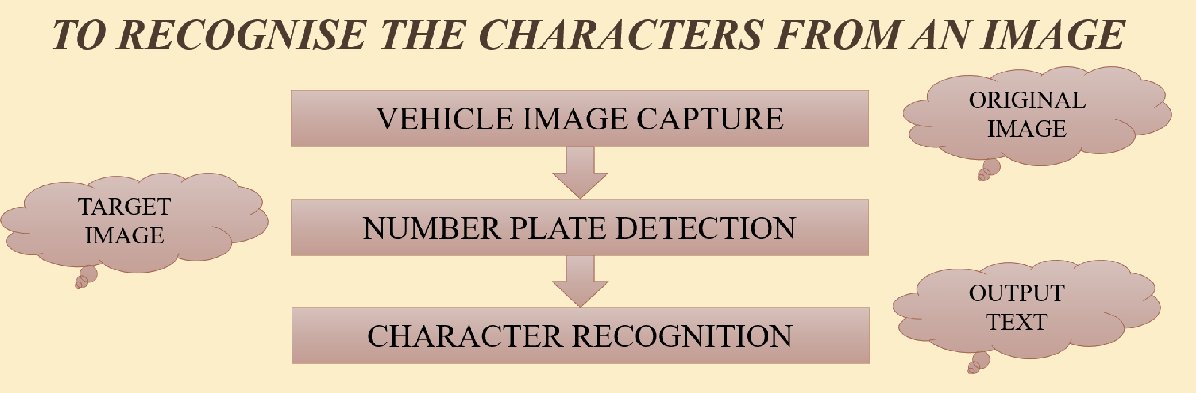


Fig.1.Introduction to Character Recognition

II.LITERATURE SURVEY

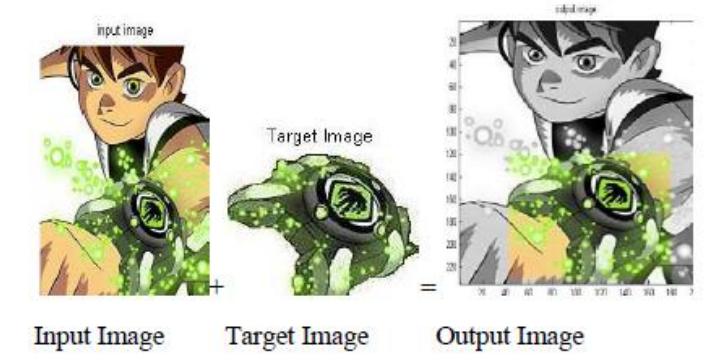
**Priyanka Prabhakar, Anupama P, Resmi S R, “Automatic Number Plate Detection and Recognition“,International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014**

Preprocessing such as applying sharpness, histogram equalization, smooth the rectangle, Threshold is applied. Methodology used is i/p image ->edge detection->hough transform->hough limit->character segmentation->recognition. The accuracy acquired through this is 98% for shorter distance and 95 % for longer distance. The scope for future is reducing computation overhead introducing parallelism into the design makes it longer economical and time efficient.

**Mahesh Babu K, M V Raghunadh, “Vehicle Number Plate Detection and Recognition using Bounding Box Method”, 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), ISBN No.978-1-4673-9545-8 106**

Preprocessing such as frame selection and number plate extraction I = 0.114 \*R+0.587\*G+0.299\*B is used. Methodology used is localization, dilation, segmentation and recognition. The accuracy acquired through this is 93.33%. The scope for future is to deal with blurred images and similar characters.

III. METHODOLOGY



The steps that are used in this License plate recognition project are as follows

1. INPUT ORIGINAL RGB IMAGE
2. CONVERSION TO GREYSCALE IMAGE
3. CANNY EDGE DETECTION
4. FIND CONTOURS BASED ON EDGES
5. INITIALIZE LICENSE PLATE CONTOUR
6. INITIALIZE X,Y COORDINATES
7. FIND THE CONTOUR WITH 4 POTENTIAL CORNERS
8. CREATE ROI AROUND IT
9. REMOVING NOISE FROM TARGET IMAGE
10. TEXT RECOGNITION
11. WRITE THE TEXT ON OUTPUT IMAGE

IV.CODE

!pip install pytesseract #pytesseract is useful in reading, editing and obtaining text from images

!sudo apt install tesseract-ocr #optical chracter recognition

from google.colab import files

from google.colab.patches import cv2\_imshow

import cv2

import pytesseract

import io

from PIL import Image #python imaging library

image = cv2.imread('car0.jpg') #Read the image file

cv2\_imshow(image)

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY) #Convert to GrayScale image

cv2\_imshow(gray\_image)

canny\_edge = cv2.Canny(gray\_image, 170, 200) #Canny Edge Detection

cv2\_imshow(canny\_edge)

# Find contours based on Edges

contours, new  = cv2.findContours(canny\_edge.copy(), cv2.RETR\_LIST, cv2.CHAIN\_APPROX\_SIMPLE)

contours=sorted(contours, key = cv2.contourArea, reverse = True)[:30]

print(contours)

# Initialize license Plate contour and x,y coordinates

contour\_with\_license\_plate = None

license\_plate = None

x = None

y = None

w = None

h = None

# Find the contour with 4 potential corners and creat ROI around it

for contour in contours:

        # Find Perimeter of contour and it should be a closed contour

        perimeter = cv2.arcLength(contour, True)

        approx = cv2.approxPolyDP(contour, 0.01 \* perimeter, True)

        if len(approx) == 4: #see whether it is a Rect

            contour\_with\_license\_plate = approx

            x, y, w, h = cv2.boundingRect(contour)

            license\_plate=gray\_image[y:y+h,x:x+w]

            break

print(x)

print(y)

print(w)

print(h)

cv2\_imshow(license\_plate)

# Removing Noise from the detected image, before sending to Tesseract

license\_plate = cv2.bilateralFilter(license\_plate, 11, 17, 17)

(thresh, license\_plate) = cv2.threshold(license\_plate, 150, 180, cv2.THRESH\_BINARY)

cv2\_imshow(license\_plate)

license\_\_plate = cv2.imread('license\_\_plate.png')

#Text Recognition

text = pytesseract.image\_to\_string(license\_\_plate)

print("License Plate :", text)

#Draw License Plate and write the Text

image = cv2.rectangle(image, (x,y), (x+w,y+h), (0,0,255), 3)

cv2\_imshow(image)

# Write the text on the image

image = cv2.putText(image, text, (x-100,y-50), cv2.FONT\_HERSHEY\_SIMPLEX, 3, (0,255,0), 6, cv2.LINE\_AA)

cv2\_imshow(image)

V.PROGRAM RESULT



STAGE 1



STAGE 2



STAGE 3



STAGE 4 

STAGE 5



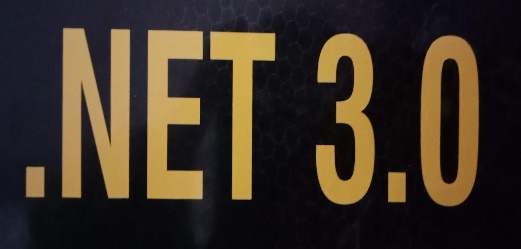
STAGE 6



STAGE 7

VI.DATASET

We have used all letters in a image and huge number of car images to detect license plate. Some of them used in this project are



















VII.CONCLUSION

This project can be further exploited for vehicle owner identification, vehicle model identification traffic control, vehicle speed control and vehicle location tracking. It can be further extended as multilingual character recognition to identify the language of characters automatically based on the training data.

It can provide various benefits like traffic safety enforcement, security- in case of suspicious activity by vehicle, easy to use, immediate information availability- as compare to searching vehicle owner registration details manually and cost effective for any country. For low resolution images some improvement algorithms like super resolution of images should be focused. Most of our project focus on processing one vehicle number plate but in real-time there can be more than one vehicle number plates while the images are being captured.

Obviously, the accuracy of the recognition is the most important in this system. Therefore, this application should be optimized and modified for overcoming the accuracy limitations. In order to make the recognition more precise, we should add some preprocesses to remove the interferences. Moreover, we would continue the further study for license plate recognition in some complicated environments, such as vehicles at dark night or in heavy rains and so on. If we could accomplish all of the objectives, this application would have a very promising future.

ACKNOWLEDGEMENT

Our warm gratitude to PES University Electronics and communication department for allowing us to work with them on Character Recognition – An application of image processing and special thanks to our respected guide  Dr.M.J.Raghavendra who  helped us enrich our knowledge about the topic by sharing his valuable understandings and for enlightening us with his skills.

REFERENCES

1. Priyanka Prabhakar, Anupama P, Resmi S R, “Automatic Number Plate Detection and Recognition“,International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014
2. Mahesh Babu K, M V Raghunadh, “Vehicle Number Plate Detection and Recognition using Bounding Box Method”, 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), ISBN No.978-1-4673-9545-8 106
3. Abhishek Kashyap, B. Suresh, Anukul Patil, Saksham Sharma, Ankit Jaiswal, “Automatic Number Plate Recognition“, International Conference on Advances in Computing, Communication Control and Networking (ICACCCN2018), ISBN: 978-1-5386-4119-4/18/$31.00 ©2018 IEEE
4. Deniing Jiang, Tulu Muluneh Mekonnen, Tiruneth Embiale,AshenafiGebrehiwot “Car Plate Recognition System” in Fifth internation conference on Intelligent Networks and Intelligent Systems, IEEE, 978-0-7695-4855-5/12 2012.
5. Rahim Panahi, Iman Gholampour, "Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High Speed Applications",. IEEE Transactions on Intelligent Transportation Systems, 2017
6. <https://www.ijert.org/a-review-paper-on-vehicle-number-plate-recognition>
7. <https://www.imedpub.com/articles/a-review-paper-on-license-plate-recognition>
8. <https://ieeexplore.ieee.org/document/8748287>