



SRI RAMACHANDRA

INSTITUTE OF HIGHER EDUCATION AND RESEARCH

(Category - I Deemed to be University) Porur, Chennai

SRI RAMACHANDRA ENGINEERING AND TECHNOLOGY

RECOGNITION OF VEHICLE NUMBER PLATE USING

RASPBERRY PI AND PI CAMERA

PROJECT REPORT

Quarter I (Year 2)

Submitted by

CHEEKATI VENKATA ROHIT E0220015

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

**COMPUTER SCIENCE AND ENGINEERING
(Cyber Security & Internet Of Things)**

**Sri Ramachandra Engineering and Technology Sri Ramachandra Institute of Higher
Education and Research, Porur, Chennai -600116**

SEPTEMBER, 2021

INTRODUCTION

Vehicle's license plate is a very specific identity by which an individual vehicle can be identified. This Vehicle plate recognition software allows one to capture a license plate, extract the numbers on the plate. In India Motor Vehicles Sales Growth rate is increasing with an average growth rate of 9.3 %, which also indirectly refers to drastic growth in road accidents. Identifying the license plate and charging unlawful vehicles on the road will help the law enforcement agents. In this paper, we present Vehicle license plate recognition using Raspberry pi. A Camera modified into blanketed to help in taking pictures the plate range pictures and it is interfaced to a Raspberry pi processor for authentication. Using the Open Computer Vision (Open CV) and Optical Character Recognition (OCR), the tool can extract numbers from the captured plate image and definitely automate the license plate reputation. The experimental consequences from severa finding out in unique locations and conditions show that the tool achieved better than most of the baseline studies considered.

Keywords : Vehicle Plate, Recognition system,OpenCV ,OCR ,Raspberry Pi ,PI Camera

RASPBERRY PI

The Raspberry Pi is a low cost, credit card sized computer which we can plug into an external monitor, standard keyboard, mouse and we can kinda use it like a desktop. Raspberry Pi is the name of a series of single board computers made by Raspberry Pi foundation. It's capable of doing almost everything, what you would expect a normal desktop/laptop would do , from browsing the internet, playing HD videos, to playing games. The latest Raspberry Pi 4 has 1.5 GHz CPU, 4GB RAM, Gigabit Ethernet, 2xUSB 2.0 Ports, 2xUSB 3.0 Ports, 2 Micro HDMI with 4K support and USB Type -C Power Support.

PI CAMERA

The Pi camera module is a portable lightweight camera that supports Raspberry Pi. It communicates with Pi using the MIPI camera serial interface protocol. It is normally used in image processing, machine learning or in surveillance projects. It is commonly used in surveillance drones since the payload of the camera is very less. Apart from these modules Pi can also use normal USB webcams that are used along with computers. In this project we use a PI camera to capture the license plate.

OPENCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. It is a very useful computer vision application such as video analysis, CCTV footage analysis and image analysis. It was written by C++ and it has more than 2500 optimized algorithms. In this project we are using opencv for image processing. Image processing is a method to perform some operations on an image in order to get an enhanced image so we can get some useful information from that. So with the help of OpenCv library we can basically analyze the image, crop the image, change the color, enhance it in order to get the output that we want.

OPTICAL CHARACTER RECOGNITION

Optical Character Recognition is a process of recognizing text inside images and converting it into an electronic form. These images could be of handwritten text, printed text like documents, receipts, name cards. OCR has two parts to it. The first part is text detection where the textual part within the image is determined. This localization of text within the image is important for the second part of OCR, text recognition, where the text is extracted from the image. Using these techniques together is how you can extract text from any image. So in this project we are using OCR to extract the vehicle number from the license plate.

IMPLEMENTATION

The device uses an onboard pc, which is usually termed as Raspberry Pi. The Raspberry Pi can correctly speak with the output and input modules which can be being used. The Vehicle Plate photograph is captured with the interfaced 2MP Pi Camera and it's being saved in an SD card for preprocessing and reputation. After the preprocessing is completed through the initiation of the OpenCV, the characters at the plate are identified by the usage of the Optical Character Recognition (OCR) and the corresponding characters observed and displayed on monitor.

Components Required:

- Raspberry Pi
- Pi camera

A. Image Acquisition

The plate range photograph acquisition is finished via way of means of the raspberry pi camera which captures the photograph while it is triggered. The positioning of the pi camera became based to without difficulty seize an over dashing automobile plate range. The captured photograph is then saved for processing.

B. Preprocessing

After the virtual photograph has been received the subsequent step is to carry out photograph preprocessing which goal is to permit the photograph to be appropriate for clean popularity with the aid of improving the photograph quality. It may be visible as a crucial and commonsection in any pc imaginative and prescient system. In this study, the preprocessing wished entails processes:

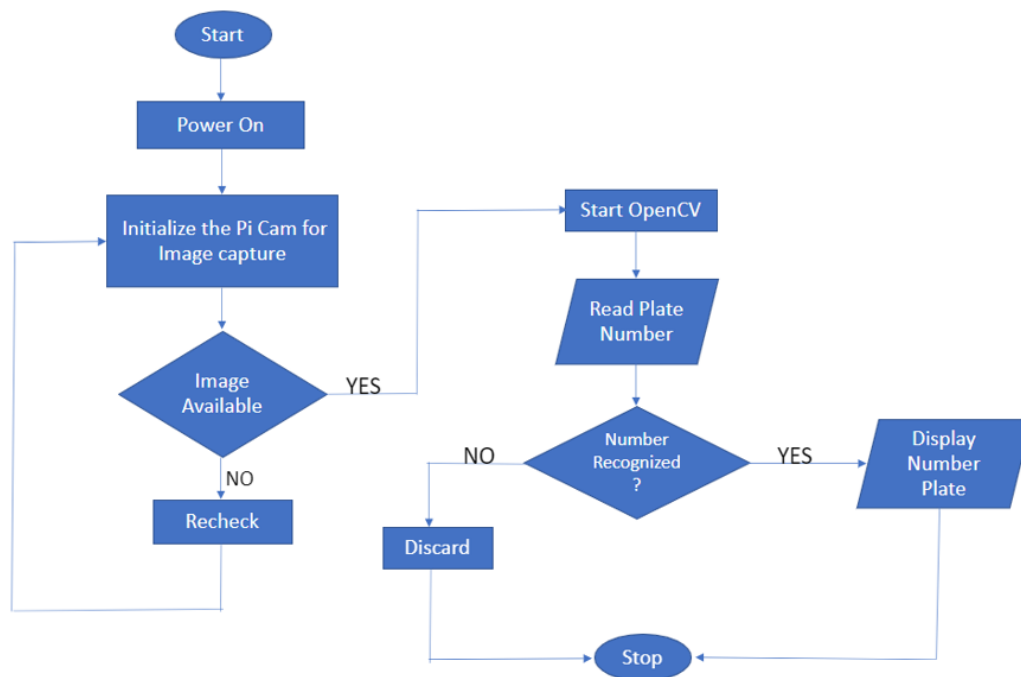
1) Size Modification

This was necessary to reduce the size of the image from the camera to a feasible aspect ratio.

2) Conversion of Color Space

Images captured with the camera can either be in a raw format or encoded into some multimedia standards. It is mostly in RGB mode, with three channels (red, green and blue). It is believed that the number of channels in the image defines the amount of colour information available in the image. Thus, the image capture must be converted to grayscale to make it appropriate for recognition.

WORK FLOW



Also in the flow chart , The process for the operation of the gadget is proven in a flowchart. The flowchart explains the step-with the aid of using-step operation of the automobile plate recognition gadget. First, the gadget is switched directly to initialize the contact display and the Pi camera. It then examines if there's a picture from the Pi camera. If true, it initializes the OpenCV for person extraction and recognition. After that is done, the end result is displayed at the display. This summarizes the entire manner that the gadget undergoes as illustrated in the flow chart. This is pretty speedy and green so long as the picture is well captured, the principle assignment is dealt with with the aid of using the OpenCV library going for walks at the Raspberry Pi.

In connecting the camera module, it is attached to the Raspberry Pi by a ribbon cable. The cable is attached to a special connector behind the Ethernet socket. The camera module is configured using:

```
Sudo raspi-config
```

```
Raspistill -o image1.jpg
```

To capture video, raspivid command is used

```
Raspivid -o video.h264 -t 10000
```

The number at the end is the recording duration in milliseconds in this case.

Commands to Install OpenCV on Raspberry PI

```
sudo apt-get install libhdf5-dev libhdf5-serial-dev
```

```
sudo apt-get install libqtwebkit4 libqt4-test
```

```
sudo pip install opencv-contrib-python
```

SOURCE CODE

```
import cv2

import imutils

import numpy as np

import pytesseract

from PIL import Image

img = cv2.imread('4.jpg',cv2.IMREAD_COLOR)

img = cv2.resize(img, (620,480) )

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #convert to grey scale

gray = cv2.bilateralFilter(gray, 11, 17, 17) #Blur to reduce noise

edged = cv2.Canny(gray, 30, 200) #Perform Edge detection

# find contours in the edged image, keep only the largest

# ones, and initialize our screen contour

cnts = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

cnts = imutils.grab_contours(cnts)

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

screenCnt = None

# loop over our contours

for c in cnts:

    # approximate the contour

    peri = cv2.arcLength(c, True)

    approx = cv2.approxPolyDP(c, 0.018 * peri, True)
```

```
# if our approximated contour has four points, then
# we can assume that we have found our screen
if len(approx) == 4:
    screenCnt = approx
    break
if screenCnt is None:
    detected = 0
    print "No contour detected"
else:
    detected = 1
if detected == 1:
    cv2.drawContours(img, [screenCnt], -1, (0, 255, 0), 3)
# Masking the part other than the number plate
mask = np.zeros(gray.shape,np.uint8)
new_image = cv2.drawContours(mask,[screenCnt],0,255,-1,)
new_image = cv2.bitwise_and(img,img,mask=mask)
# Now crop
(x, y) = np.where(mask == 255)
(topx, topy) = (np.min(x), np.min(y))
(bottomx, bottomy) = (np.max(x), np.max(y))
Cropped = gray[topx:bottomx+1, topy:bottomy+1]
#Read the number plate
text = pytesseract.image_to_string(Cropped, config='--psm 11')
print("Detected Number is:",text)
```



```
cv2.imshow('image',img)
cv2.imshow('Cropped',Cropped)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

RESULT

The implementation of the layout exceeded the entire vital layout take a look at conducted. Each level in the improvement system becomes examined and evaluated in relation to the prevailing setup. The stages involved are pre-implementation and post-implementation.

A.Pre-implementation testing

The pre-implementation checking is completed on the additives to make sure that everything works as specified.

B. Post-implementation testing

The test was conducted to ensure that the system components are working properly. We carried out a look at the usage of exceptional vehicle numbers at diverse situations to display the effectiveness of our gadget. These exams show that the gadget plays pretty well compared to the present systems. It captured the license plate and recognised the text and numbers and finally it displaying it in the output terminal.

CONCLUSION

The improvement of the car plate reputation system indicates how the usage of the OpenCV and OCR may be applied withinside the individual extraction and reputation of car plates. Although, this layout is only a evidence of the concept (prototype) and hence, consists of the very preliminary step in a take a look at that has the capability to be improved withinside the future. In order to enhance the layout efficiency, other kinds of individual extraction and reputation technology will be examined.

REFERENCES

- ❖ <https://circuitdigest.com/microcontroller-projects/how-to-install-python-opencv-on-raspberry-pi>
- ❖ <https://circuitdigest.com/microcontroller-projects/raspberry-pi-configuration>
- ❖ <https://circuitdigest.com/microcontroller-projects/license-plate-recognition-using-raspberry-pi-and-opencv>
- ❖ <https://iotdesignpro.com/projects/real-time-license-plate-recognition-using-raspberry-pi-and-python>
- ❖ <https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=iot%20using%20raspberry%20pi>