Abstract

The primary purpose of the bus monitoring system is to track which vehicle entered the college and obtaining their information. The proposed bus monitoring system consists of two fundamental sections: To collect data from the vehicle which is entering monitoring the city vehicles from website or android based phones. Automatic number plate recognition (ANPR) is an image processing technology which uses number (license) plate to identify the vehicle. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate.

INDEXING

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

I. INTRODUCTION

I.1 Scope:

The scope of the proposed system can be expanded by adding more sensor and code modification. . The system can incorporate different other sensors like camera, ir sensor. . Different other techniques like ANPR can be also be used for further improvement in result. These techniques can use this data for better result in future about Bus Check-In Monitoring system.

I.2 Importance:

Bus check in monitoring system is the process where a vehicle enters in to the house its scan the number plates and send data to the user. It uses in many ways like household and institutional purposes and in various places and we know the complete information of which vehicle is out and in and we can save that data for future.

I.3 Purpose:

There are many great applications for BUS CHECK-IN MONITORING SYSTEM but this project will detect fast which the vehicle is entered. The purpose of the project is to know which vehicle has entered and data can be stored.

CHAPTER-2

COMPONENTS USED

II. COMPONENTS USED

II.1 Raspberry Pi

II.2 Pi Camera v2 Module

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. We want to see the Raspberry Pi being used by kids all over the world to learn to program and understand how computers work.

To Set Up & Start your Raspberry Pi
The Raspberry Pi board comes equipped with an SD card. This slot permits us to insert an SD card and that can use it as our devices. The SD card is a main storage device for raspberry pi board like a hard disk of a personal computer. The bootable Linux operating system is loaded onto the card, you are planning to use. The raspberry pi supports Linux, Qtonpi, ARM, Mac operating systems. You can select one OS; you will need to write it to an SD card using a Disk manager application. You can also use other storage mechanism, like USB external hard drive or USB drive. There are a numerous brands of SD cards are available in the market in different sizes. The raspberry pi supports max 64 GB SD card.

The limited CPU power and RAM of the Raspberry Pi make it particularly challenging to squeeze all the machine learning ANPR code into a small footprint.

Thankfully, Plate Recognizer did not sacrifice its plate recognition algorithms, but instead revisited and optimized each aspect to fit the Raspberry Pi. For example, we leveraged the extractor architecture to optimize for latency, allowing us to use about 2x fewer operations with 25-30% fewer parameters and run about 25-40% faster. Moreover, we were able to maintain the TensorFlow model with two neural networks that we currently maintain with our ANPR SDK on Windows, Mac and Linux machines

CHAPTER-3

SYSTEM DESIGN

III. WORKING FLOW AND DESIGN OF PROJECT

III.1 About connections

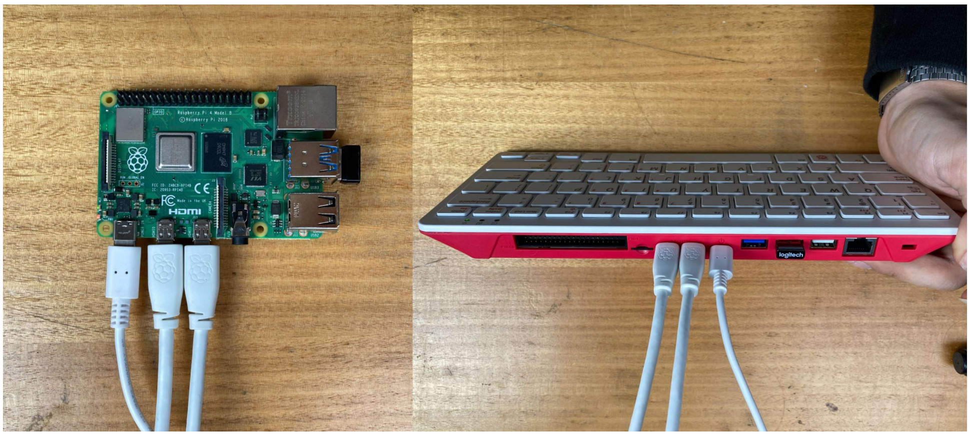
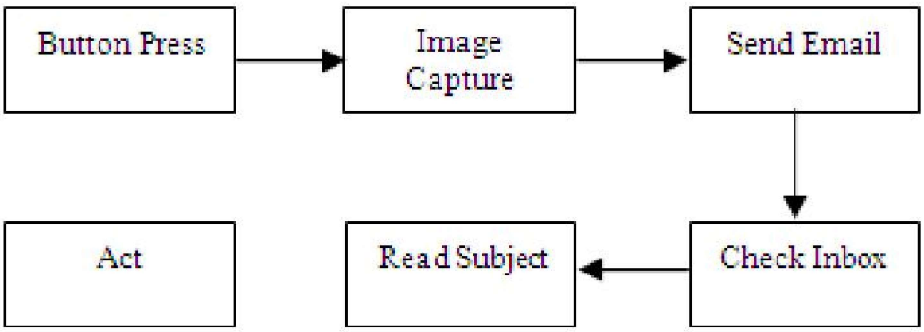


Figure 3.1



The pictorial representation of the BUS CHECK-IN MONITORING SYSTEM.

III.2. WORKING

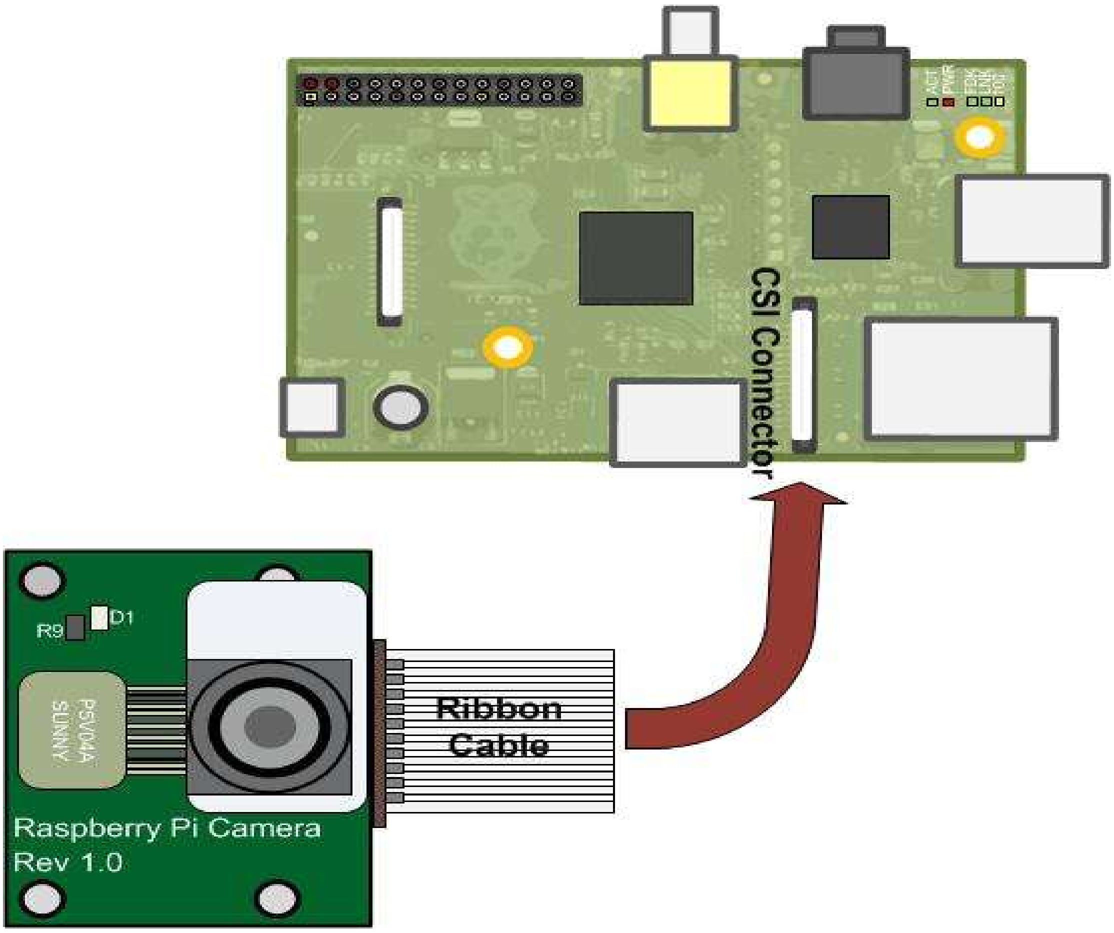


Figure 3.2

Raspberry Pi High Quality Camera is the latest camera accessory from the Raspberry Pi Foundation. It offers higher resolution (12 megapixels, compared to 8 megapixels), and sensitivity (approximately 50% greater area per pixel for improved low-light performance) than the existing Camera Board V2, and is designed to work with interchangeable lenses in both C- and CS-mount form factors.

The camera board attaches to the Raspberry Pi via a ribbon cable. One end of the ribbon cable goes to the camera PCB and the other end attached to Raspberry Pi hardware itself. You need to get the ribbon cable connections the right way, or the camera will not work. On the camera PCB, the blue backing on the cable should be facing away from the PCB, and on the Raspberry Pi hardware it should be facing towards the Ethernet connection.

Create a Camera Board object
Create a camera board object by executing the following command on the MATLAB prompt.

```
clear rpi
rpi = raspi();
cam = cameraboard(rpi,'Resolution','640x480');
```

The cam is a handle to a cameraboard object. Let's display the images captured from Raspberry Pi Camera Board in MATLAB.

```
for i = 1:100
    img = snapshot(cam);
    image(img);
    drawnow;
end
```

Let's try a simple image inversion algorithm on the images captured from Raspberry Pi Camera Board.

```
figure(1);
for i = 1:100
   img = snapshot(cam);
   img = 255 - img;
   image(img);
   drawnow;
end
```

The image inversion creates a color negative effect. The Raspberry Pi Camera Board itself can invert the images by setting ImageEffect property to 'negative'.

```
figure(1);
cam.ImageEffect = 'negative';
for i = 1:100
   img = snapshot(cam);
   image(img);
   drawnow;
end
```

Here is a more interesting image effect.

```
figure(1);
cam.ImageEffect = 'sketch';
for i = 1:100
   img = snapshot(cam);
   image(img);
   drawnow;
```

end

CHAPTER-4

IMPLEMENTATION

IV . IMPLEMENTATION

IV.1.Software installation:

The only software required for our project is: RASPBERRY PI

And here we are going to explain about, how to install Raspberry pi into your system.

IV.1 Installation:

Step-1: Connection of "Raspberry pi to camera and system".

Figure 4.1.1

Step-2: click on the programming on the system display.

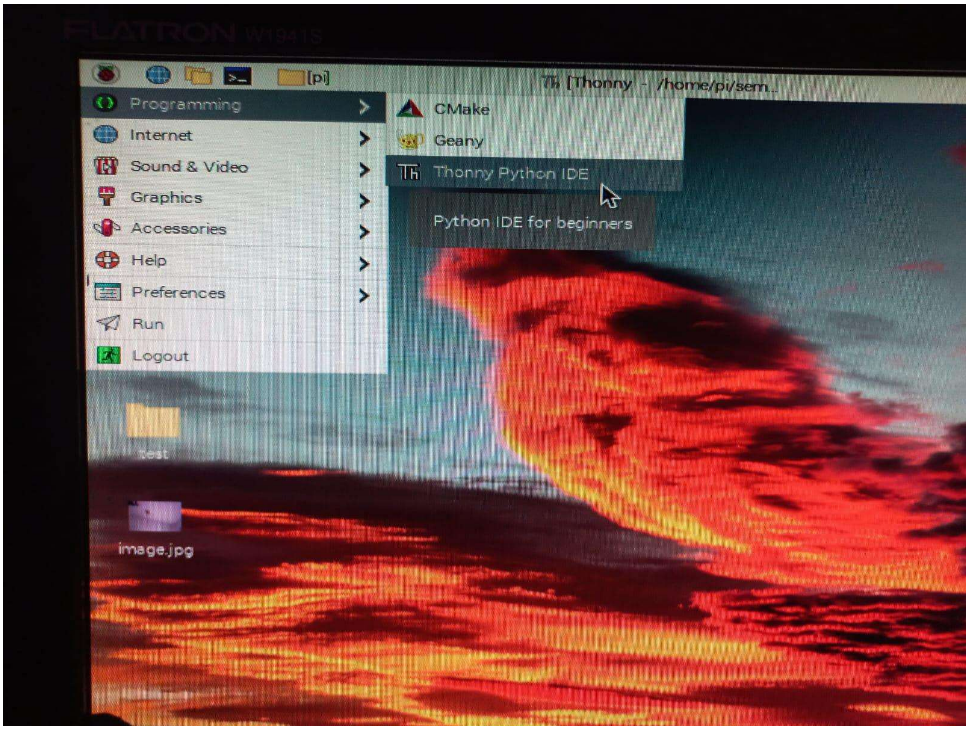


Figure 4.1.2

Step-3: Input the code from the file.

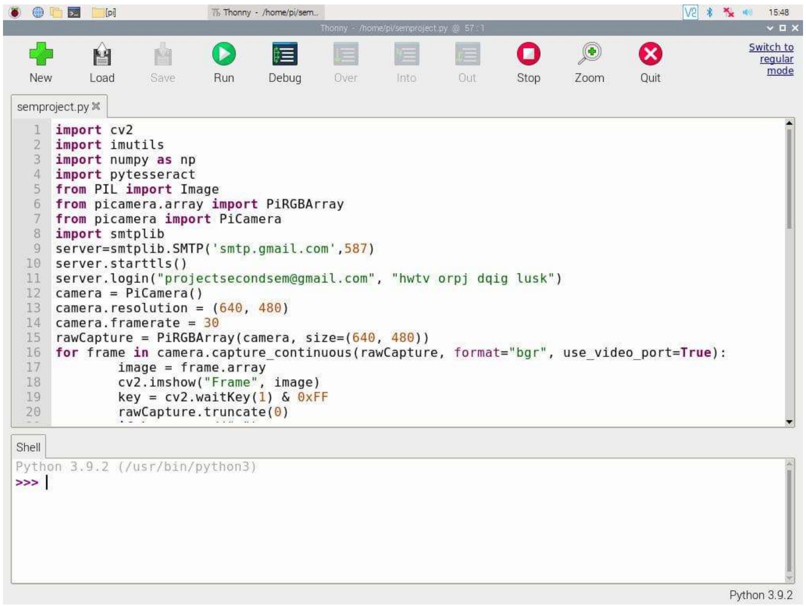


Figure 4.1.3

Step-4: The number plate has been detected and the code is running.

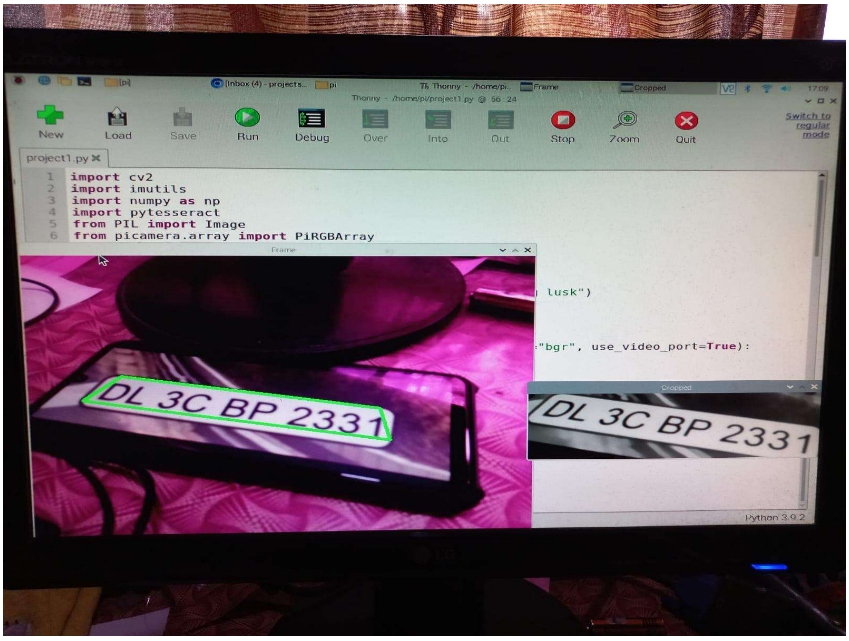


Figure 4.1.4

Step-5: After the number plate is recognized the output is displayed.

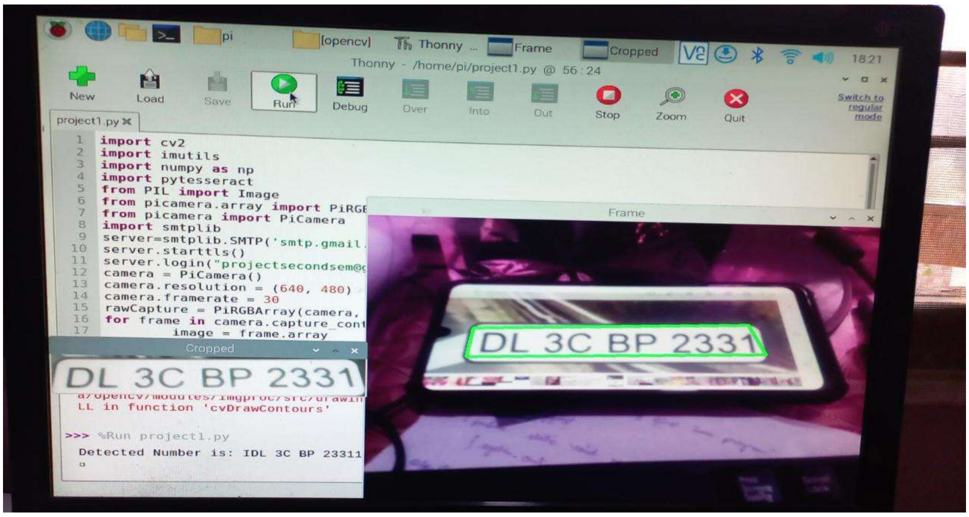


Figure 4.1.5

IV.2MATERIALS AND METHOD

IV.3Components:

The components that we have used to done this project and the purposes of each component are listed below:
And we are explaining about the connection between the components.

RASPBERRY PI:

The Raspberry pi development board is shown in Figure. Raspberry Pi (/paɪ/) is a series of small single-board computers (SBCs) developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom.[14] The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries.[15][16][17] The original model became more popular than anticipated,[18] selling outside its target market for uses such as robotics. It is widely used in many areas, such as for weather monitoring,[19] because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of the HDMI and USB standards

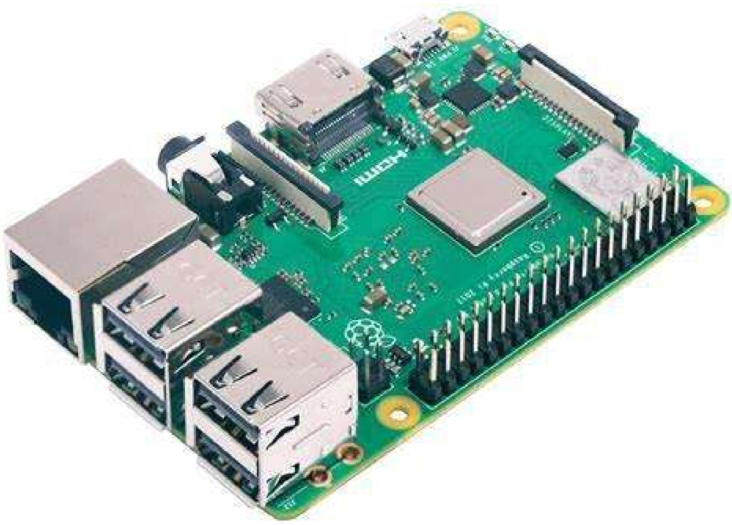


Figure 4.2.1

PI CAMERA V2 MODULE :

The Raspberry Pi Camera Module 2 replaced the original Camera Module in April 2016. The v2 Camera Module has a Sony IMX219 8-megapixel sensor (compared to the 5-megapixel OmniVision OV5647 sensor of the original camera).

The Camera Module 2 can be used to take high-definition video, as well as stills photographs. It’s easy to use for beginners, but has plenty to offer advanced users if you’re looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion, and other video cleverness. You can also use the libraries we bundle with the camera to create effects.

You can read all the gory details about IMX219 and the Exmor R back-illuminated sensor architecture on Sony’s website, but suffice to say this is more than just a resolution upgrade: it’s a leap forward in image quality, colour fidelity, and low-light performance. It supports 1080p30, 720p60 and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi.

The camera works with all models of Raspberry Pi 1, 2, 3 and 4. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Pi camera Python library. See the Getting Started with Pi camera resource to learn how to use it.

All models of Raspberry Pi Zero require a Raspberry Pi Zero camera cable; the standard cable supplied with the camera module is not compatible with the Raspberry Pi Zero camera connector. Suitable cables are available at low cost from many Raspberry Pi Approved Resellers, and are supplied with the Raspberry Pi Zero Case.

The camera module is very popular in-home security applications, and in wildlife camera traps.

Figure 4.2.2

HDMI CABLE:

HDMI cable is a boon to having neat interference and a cut-down web of wires when it comes to transmitting high-definition audio and video signals and used to connect the monitor and raspberry pi

Figure 4.2.3

CHAPTER-5

CODE

CODE

```
import cv2
import imutils
import numpy as np
import pytesseract
from PIL import Image
from picamera.array import PiRGBArray
from picamera import PiCamera
import smtplib
server=smtplib.SMTP('smtp.gmail.com',587)
server.starttls()
server.login("Your Email ID", "Email ID Password")
camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 30
rawCapture = PiRGBArray(camera, size=(640, 480))
for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
        image = frame.array
        cv2.imshow("Frame", image)
        key = cv2.waitKey(1) & 0xFF
        rawCapture.truncate(0)
        if key == ord("s"):
             gray = cv2.cvtColor(image, 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
             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
             for c in cnts:
                peri = cv2.arcLength(c, True)
                approx = cv2.approxPolyDP(c, 0.018 * peri, True)
                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(image, [screenCnt], -1, (0, 255, 0), 3)
mask = np.zeros(gray.shape,np.uint8)
new_image = cv2.drawContours(mask,[screenCnt],0,255,-1,)
new_image = cv2.bitwise_and(image,image,mask=mask)
(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]
text = pytesseract.image_to_string(Cropped, config='--psm 11')
print("Detected Number is:",text)
server.sendmail("Sender's Email ID@gmail.com","Sender's Email ID@gmail.com",text)
cv2.imshow("Frame", image)
cv2.imshow('Cropped',Cropped)
cv2.waitKey(0)
break
cv2.destroyAllWindows()
```

CHAPTER-6
RESULT & CONCLUSION

VI.1- RESULT

The Raspberry Pi camera sensor is able to detect the number plate from the vehicle. The data is stored for total number of vehicles which are entering. The system which includes raspberry pi and system along with camera are sensitive to sense low emissions of the gases like ammonia and methane emitted due to spoilage of food article. The level of gases emitted will vary depending on the decay of food. The detection of these gases can be used to control the decay of food. The humidity sensors can be employed to sense the humidity content in the food. Various other sensors like temperature, pressure, moisture etc can also be used to detect bad virus and bacterial growth in the food.

VI.2- CONCLUSION

This project mainly performs four tasks. The first task is to input an image of the car and this will happen with the help of webcam of the computer for the prototype. When the image is fed the image is enhanced in quality. The enhancement is done in the resolution and the thresholding. The image is constraint to a fixed image frame size. After the enhancement the image is processed to segment the number plate from the full picture based on the mathematical model of the rectangle. The segmented number plate is then processed for ocr.To segment all the characters in the picture in the form of text and then it can be stored in a database or can be displayed as in this prototype. The project is designed so that we can understand a technology used in nowadays. Automatic license plate system and ocr systems used in most of the developed countries like Germany ,France,singapore,japan,etc.
It is seen that security forces all over the world face problem to locate or register vehicle number.

CHAPTER-7

FUTURE ENHANCEMENT

VII . FUTURE ENHANCEMENT

As a future work the developed system would be concentrated upon increasing the accuracy of text localization and graphics removal in caption text images. It can be evaluated using various other available image data bases and using various other classifiers. The methods can be further improvised and applied for automatic mixed mail sorting.

CHAPTER-8

REFERENCES

VIII. REFERENCES

- https://www.marketsandmarkets.com/Market-Reports/anpr-system-market-140920103.html
- https://opensource.google/projects/tesseract
- https://www.linuxjournal.com/article/9676
- https://pdfs.semanticscholar.org/bdca/d2b56e3a38ef543f6fb0a602deb5f453493b.pdf?_ga=2.28647292.1514298175.1598968225-502553827.1598968225
- https://pdfs.semanticscholar.org/4d31/46d2b4bf23558ec0baf93506be5b96437fc2.pdf
- https://www.researchgate.net/publication/299858935_Proposal_for_Automatic_License_and_Number_Plate_Recognition_System_for_Vehicle_Identification
- https://opencv.org/#
- https://pypi.org/project/Pillow/
- https://www.python.org/about/gettingstarted/
- https://numpy.org/learn 50

Thank you