A

Mini Project

On

ROAD SIGN RECOGNITION SYSTEM FOR

SEMI-AUTOMATED VEHICLES USING RASPBERRY PI

**(Submitted in partial fulfillment of the requirements for the award of Degree)**

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

CHAPPIDI GAYATHRI (197R1A0572)

K. ANITHA KUMARI (197R1A0582)

Y. SUMAGNA (197R1A05B9)

Under the Guidance of

TABEEN FATIMA

(Assistant Professor)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

**(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya(V), Medchal Road, Hyderabad-501401)**

2019-2023

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project entitled **“ROAD SIGN RECOGNITION SYSTEM FOR SEMI-AUTONOMOUS VEHICLES USING RASPBERRY Pi”** being submitted by **CHAPPIDI GAYATHRI (197R1A0572), K ANITHA KUMARI (197R1A0582), Y.SUMAGNA (197R1A05B9)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

|  |  |
| --- | --- |
| **Ms. TABEEN FATIMA** | **Dr. A. RAJI REDDY** |
| **(Assistant Professor)**  **INTERNAL GUIDE** | **DIRECTOR** |
| **Dr. K. SRUJAN RAJU**  **HOD-CSE** | **EXTERNAL EXAMINER** |

**Submitted for viva voice Examination held on \_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to our guide **Ms. Tabeen Fatima**, Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr**. **Punyaban Patel, Ms. Shilpa, Dr. M. Subha Mastan Rao & J. Narasimha Rao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful **to Dr. K. Srujan Raju**, Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to **Sri.** **Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

|  |  |
| --- | --- |
| **CHAPPIDI GAYATHRI** | **(197R1A0572)** |
| **K. ANITHA KUMARI** | **(197R1A0582)** |

**Y. SUMAGNA (197R1A05B9)**

**ABSTRACT**

Road Safety Statistics demonstrate that about 1.24 million people die annually on road due to accidents. The objective of this project is to introduce a method of road sign detection using Raspberry pi for Self- driving car. It enhances safety by informing the drivers about the current state of traffic signs on the road and giving valuable information about precaution. Road sign recognition is one of the important tasks of intelligent transportation systems (ITS). The Digital image processing play important role in the sign capturing and detection system. The image processing algorithms to takes the necessary action for resizing the captured signs. The Raspberry pi camera port used to capturing the road signs with image enhancement techniques.

**LIST OF FIGURES/TABLES**

**FIGURE NO FIGURE NAME PAGE NO**

Figure 4.1 PROECT BLOCK DIAGRAM 11

Figure 4.1 Raspberry Pi 3 Model B 12

Figure 4.2 Model B 12

Figure 4.3 USB Web Camera 14

Figure 4.4 USB Cables 14

**SCREENSHOT NO. SCREENSHOT NAME PAGENO**

Screenshot 4.1 Project Block Diagram 11

Screenshot 4.2 Raspberry Pi Model 3 12

Screenshot 4.3 Circuit Diagram of Raspberry Pi 14

Screenshot 4.4 USB Cables 15

Screenshot 4.5 Use case Diagram 16

Screenshot 4.6 Class Diagram 17

Screenshot 4.7 Sequence Diagram 18

Screenshot 4.8 Activity Diagram 19

Screenshot 6.1 Road Sign Detected 24

Screenshot 6.2 Road Sign Detected 24

Screenshot 6.3 Instrument Diagram 25

**TABLE OF CONTENT**

**ABSTRACT i**

**LIST OF FIGURES ii**

**LIST OF SCREENSHOTS iii**

1. **INTRODUCTION 1**

**1.1.** OBJECTIVE **1**

1. **EMBEDDED SYSTEM 2**
   1. INTRODUCTION TO EMBEDDED SYSTEM **2**

**2.2.** FEATURES OF EMBEDDED SYSTEM **2**

**2.3.** LITERATURE SURVEY **3**

**2.4**. CHARACTERISTIS OF EMBEDDED SYSTEM **4**

**2.5.** REAL TIME SYSTEMS **7**

1. **SYSTEM ANAYLSIS 9**

**3.1.** EXISTING SYSTEM **9**

**3.1.1.** DISADVANTAGES **9**

**3.2.** PROPOSED SYSTEM **9**

**3.2.1.** ADVANTAGES **10**

**3.3.** HARDWARE & SOFTWARE REQUIREMENTS **10**

**4. ARCHITECTURE 11**

**4.1.** PROJECT ARCHITECTURE **11**

**4.2.** USE CASE DIAGRAM **16**

**4.3.** CLASS DIAGRAM **17**

**4.4** SEQUENCE DIAGRAM **18**

**4.5** ACTIVITY DIAGRAM **19**

**5. IMPLEMENTATION 20**

**5.1.** SAMPLE CODE **20**

**6. SCREENSHOTS 25**

**7. CONCLUSION & FUTURE SCOPE 27**

**7.1.** CONCLUSION **27**

**7.2.** FUTURE SCOPE **27**

**8. BIBLIOGRAPHY 28**

**8.1.** REFERENCE **28**

**8.2.** GITHUB LINK **29**

**1. INTRODUCTION**

The field of traffic sign recognition is not very old with the first paper on the topic published in Japan in 1984 when the aim was to try computer vision methods for the detection of objects. Since then however, the field has continued to expand at an increasing rate. Traffic sign recognition is used to maintain traffic signs, warn the distracted driver, and prevent his/her actions that can lead an accident. A real-time automatic speed sign detection and recognition can help the driver, significantly increasing his/her safety. It enhances safety by informing the drivers about the current state of traffic signs on the road and giving valuable information about precaution. Road signs are placed either at the sides of road or above as navigation guides. With continuous raise in road traffic, mishaps chances likewise raises. Road safety statistics demonstrate that about 1.24 million people die annually on road due to accidents. Thus, the research has focused in recent years on intelligent systems that can avoid the collisions and mishaps. They are developed to enhance Road safety and driving comfort. This system introduces a method of road detection using Raspberry Pi for self-driving car. When vehicles are moving on the road, it is difficult to find road sign such as lane, crosswalk, stop line, turn left, turn right etc., only using a single camera. A web camera mounted on the moving prototype vehicle captures the image and processes the image using Haar Cascade Classifier. Several image of road signs are collected, trained and stored as databases in the memory of the Raspberry pi.

**1.1 OBJECTIVE**

Road Safety Statistics demonstrate that about 1.24 million people die annually on road due to accidents. The objective of this project is to introduce a method of road sign detection using Raspberry pi for Self-driving car. It enhances safety by informing the drivers about the current state of traffic signs on the road and giving valuable information about precaution.

**2. EMBEDDED SYSTEM**

**2.1. INTRODUCTION TO EMBEDDED SYSTEMS**

Our day-to-day life is becoming more and more dependent on "Embedded Systems and Digital Technologies. Embedded technologies are bonding into our daily activities even without our knowledge. We know the fact that the refrigerator, washing machine, microwave oven, air conditioner, television, DVD players, and music systems that we use in our home are built around an embedded system. It is nothing but an intelligent embedded system. In your vehicle itself the presence of specialized embedded systems vary from intelligent head lamp controllers, engine controllers and ignition control Systems to complex air bag control systems to protect you from a severe accident.

People experience the power of embedded systems and enjoy the features and comfort provided by them. Most of us are totally unaware or ignorant of the intelligent embedded systems giving us so much comfort and security. Embedded systems are like reliable servants- they don't like to reveal their identity and neither they complaint about their workloads to their owners or bosses. They are always working behind the scenes and are dedicated to their assigned task till their last breath. This book gives you an overview of embedded systems, the various steps involved in their design and development and the major domains where they are deployed.

**2.2. FEATURES OF EMBEDDED SYSTEMS**

The versatility of the embedded computer system lends itself to utility in all kinds of enterprises, from the simplification of deliverable products to a reduction in costs in their development and manufacture. Usually a low power consumption CPU with a limited amount of memory is used in embedded systems. Complex systems with rich functionality employ special operating systems that take into account major characteristics of embedded systems. Embedded operating systems have minimized footprint and may follow real-time operating system specifics. The special computers system is usually less powerful than general purpose systems, although some expectations do exist where embedded systems are very powerful and complicated. Usually a low power consumption CPU with a limited amount of memory is used in embedded systems. Some embedded systems have to operate in extreme environment conditions such as very high temperature & humidity.

Many embedded systems use very small operating systems; most of these provide very limited operating system capabilities. Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance some embedded systems are mass-produced, benefiting from economies of scale. Some embedded systems have to operate in extreme environment conditions such as very high temperature & humidity. For high volume systems such as portable music players or mobile phones, minimizing cost is usually the primary design consideration. Engineers typically select hardware that is just "good enough" to implement the necessary functions. For low volume or prototype embedded systems, general purpose computers may be adapted by limiting the programs or by `replacing the operating system with a real-time operating system.

**2.3. LITERATURE SURVEY**

Given the definition of embedded systems earlier is this chapter; the first such systems could not possibly have appeared before 1971. That was the year Intel introduced the world's first microprocessor. This chip, the 4004, was designed for use in a line of business calculators produced by the Japanese Company Busicom. In 1969, Busicom asked Intel to design a set of custom integrated circuits-one for each of their new calculator models. The 4004 was Intel's response rather than design custom hardware for each calculator, Intel proposed a general- purpose circuit that could be used throughout the entire line of calculators. Intel's idea was that the software would give each calculator its unique set of features. The microcontroller was an overnight success, and its use increased steadily over the next decade. Early embedded applications included unmanned space probes, computerized traffic lights, and aircraft flight control systems. In the 1980s, embedded systems quietly rode the waves of the microcomputer age and brought microprocessors into every part of our kitchens (bread machines, food processors, and microwave ovens), living rooms (televisions, stereos, and remote controls), and workplaces (fax machines, pagers, laser printers, cash registers, and credit card readers).

It seems inevitable hat the number of embedded systems will continue to increase rapidly. Already there are promising new embedded devices that have enormous market potential; light switches and thermostats that can be central computer, intelligent air-bag systems that don't inflate when children or small adults are present, pal-sized electronic organizers and personal digital assistants (PDAs), digital cameras, and dashboard navigation systems. Clearly, individuals who possess the skills and desire to design the next generation of embedded systems will be in demand for quite some time

**2.4. CHARACTERISTICS OF EMBEDDED SYSTEMS**

Embedded computing systems generally exhibit rich functionality complex functionality is usually the reason for introducing cups into the design. However, they also exhibit many non- functional requirements that make the task especially challenging:

* Real-time deadlines that will cause system failure if not met.
* Multi-rate operation.
* In many cases, low power consumption.
* Low manufacturing cost, which often means limited code size.

They may consider the performance characteristics of a few computational kernels of their software, but rarely analyze the total application. They never consider power consumption and manufacturing cost. The need to juggle all these requirements makes embedded system programming very challenging and is the reason embedded system designers need to understand computer architecture.

**TYPES OF EMBEDDED SYSTEM**

Based on functionality and performance embedded systems categorized as 4 types

1. Standalone embedded systems.

2. Real time embedded systems.

3. Networked information appliances.

4. Mobile devices.

1. **STANDALONE EMBEDDED SYSTEM:** As the name implies, standalone systems work in standalone mode. They take i/p. Process them and produce the desire o/p. The i/p can be an electrical signal from transducer or temperature signal or commands from human being. The o/p can be electrical signal to drive another system an led or LCD display.

Ex digital camera, microwave oven, CD player, air conditioner etc.

1. **REAL TIME EMBEDDED SYSTEM**: In this type of an embedded system a specific work has to be complete in a particular period of time.

Hard real time systems: - Embedded real time used in missiles.

Soft real time systems: - DVD players.

**3**. **NETWORKED INFORMATION APPLIANCES:** Embedded systems that are provided with n/w interfaces and accessed by n/w such as local area n/w or internet are called network information appliances.

EX. a web camera is connected to the internet. Camera can send pictures in real time to any computers connected to the internet.

**4. MOBLIE DEVICES:** Actually it is a combination of both VLSI and Embedded system Mobile devices such as mobile phone, personal digital assistants, smart phones etc. Are special category of embedded system. Embedded systems are based on the concept of the micro-controller, a single integrated circuit that contains all the technology required to run an application. **Road Sign Recognition system for Autonomous vehicle using Raspberry Pi**

* Central processing Unit.
* Input/output interfaces (such as serial ports).
* Peripherals (such as timers).
* ROM ,EEPROM (or) Flash memory for program storage.
* RAM for data storage Clock generator.

By integrating all of these features into a single chip it is possible to greatly reduce the number of chips and wiring necessary to control an electronic device, dramatically reducing its complexity, size and cost.

* **SIZE AND WEIGHT**: Micro-controllers are designed to deliver maximum performance for Minimum size and weight. A centralized on-board computer system would greatly outweigh a collection of microcontrollers.
* **EFFICIENCY**: Microcontrollers are designed to perform repeated functions for long periods of time without failing or requiring service
* **MICRO CONTROLLER**: It is a chip through which we can connect many other devices and also those are controlled by the program the program which burn into it. Here we are using Arduino as the main device to control.

**2.5. REAL TIME SYSTEMS**

One subclass of embedded is worthy of an introduction at this point. As commonly defined, a real- time system is a computer system that has timing constraints. In other words, a real-time system is partly specified in terms of its ability to make certain calculations or decisions in a timely manner. These important calculations are said to have deadlines for completion. And, for all practical purposes, a missed deadline is just as bad as a wrong answer. The issue of what if a deadline is missed is a crucial one. For example, if the real-time system is part of an airplane's flight control system, it is possible for the lives of the passengers and crew to be endangered by a single missed deadline. However, if instead the system is involved in satellite communication, the damage could be limited to a single corrupt data packet. The more severe the consequences, the more likely it will be said that the deadline is "hard" and thus, the system is a hard real-time system. Real-time systems at the other end of this discussion

are said to have "soft" deadlines.

All of the topics and examples presented in this book are applicable to the designers of real-time system who is more delight in his work. He must guarantee

reliable operation of the software and hardware under all the possible conditions and to the degree that human lives depend upon three system's proper execution, engineering calculations and descriptive paperwork.

* **APPLICATION AREAS:** Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, Data communication, telecommunications, transportation, military and so on.
* **CONSUMER APPLIANCES:** At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air- conditioner, VCO player, video game consoles, video recorders etc. Today’s high-tech car has about 20 embedded systems for transmission control, engine spark control, air-conditioning, navigation etc. Even wristwatches are now becoming embedded systems. The palmtops are powerful embedded systems using which we can carry out many general-purpose tasks such as playing games and word processing
* **OFFICE AUTOMATION:** The office automation products using em embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.
* **INDUSTRIAL AUTOMATION:** Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs.

**3. SYSTEM ANALYSIS**

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

**3.1 EXISTING SYSTEM**

Classifying, traffic signs is a very important task for autonomous driving systems as the safety of everyone as well as the passenger depends on it. Depending on the country, traffic signs possess a variety in their visual appearance making it harder for classification systems to succeed. Nowadays, Intelligent Autonomous Vehicles together with Advanced Driver Assistance Systems

**3.1.1. DISADVANTAGES OF EXISTING SYSTEM**

* Delay in the quick movement of the vehicle
* No camera based sign detection
* System with no indicator with controller application

#### **3.2. PROPOSED SYSTEM**

To design a good recognition system, the system needs to have a good discriminative power and a low computational cost. The system should be robust to the changes in the geometry of sign (such as vertical or horizontal orientation) and to image noise in general. Next the recognition should be started quickly in order to keep the balanced flow in the pipeline of Raspberry Pi allowing for processing of data in real time. Finally, the optical character recognition engine must be able to interpret a pre-processed image into a text file.

**3.2.1. ADVANTAGES OF THE PROPOSED SYSTEM**

* Image signal detection
* Indicating through display
* Traffic assistance driving systems
* Avoiding obstacles
* Automatic driving systems

#### **3.3. HARDWARE & SOFTWARE REQUIREMENTS**

#### **3.3.1. HARDWARE REQUIREMENTS**

* Web Camera
* LCD Display
* DC Motor

**3.3.2. SOFTWARE REQUIREMENTS**

* Ubuntu 16.4 or higher version
* Python
* OpenCV

#### **4**. **ARCHITECTURE**

**4.1 PROJECT ARCHITECTURE**

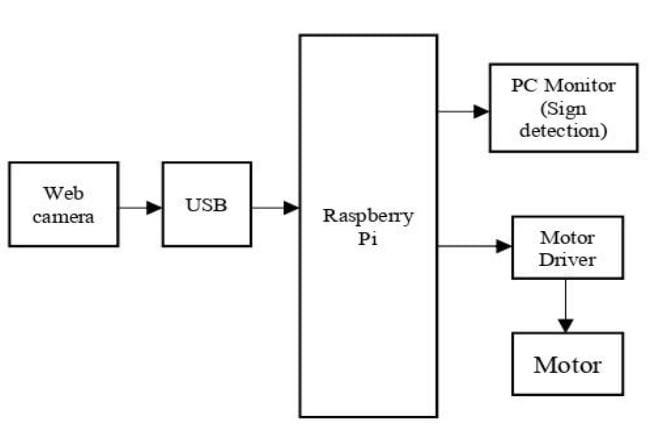


Figure 4.1 PROECT BLOCK DIAGRAM

**DC MOTOR**:

A DC motor is any of a class of rotary electrical motors that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor DC Motor L293D: Connect 5V to Enable 1 , Vss , and Vs on the L293D. Connect digital output pins (we're using 6 and 7) to input 1 and input 2 on the L293D. Connect your Arduino's GND to both GND pins on the same side of the L293D. Finally, connect output 1 and output 2 of the L293D to your motor pins.

**RASPBERRY PI**

The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi

Model B+ and Raspberry Pi 2 Model B.

Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processer, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs

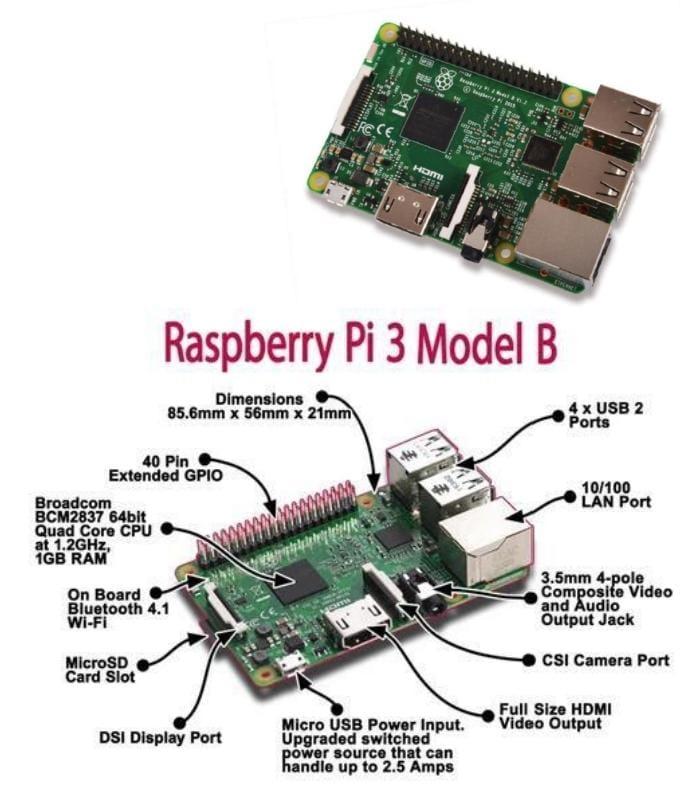
****

Figure 4.2 Raspberry Pi

**MODEL B**

**RASPBERRY PI SPECIFICATIONS OVERVIEW:**

Processor: Broadcom BCM2835

CPU: ARM

1176JZFS (ARM 11

w/v6core,floatingpt

@ 700MHz)

RAM: 512 MB

USB: 2 USB 2.0

Network: Ethernet

Video out: HDMI

Audio out: 3.5 mm jack

SD Card Storage (Up to 32GB)

Micro USB power

Display Serial Interface Port (DSI)

Camera Serial Interface Port (CSI)

In April 2012, the original Raspberry Pi, also known as the model B, was launched. It offered an ARMv6 700 MHz single-core processor, a Video Core IV GPU and 512MB of RAM, all packed in a credit card size board.

The model B supported 26-pin GPIO and a SD card slot for loading the operating system. Priced at $35 per unit

**USB WEBCAM:**

Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires much [bandwidth,](https://en.wikipedia.org/wiki/Bandwidth_(computing)) such streams usually use [compressed formats.](https://en.wikipedia.org/wiki/Video_compression)

**Universal serial bus**, **USB** (pronounced yoo-es-bee) is a [plug and play](https://www.computerhope.com/jargon/p/pnp.htm) interface that allows a computer to communicate with [peripheral](https://www.computerhope.com/jargon/p/peripher.htm) and other devices. USB-connected devices cover a broad range; anything from keyboards and mice, to music players and flash drives. USB may also be used to send power to certain devices, such as powering smartphones and tablets and charging their batteries

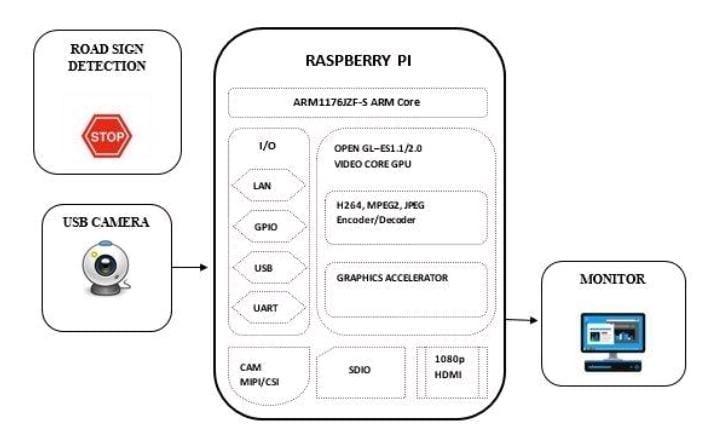


Figure 4.3 Circuit Diagram of Raspberry Pi

**USB CONNECTOR TYPES**

USB connectors come in different shapes and sizes. Most of the versions of USB connectors, including the standard USB, Mini USB, and Micro USB, have two or more variations of connectors. Further information on each type is provided below.



Figure 4.4 USB Cables

**CAMERA**

A **webcam** is a [video camera](https://en.wikipedia.org/wiki/Video_camera) that feeds or [streams](https://en.wikipedia.org/wiki/Streaming_media) an image or video in real time to or through a [computer network,](https://en.wikipedia.org/wiki/Computer_network) such as the [Internet.](https://en.wikipedia.org/wiki/Internet) Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware.Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires much [bandwidth,](https://en.wikipedia.org/wiki/Bandwidth_(computing)) such streams usually use [compressed formats.](https://en.wikipedia.org/wiki/Video_compression) The maximum resolution of a webcam is also lower than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions.

The term "webcam" (a [clipped compound)](https://en.wikipedia.org/wiki/Clipped_compound) may also be used in its original sense of a [video camera](https://en.wikipedia.org/wiki/Video_camera) connected to the [Web](https://en.wikipedia.org/wiki/World_Wide_Web) continuously for an indefinite time, rather than for a particular session, generally supplying a view for anyone who visits its [web page](https://en.wikipedia.org/wiki/Web_page) over the Internet. Some of them, for example, those used as online [traffic cameras,](https://en.wikipedia.org/wiki/Traffic_camera) are expensive, rugged [professional video cameras.](https://en.wikipedia.org/wiki/Professional_video_camera)

**Mini-USB**, also known as **mini-B**, is used with [digital cameras](https://www.computerhope.com/jargon/d/digicame.htm) and computer [peripherals.](https://www.computerhope.com/jargon/p/peripher.htm) Mini-USB has largely been replaced by Micro-USB and USB-C cables on newer devices.

**Micro-USB**, announced in [2007,](https://www.computerhope.com/history/2007.htm) was designed to replace mini-USB. The two varieties of Micro-USB are **Micro-A** and **Micro-B**, both featuring a connector size of 6.85 x 1.8 mm, although Micro-A connectors feature a larger maximum overmold size. Micro-USB cables are often used to connect computer peripherals, video game [controllers,](https://www.computerhope.com/jargon/c/controll.htm) and for charging [smartphones.](https://www.computerhope.com/jargon/s/smartphone.htm) While many companies are upgrading to USB type-C connectors (next section), Micro-USB is still commonly used with electronic devices

**4.2 USE CASE DIAGRAM**

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures

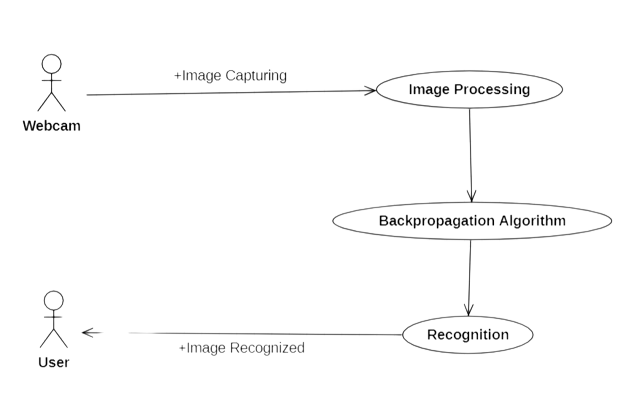


Figure 4.2: Use Case Diagram for Road Sign Recognition System

#### **4.3 CLASS DIAGRAM**

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods), and the relationships among objects

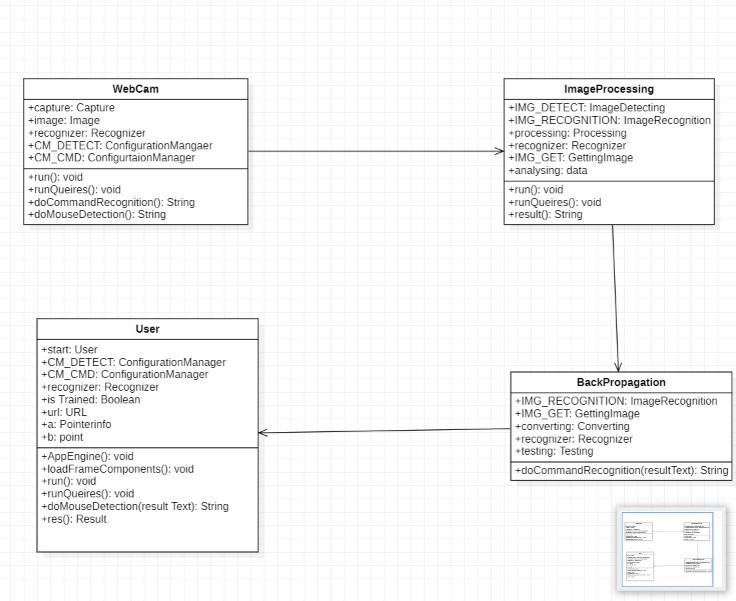
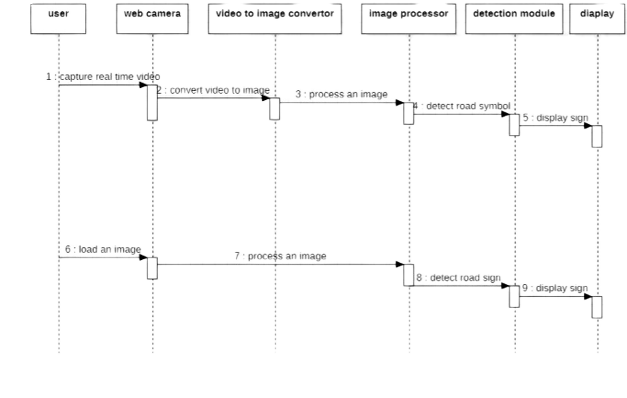


Figure 4.3 Class Diagram for Road Sign Recognition System

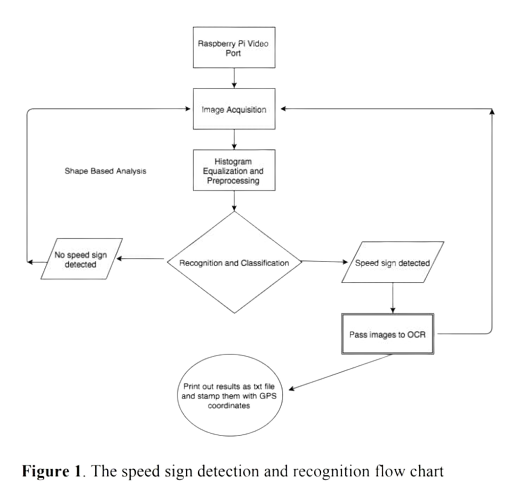
#### **4.4 SEQUENCE DIAGRAM**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development

 Figure 4.4 Sequence Diagram for Road Sign Recognition System

#### **4.5 ACTIVITY DIAGRAM**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores

Figure 4.5 Activity Diagram for Road Sign Recognition System

Message Enhancement

#### **5. IMPLEMENTATION**

#### **5.1.SAMPLE CODE**

import cv2

import numpy as np

from scipy.stats import itemfreq

import RPi.GPIO as GPIO

import time

import os, signal

GPIO.setmode(GPIO.BOARD)

GPIO.setup(11,GPIO.OUT, initial = 0)

GPIO.setup(12,GPIO.OUT, initial = 0)

GPIO.setup(13,GPIO.OUT, initial = 0)

GPIO.setup(15,GPIO.OUT, initial = 0)

#set GPIO Pins

GPIO\_TRIGGER = 38

GPIO\_ECHO = 40

#set GPIO direction (IN / OUT)

GPIO.setup(GPIO\_TRIGGER, GPIO.OUT)

GPIO.setup(GPIO\_ECHO, GPIO.IN)

def stoprobo():

GPIO.output(11,GPIO.LOW)

SS GPIO.output(12,GPIO.LOW)

GPIO.output(13,GPIO.LOW)

GPIO.output(15,GPIO.LOW)

defforwardrobo():

GPIO.output(11,GPIO.LOW)

GPIO.output(12,GPIO.LOW)

GPIO.output(13,GPIO.LOW)

GPIO.output(15,GPIO.LOW)

def leftrobo():

GPIO.output(11,GPIO.HIGH)

GPIO.output(12,GPIO.LOW)

GPIO.output(13,GPIO.LOW)

GPIO.output(15,GPIO.HIGH)

def rightrobo():

GPIO.output(11,GPIO.LOW)

GPIO.output(12,GPIO.HIGH)

GPIO.output(13,GPIO.HIGH)

GPIO.output(15,GPIO.LOW)

def rightrobouturn():

GPIO.output(11,GPIO.LOW)

GPIO.output(12,GPIO.HIGH)

GPIO.output(13,GPIO.HIGH)

GPIO.output(15,GPIO.LOW)

def get\_dominant\_color(image, n\_colors):

pixels = np.float32(image).reshape((-1,3))

criteria=(cv2.TERM\_CRITERIA\_EPS+cv2.TERM\_CRITERIA\_MAX\_ITER, 200, .1)

flags = cv2.KMEANS\_RANDOM\_CENTERS

flags = cv2.KMEANS\_RANDOM\_CENTERS

flags, labels, centroids = cv2.kmeans( pixels, n\_colors, None, criteria, 10, flags) palette = np.uint8(centroids)

return palette[np.argmax(itemfreq(labels)[:, -1])]

clicked=False

def onMouse(event,x, y, flags, param):

global clicked

if event == cv2.EVENT\_LBUTTONUP:

clicked = True

cameraCapture = cv2.VideoCapture(0)

cv2.namedWindow('ROAD SIGN DETECTION')

cv2.setMouseCallback('ROAD SIGN DETECTION', onMouse)

# Read and process frames in loop

success, frame = cameraCapture.read()

stoprobo()

while success and not clicked:

cv2.waitKey(1)

success, frame = cameraCapture.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

#cv2.imshow('grey\_image',gray)

img = cv2.medianBlur(gray, 37)

#cv2.imshow('median\_blurred\_image',img)

circles = cv2.HoughCircles(img, cv2.HOUGH\_GRADIENT, 1, 50, param1=120, param2=40)

if not circles is None:

circles = np.uint16(np.around(circles))

max\_r, max\_i = 0, 0

for i in range(len(circles[:, :, 2][0])):

if circles[:, :, 2][0][i] > 50 and circles[:, :, 2][0][i] > max\_r:

max\_i = i

max\_r = circles[:, :, 2][0][i]

x, y, r = circles[:, :, :][0][max\_i]

if y > r and x > r:

square = frame[y-r:y+r, x-r:x+r]

dominant\_color = get t\_dominant\_color(square, 2)

if dominant\_color[2] > 100:

print(" U TURN ")

rightrobouturn()

elif dominant\_color[0] > 80:

zone\_0=square[square.shape[0]\*3//8:square.shape[0]

\*5//8,square.shape[1]\*1//8:square.shape[1]\*3//8]

zone\_0\_color = get\_dominant\_color(zone\_0, 1)

zone\_1 = square[square.shape[0]\*1//8:square.shape[0]

\* 3//8, square.shape[1]\*3//8:square.shape[1]\*5//8]

cv2.imshow('zone\_1',zone\_1)

zone\_1\_color = get\_dominant\_color(zone\_1, 1)

zone\_2 = square[square.shape[0]\*3//8:square.shape[0]

\* 5//8, square.shape[1]\*5//8:square.shape[1]\*7//8]

cv2.imshow('zone\_2',zone\_2)

zone\_2\_color = get\_dominant\_color(zone\_2, 1)

if zone\_1\_color[2] < 60:

if sum(zone\_0\_color) > sum(zone\_2\_color):

print("LEFT")

leftrobo()

time.sleep(5)

forwardrobo()

# else:

print("RIGHT")

rightrobo()

time.sleep(5)

forwardrobo()

else:

if sum(zone\_1\_color) > sum(zone\_0\_color) and sum(zone\_1\_color) > sum(zone\_2\_color):

print("FORWARD")

forwardrobo()

elif sum(zone\_0\_color) > sum(zone\_2\_color):

print("LEFT")

leftrobo()

time.sleep(5)

forwardrobo()

else:

print("RIGHT")

rightrobo()

time.sleep(5)

forwardrobo()

else:

print("STOP")

stoprobo()

for i in circles[0, :]:

cv2.circle(frame, (i[0], i[1]), i[2], (0, 255, 0), 2)

cv2.circle(frame, (i[0], i[1]), 2, (0, 0, 255), 3)

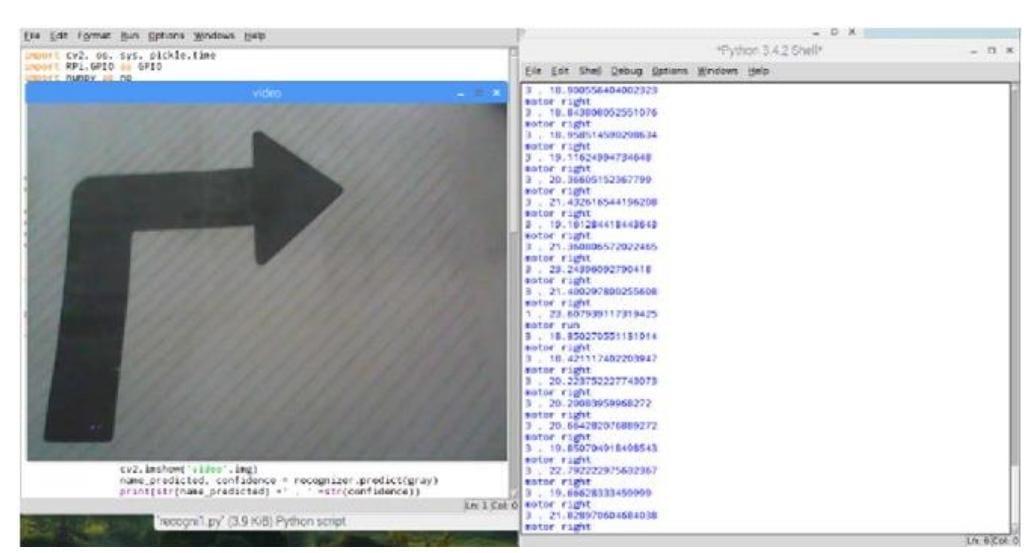
cv2.imshow('ROAD SIGN DETECTION', frame)

cv2.destroyAllWindows() cameraCapture.release()

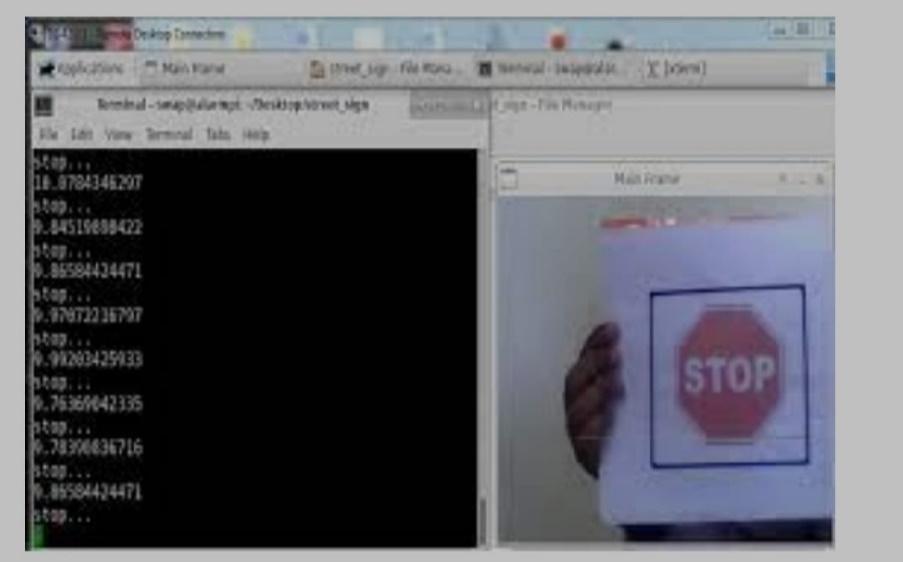
GPIO.cleanup()

#### **6. SCREENSHOTS**

**6.1 RESULT SCREENSHOT**



5.1 Road Sign Detection



5.2 Road Sign Recognition



Figure 5.3 Instrument Used For Example

#### **7. CONCLUSION**

#### **7.1 PROJECT CONCLUSION**

Road sign detection and vehicle controlling in the semi-automated vehicle when the driver is distracted is the main objective of this project. It enhances safety by informing the drivers about the current traffic signs on the road and giving valuable information about precaution. This project’s implementation focused on real-time video processing, however, for future work, the use of car's dynamics (direction, trajectory, speed changes etc.) should be considered to improve the system’s robustness of the speed sign reading process. A comparison of the performance within an embedded system of this project will provide the baseline of the improvements

#### **7.2 FUTURE SCOPE**

The proposed system detects road signs. As the artificial intelligence be further implemented to and warning signs such crossing etc. along with Raspberry

• This can be implemented

- Color segmentation

- Shape recognition and sign

- Neural network

The proposed system detects road signs. Intelligence is emerging, this project can

to recognize the speed limit signs as pedestrian crossing, rail road Raspberry Pi. implemented by sign description

**8.BIBLOGRAPHY**

#### **8.1 REFERENCES**

1.K. Vinothini and S. Jayanthy, "Road Sign Recognition System for Autonomous

Vehicle using Raspberry Pi," 2019 5th International Conference on Advanced

Computing & Communication Systems (ICACCS), 2019, pp. 78-83, doi:

10.1109/ICACCS.2019.8728463.

2. N Radhakrishnan, and S Maruthi, “Real-time indian traffic sign detection using Raspberry Pi and

3. Open CV”, International Journal of Advance Research in Science and Engineering, Vol.06 Issue no. 11,

2017.

4. D. Priyanka, K. Dharani, C. Anirudh, and K. Akshay, “Traffic Light and Sign Detection for

Autonomous Land Vehicle Using Raspberry Pi,” IEEE International Conference on Intelligent

Computing and Control, pp. 160–164, 2017.

5. Raspberry Pi, https://www.raspberrypi.org/

6. OpenCV, <https://opencv>.

#### **8.2 GITHUB LINK**