# Vision Based Intelligent Vehicle Control

By

**Savera Yousaf**

**Zoha Ali Kayani**

**Muhammad Talha Khalid**

**Hassaan Bin Nadeem**

**Project Supervisor: Assoc. Prof. Dr Ata Ur Rehman**

Submitted to the faculty of Department of Electrical Engineering,

Military College of Signals, National University of Sciences and Technology,

Islamabad,

in partial fulfillment for the requirements of B.E Degree in Electrical (Telecom)

Engineering.

June 2023

In the name of ALLAH, the Most benevolent, the Most Courteous

## CERTIFICATE OF CORRECTNESS AND APPROVAL

*This is to officially state that the thesis work contained in this report*

**"Vision Based Intelligent Vehicle Control"**

*is carried out by*

**Savera Yousaf, Zoha Ali Kayani, M Talha Khalid, Hassaan Bin Nadeem**

*under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.*

**Approved by**

**Supervisor**
**Assoc. Prof. Dr Ata Ur Rehman**
**Department of EE, MCS**

Date:___________

## DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

## ACKNOWLEDGEMENTS

Allah Subhan’Wa’Tala is the sole guidance in all domains.

Our parents, colleagues and most of all supervisor, Assoc. Prof. Dr Ata Ur Rehman without your guidance.

The group members, who through all adversities worked steadfastly.

## Plagiarism Certificate (Turnitin Report)

This thesis has 14% similarity index. Turnitin report endorsed by Supervisor is attached.

Savera Yousaf
00000307200

Zoha Ali Kayani
00000307800

M Talha Khalid
00000283519

Hassaan Bin Nadeem
00000325409

Signature of Supervisor

# ABSTRACT

This project develops prototype of self-driving car. The vehicle is equipped with camera. To easily identify road, we used histogram analysis. This method requires little memory and lower processing power. A raspberry pi is sufficient for it. For pre-processing, we applied binary thresholding, perspective warping and region of interest. The car can drive itself on the road.

Drive optimization is achieved by averaging method for smoothing the road. In this novel method, use of histogram makes the process applicable in small computers. As small computers like Raspberry Pi have very small amount of computing capability, our system is capable to run the full process. The image processing method we proposed here is far better for efficient use of memory and processing power. We used histogram for its superiority in any color intensity variation.

This aim of this project is to develop an autonomous vehicle that can efficiently and safely navigate roads by detecting the curve direction without human intervention. We integrate advanced technologies like computer vision, image processing, and machine learning to enable the vehicle to interpret and perceive its environment to make decisions and control the movements of the car based on its decisions.

# Table of Contents

# List of Figures

# CHAPTER:01 INTRODUCTION

# CHAPTER 01

# INTRODUCTION

## 1.1 Chapter Overview

This chapter sketches a detailed introduction of Vision Based Intelligent Vehicle Control, accentuating the overview, problem statement, proposed solution, objectives and scope of the project are under discussion.

## 1.2 Brief Introduction

A driverless vehicle is also known as self-driving car. It implements a hardware and software integration to enable a vehicle to sense its environment and moving safely without any human input. It uses a variety of sensors to perceive the surroundings. Advanced Control systems interpret sensor values and determines optimum navigation path.

There is a great amount of road accidents occurring in Pakistan due to two-way roads. Most of them are caused by face-to-face collision. As reckless drivers try to overtake, they jump into the wrong lane and this creates catastrophic accidents.

According to SAE International, there is six levels of automation. The levels are:

- Level 0 is warning system but no automated control implemented.

- Level 1 dictates shared control between human and computer system. Such as lane keeping assistance, automated emergency braking, adaptive cruise control, etc.
- Level 2 automated system controls the car but the human needs to be always ready to take over in case of system failure.
- Level 3 relaxes the human driver, as human can move his eyes away.
- Level 4 assures that driver can go even sleep. This can be applied in specified limited locations.
- Level 5 is fully reliable autonomous vehicle. There will be no need for driver.

Level of Automation is shown in following figure 1-1

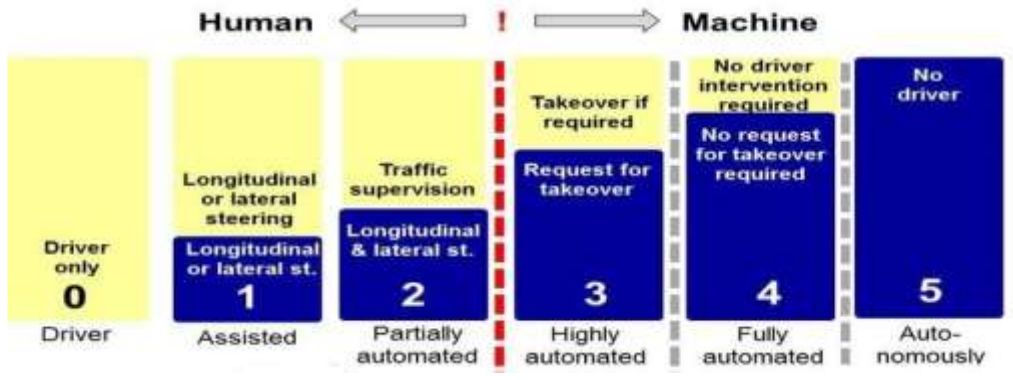


*Figure 1-1 Automation Level*

### 1.3 Project Overview

This project improves road safety by incorporating automation in cars. Images from front camera are processed by on board computer to detect and drive the car on lane. This will increase safety and eliminate wrong lane collisions. Now a days cars are fully manual; drivers need to take full control all the time and every moment. But with the help of this projects system, the car will be able to detect road and even drive through the lane. The computer takes visual snapshots and with processing detects road. It can drive the car along the lane.

As the system uses modular design, future improvements are possible by integrating new sensors and updating software.

## 1.4 Problem Statement

80% of auto accidents are the result of driver error. Accidents frequently involve a person or animal crossing the path accidentally, and a normal person is often unable to act or react right afterwards. The elderly, young children, and individuals who have difficulty driving a car must depend on others to get them where they need to go. The biggest cause of traffic jams, which wastes a lot of time and fuel and can be dangerous at times when there is an emergency, is careless and hasty driving techniques. This project is the first step towards automation of cars which require very less human effort.

## 1.5 Proposed Solution

A proposed solution for a Raspberry Pi self-driving car involves integrating the various components necessary for autonomous driving and developing software to enable the car to operate safely and efficiently. Here are some of the steps involved in the proposed solution:

- **Hardware**

  The first step is to assemble the hardware components required for autonomous driving, such as cameras, actuators, and a Raspberry Pi board. The hardware components are carefully chosen and integrated to ensure reliable performance and efficient operation.

- **Software**

  The next step is to develop the software necessary to enable autonomous driving. This involves creating software modules for perception, planning, and control. The perception module processes the data collected by the sensors and generates a representation of the environment. The planning module determines the car's trajectory based on the environment representation. The control module sends commands to the actuators to execute the planned trajectory.

- **Testing**

  Once the software modules are developed, they must be tested thoroughly to ensure that the car can operate safely and efficiently in various driving scenarios.

- **Integration**

  After the software modules have been tested, they must be integrated into a complete autonomous driving system. This entails attaching the hardware to the Raspberry Pi board and integrating the several software modules to make sure they function together properly.

- **Deployment**

  Once the autonomous driving system has been integrated and tested, it can be deployed on the self-driving car.

In summary, a proposed solution for a Raspberry Pi self-driving car involves integrating hardware and software components to enable autonomous driving and testing the system thoroughly to ensure safe and efficient operation.

## 1.6 Working Principle

The working principle of a Raspberry Pi self-driving car involves multiple components working together to enable autonomous driving capabilities. Here are the key components and how they work together:

- **Camera**

  The self-driving car uses camera, lidars, and ultrasonic sensors, to perceive the environment around it. The camera collects images to feed it to the onboard computer.

- **Onboard Computer**

  The Raspberry Pi is the onboard computer that processes the data collected by the camera. It uses computer vision techniques to detect objects plan the car's trajectory.

- **Actuators**

  The self-driving car uses actuators, such as motors, to control the car's movement. The Raspberry Pi sends commands to the actuators to steer the car, accelerate, brake, or change gears.

- **Software**

  The self-driving car software includes various modules, such as perception, planning, and control. The perception module processes the data from the camera, the planning module determines the car's trajectory and the control module sends commands to the actuators to execute the planned trajectory.

Autonomous driving is a multi-step procedure. The computer takes picture through the camera and process the image frame to find road. After separation of the road from the image background, calculates the curve of the road. If the road goes straight, then command the driver motors to go forward; if the road curves to left, then turns the wheel so the car goes left; if the road curves to right, then the left wheel turns more to make the car turn right. The process then loops again from taking image.

In summary, the working principle of a Raspberry Pi self-driving car involves sensors to perceive the environment, an onboard computer to process data, actuators to control the car's movement and software to enable autonomous driving.

## 1.7 Objectives

### 1.7.1 General Objectives

To develop a fully autonomous automobile that can minimize human labour, lower accident rates, improve fuel efficiency, and improve traffic flow. Self-driving cars were developed to help society by, for example, providing transportation to individuals who are unable to drive due to age or physical incapacity.

### 1.7.2 Academic Objectives

The academic objectives that we want to achieve while developing this project are:

- **Deepening understanding of robotics**

  The project helped us with an opportunity to explore robotics in-depth and gain hands-on experience in applying them to real-world problems.

- **Developing skills in software engineering**

  Developing the software components require skills in software engineering, including design patterns, coding best practices, and software testing. The project can provide an opportunity to develop these skills and gain practical experience in developing large-scale software systems.

- **Applying image processing techniques**

  The car relies on image processing techniques as well as computer vision to perceive and interpret the environment. The project provided an opportunity to apply these techniques to a real-world problem.

- **Enhancing problem-solving skills**

  Developing this project require complex problem-solving skills, including identifying the relevant variables, designing algorithms to process data, and debugging the system. The project provided an opportunity to develop and enhance these skills.

- **Encouraging interdisciplinary collaboration**

  Developing a self-driving car project can require collaboration between different fields such as robotics, computer science, and electrical engineering. The project can provide an opportunity to work with others and learn from their expertise.

- **Contributing to scientific research**

  This project can contribute to the wider scientific research on autonomous driving and robotics. The project can provide an opportunity to make a meaningful contribution to the field.

In a nutshell, developing a Raspberry Pi self-driving car project can provide an opportunity to deepen understanding of robotics, develop skills in software engineering, enhance problem-solving skills, encourage interdisciplinary collaboration, and contribute to scientific research.

## 1.8 Scope

The scope of autonomous vehicle is vast. The modern world is giving importance on road safety and modernization. For this automation is necessary. Automated vehicles are emerging and biggest tech-giants are investing in this field. New experiments are being conducted by GOOGLE, APPLE, TESLA, BMW and other world leading companies. Both technology and automotive companies are incorporating new features like lane assist, parking assist, and collision avoidance system. So this field is emerging quickly. Starting with simple mathematical algorithms, there is a possibility of machine learning too.

## 1.9 Relevant Sustainable Development Goals

The Locally Relevant Socio-Economic Issue that the Project addresses is “Industry ,Innovation and Infrastructure”.

SDG 9 states that we should "build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation."A self-driving car project can contribute to this goal in several ways:

- **Innovation**

  The development of a self-driving car using a low-cost, widely available computer like the Raspberry Pi is an innovative use of technology that has the potential to disrupt traditional approaches to transportation.

- **Sustainable Infrastructure**

  A self-driving car could be part of a sustainable transportation infrastructure, reducing the need for personal vehicles and decreasing traffic congestion, air pollution, and carbon emissions.

- **Inclusive Industrialization**

  By using low-cost, accessible technology like the Raspberry Pi, a self-driving car project can promote inclusive and sustainable industrialization by enabling individuals and communities to participate in the development of autonomous vehicle technology.

- **Resilient Infrastructure**

  A self-driving car project could also contribute to the development of more resilient transportation infrastructure by improving safety, reducing the risk of accidents caused by human error, and potentially reducing the impact of natural disasters on transportation networks.

Overall, a Raspberry Pi self-driving car project can address SDG 9 by promoting innovation, sustainability, inclusivity, and resilience in transportation infrastructure and industrialization.

## 1.10 Structure of Thesis

Chapter 2 contains the literature review and the background and analysis study this thesis is based upon.

Chapter 3 contains the detailed technical specifications of the project.

Chapter 4 contains the proposed solution of the project.

Chapter 5 highlights the conclusion of the project.

Chapter 6 highlights the future work needed to be done for the commercialization of this project.

# CHAPTER 2:

# LITERATURE REVIEW

# CHAPTER 02

# LITERATURE REVIEW

## 2.1 Chapter Overview

This chapter deals with comprehensive details of autonomous vehicles and solutions offered worldwide and their limitations.

## 2.2 Literature Review

### 2.2.1 Retrieve Data from Images

In 1975 Hummel  publishes a paper about how we can retrieve various data from raw grayscale images. He suggested various techniques for analyzing histograms that makes it easy to get features from the gray level of images. A technique that uses the properties of the road's grey level histogram to identify lane markings was proposed by Juan Pablo et al. Following a decision tree analysis of each lane marker, structures defining the lane borders are created by analyzing the relationships between lane markers.

### 2.2.2 Hough Transform for Lane Detection

Patiyuth et al  proposed a system that is mostly accurate in an environment where there are no objects or obstruction, the road is clearly seen and the light is not too bright. They used Hough transformation for lane line detection. Pritha Gupta et al  proposed an autonomous car that also features obstacle avoidance and remote control. They uses Hough line transformation. They also have an extra feature of region of interest. The region of interest is a very good way to reduce calculating area in the view. Only lower third of the view is used for finding lane. Narayan P. P. et al  uses ultrasonic sensor to avoid obstacles. Though they used Hough line transformation for finding lane lines and gave satisfactory results. They also compressed there video before streaming. Network bandwidth made a significant impact on video transmission.

### 2.2.3 Preprocessing of Images

R. Muthalagu et al  proposed a better way to detect lane or road markings. They emphasized on preprocessing of the image. Both Sobel operator and color thresholding is used. Sobel operator is related to canny edge detection, which is further used with Hough transformation. At the end they applied sliding window search technique. The first located sites of maximum are fitted with a third order spline. First lane is designated by the spline.  There is a shortcoming in this method, it heavily depends on preprocessing of image. This preprocessing is computationally expensive. Also, if there is a steep curve then it might be missed from defined ROI.

### 2.2.4 Edge Detection

A tiny autonomous car was created by Rami Watheq Yaseen et al. They employed the deterministic kalman filter. This kalman filter is used to recognise and monitor lanes in this situation. Additionally, they developed a finite state system that coordinated lane following, parking techniques, and path planning. The changes they made for the lane detecting involved taking into account the two areas of interest. As a result, the search space is less. The vertical Sobel operator provides the foundation for edge detection.

Without using a random sample consensus, the vanishing point is found by connecting the two appropriate lane markers.

# CHAPTER 3:

# TECHNICAL SPECIFICATION

# CHAPTER 03

# TECHNICAL SPECIFICATIONS

## 3.1 Overview

This chapter covers the project's technical requirements as well as the specifics of theproject's key steps as it progresses.

## 3.2 System Design

Input, output, and a central processing computer make up the majority of the system. This is shown in following figure 3-1

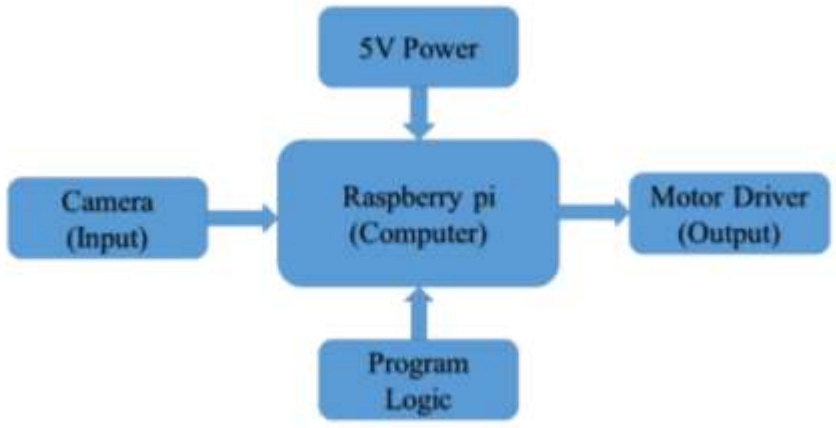


*Figure 3-1 Main components*

## 3.3 Hardware Requirements

### 3.3.1 Chassis:

The load-bearing portion of a car's structure should be referred to as the "chassis" when using the word. The horizontal portion of the vehicle is what connects the other components of the structure. The drive unit can deliver power to the wheels thanks to a collection of mechanical components called the chassis. The vehicle's engine and gearbox loads, as well as other parts and occupants, are supported by the chassis. Maintaining the car's shape helps keep it stiff and prevents deformation when it is in operation.

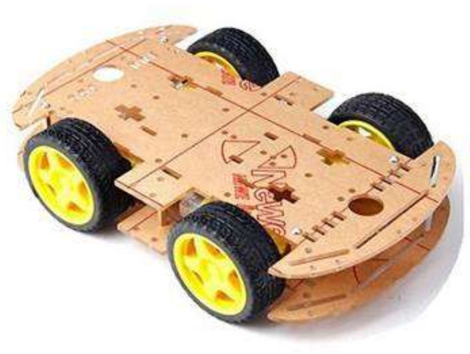


*Figure 3-2 Robot Car Chassis*

### 3.3.2 Raspberry pi:

The Raspberry Pi is a relatively affordable Linux-powered computer that also features a set of GPIO (general purpose input/output) pins that let you explore the Internet of Things (IoT) and control electrical components for physical computing.

The Raspberry Pi is ideal for adaptive technology since it can play films or display graphics at high quality to construct systems like embedded system prototyping. This product enables the construction of complicated and efficient structures at a lower cost. Raspberry Pi boards are the most habituated Single Board Computer (SBC) nowadays because of their amazing features that include an ARM Cortex Processor, Bluetooth and WiFi support, multiple USB ports, a good Broadcom Video Core VI GPU, Linux based operating system, HDMI, and others.

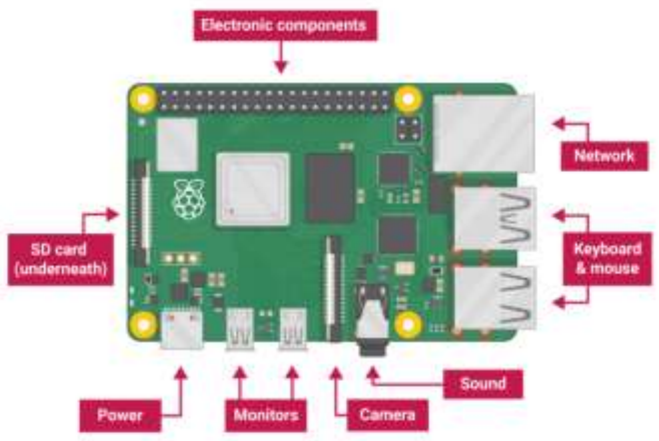


*Figure 3-3 Raspberry Pi*

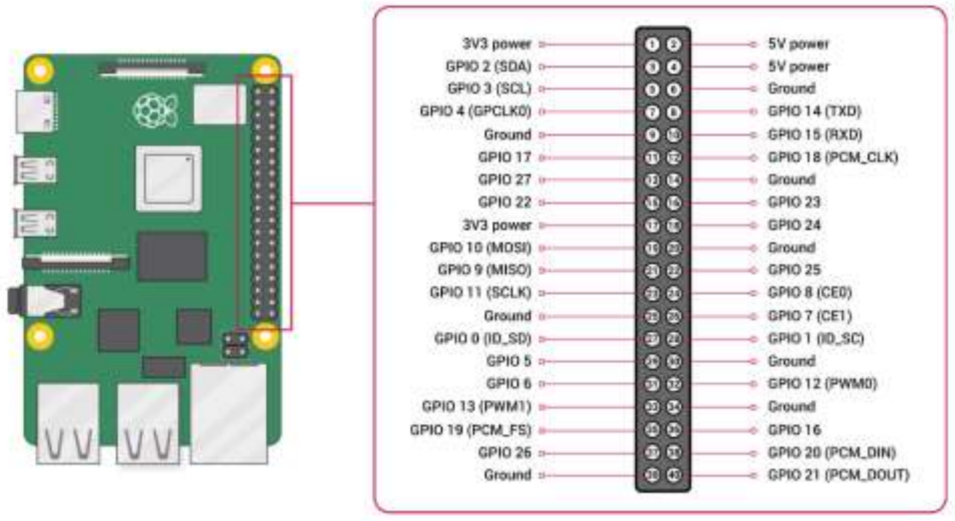
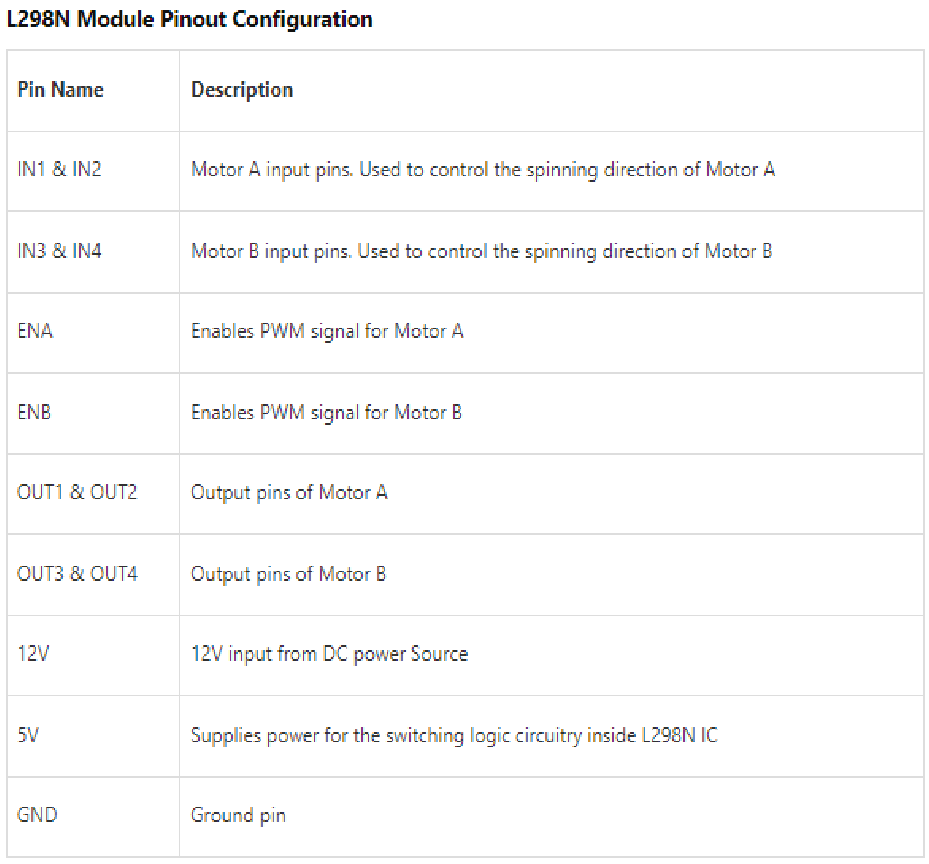


*Figure 3-4 Pin Configuration of Raspberry pi*

### 3.3.3 Motor Driver:

Using a tiny voltage signal from a microcontroller or control system, we employ motor drivers to supply the motor with high power. If the CPU sends the motor driver a HIGH input, the motor will revolve in that direction as long as one pin is HIGH and the other is LOW. The twin H-Bridge motor driver L298N enables simultaneous speed and direction control of two DC motors. The module can run DC motors with peak currents up to 2A and voltages between 5 and 35V.



**L298N Module Pinout Configuration**

| Pin Name | Description |
|---|---|
| IN1 & IN2 | Motor A input pins. Used to control the spinning direction of Motor A |
| IN3 & IN4 | Motor B input pins. Used to control the spinning direction of Motor B |
| ENA | Enables PWM signal for Motor A |
| ENB | Enables PWM signal for Motor B |
| OUT1 & OUT2 | Output pins of Motor A |
| OUT3 & OUT4 | Output pins of Motor B |
| 12V | 12V input from DC power Source |
| 5V | Supplies power for the switching logic circuitry inside L298N IC |
| GND | Ground pin |

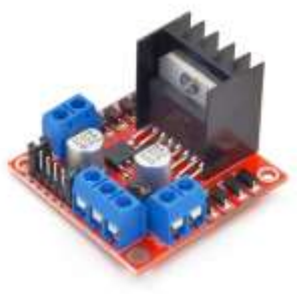


*Figure 3-5 Motor Driver L298N*

### 3.3.4 Power Bank:

When you're on the go and need to recharge your electronic gadgets, a power bank is a portable charger that you may use. Power banks work by storing energy in their batteries, which charge electronic devices. When the power bank connects to a device, it will begin transferring its stored energy to the device. Power banks typically use USB ports to connect to devices, though some may also have wireless charging capabilities.

*Figure 3-6 Power Bank*

### 3.3.5 Camera:

A webcam is a computerized device commonly built into a computer. Its main function is to transmit pictures over the Internet.

*Figure 3-7 Webcam*

### 3.3.6 Ultrasonic Sensor:

The time elapsed between the emission and reception is measured by ultrasonic sensors to determine the target's distance. While an ultrasonic sensor employs a single ultrasonic element for both emission and reception, an optical sensor uses a transmitter and a receiver. Ultrasonic sensors are predominantly utilized as proximity sensors to detect the presence of objects within a specific range. They are commonly employed in automobile self-parking technology and anti-collision safety systems, as well as in robotic obstacle detection systems and manufacturing technology. These sensors operate by emitting the time it takes for the waves to return. The information gathered is then used to calculate the distance between the sensor and the object. The configuration pin of HC-SR04 is VCC (1), TRIG (2), ECHO (3), and GND (4). The supply voltage of VCC is 5V and you attach TRIG and ECHO pin to any Digital I/O in your Microprocessor Board to power it.

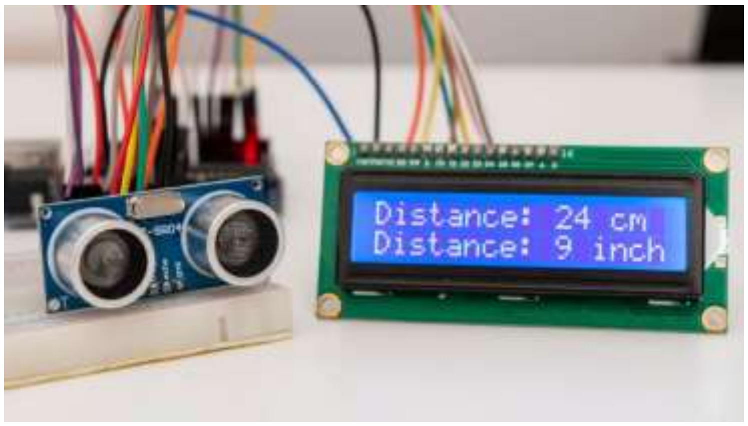


*Figure 3-8 Ultrasonic Sensor*

### 3.3.7 Jumping Wires:

A jumper wire is an electric cable used to link distant printed circuit board electric circuits. It is possible to short-circuit and jump to the electrical circuit by connecting a jumper wire to the circuit. Simply said, jumper wires are cables having connector pins at each end that may be used to connect two places without soldering.

*Figure 3-9 Jumping Wires*

## 3.4 Software Requirements

We need both hardware and software tools to build the system. The software takes the lead in driving the hardware. In our work, we employ the following software, programming language, library, package, etc.

### 3.4.1 Operating System:

Raspberry Pi OS is a free, open-source Debian Linux-based operating system engineered for use on Pi boards. The latest version of raspberry pi OS was released on February 21st 2023 which uses kernel version 5.15 and Debian version 11 which we used for our project.

### 3.4.2 OpenCV:

OpenCV is a large open-source library that has many applications in fields such as computer vision, machine learning, and image processing. It is essential for real-time operations, which are crucial in modern systems. OpenCV is used for detecting objects, faces, diseases, number plates, and even handwriting in images and videos. Additionally, OpenCV is compatible with several programming languages, including Python, and can be easily imported with a single line of code.

```
pip install opencv-python
```

### 3.4.3 Python:

Python is the programming language commonly used for developing software for the Raspberry Pi. It is pre-installed in Raspberry Pi OS. However, we installed additional libraries and modules as per our project requirements as follows

- **Numpy**

  NumPy is a widely-used Python library for scientific computing and data analysis. It provides an efficient multi-dimensional array object, as well as a large collection of mathematical functions to operate on these arrays. NumPy is the fundamental package for scientific computing in Python.

- **Tensor Flow**

  An open-source software package called TensorFlow is used for dataflow and differentiable programming on a variety of applications. Machine learning models may be quickly created, trained, and deployed across a number of platforms thanks to the library's tools and APIs. In our project, object detection is carried out using a TensorFlow classifier. On a Webcam stream, we load the classifier and utilize it to recognize objects.

- **Rpi. Gpio**

RPi. GPIO is a Python module that provides a simple interface for programming the GPIO pins on a Raspberry Pi. It allows developers to control the input and output pins of the Raspberry Pi's General Purpose Input/Output (GPIO) interface using Python code. The RPi. GPIO module is specifically designed for use with the Raspberry Pi and provides a consistent interface across all models and versions of the board.

### 3.4.4 IDE:

An Integrated Development Environment (IDE) can help you write, debug, and test your code. We used Visual Studio Code for our project.

### 3.4.5 VNC Viewer:

VNC Viewer turns a smartphone into a virtual desktop, providing instant access to one's Mac, Windows, and Linux PCs from anywhere in the globe.

# CHAPTER 4:

# PROPOSED SOLUTION

# CHAPTER 04

# PROPOSED SOLUTION

## 4.1 Chapter Overview

The following chapter explicates the entire technical working and methodology of the project. The implementation of hardware components and different software modules is under discussion.

## 4.2 Methodology

Our solution can be broken down into three stages. In the first stage, we set up hardware components and make all the required connections. Stage two consists of five software modules for controlling the movements of the car which are integrated together as well as the hardware. In the third stage, the protype is tested to find errors and finding the optimal values for best results.

## 4.3 Stage 1: Hardware Implementation

Raspberry Pi is the main processing unit of the hardware section. Motors drive car with the help of motor driver. Camera and Ultrasonic sensor are used for detection. Raspberry Pi is powered by a power bank.

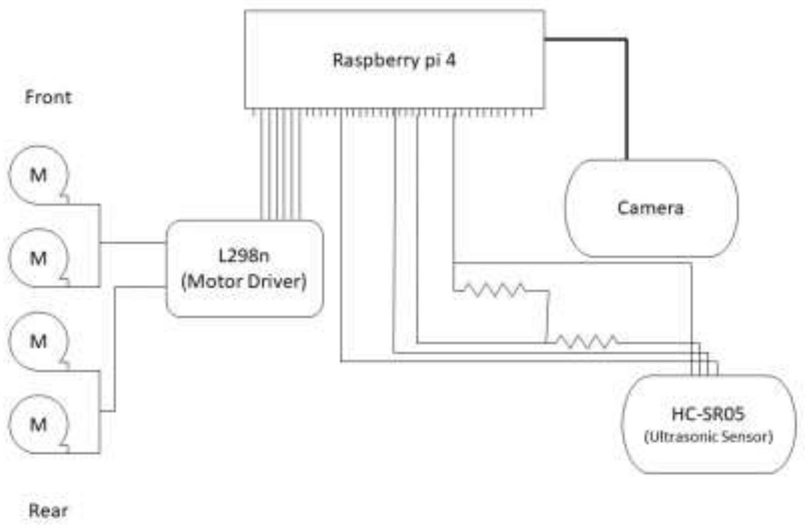


*Figure 4-1 Schematic of Autonomous car*

### 4.3.1 L298n (Motor Driver):

DC motors are connected with the wheels to move the car. The motor are further connected to L298n motor driver in the form of pair as shown in figure. The motor driver takes input from Raspberry Pi through In1, In2, In3 and In4 pins. Each pair of motors is enabled by En1 and En2 pins. These pins are connected to Raspberry Pi through GPIO pins. The motor module gets power through $V_{CC}$ pin that is connected to 5V pin of Raspberry Pi. Ground is connected to ground pin of Raspberry Pi. This motor driver module gives output to the motor through OUT1, OUT2, OUT3 and OUT4.

### 4.3.2 HC-SR05 (Ultrasonic Sensor):

Ultrasonic Sensor is used for object detection. Vcc is connected to 5V pin of Raspberry Pi. Ground is connected to ground pin of Raspberry Pi. Echo pin is connected to GPIO pin with a 1kΩ resistor. Trig pin is also connected to a GPIO pin of Raspberry Pi in parallel with the Echo pin, with a 2kΩ resistor.

### 4.3.3 Webcam:

Webcam is used for lane detection. It is connected with Raspberry Pi by USB port.

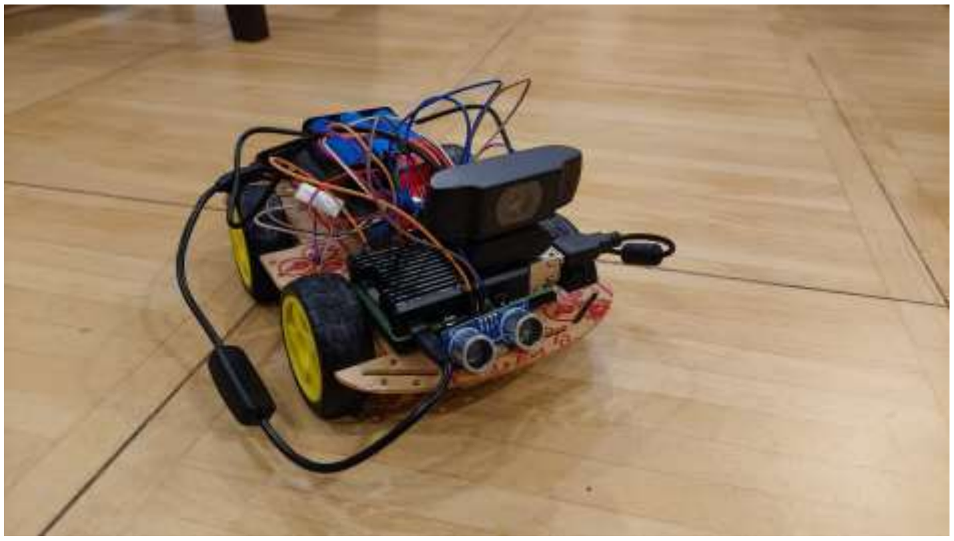


*Figure 4-2 Hardware Implementation*

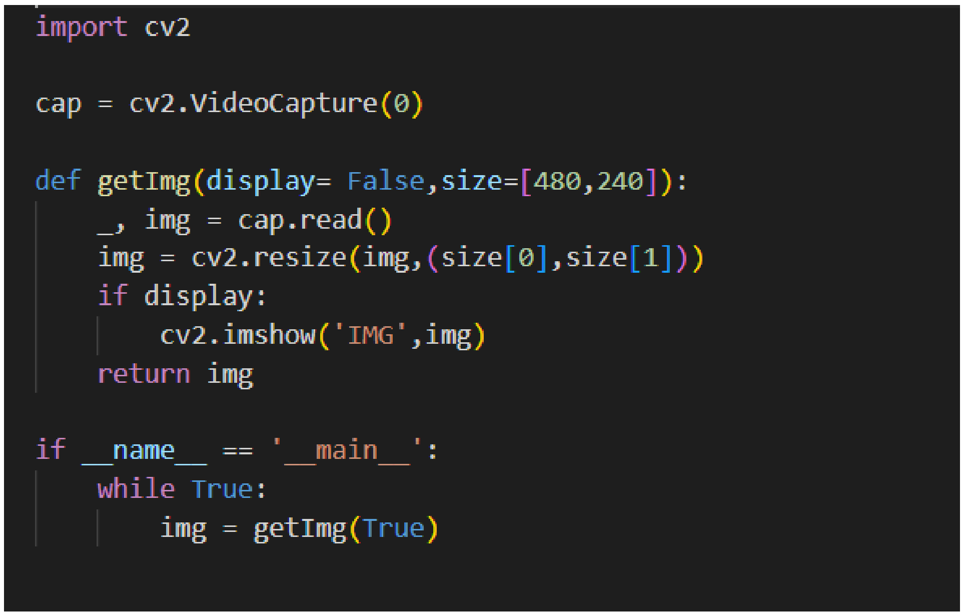
## 4.4 Stage 2: Software Modules

There are five modules that are integrated together that get input from the environment, process the input and then control the hardware to move the car and detect objects in the path. The purpose of using modules for separate tasks is to make the project more flexible. New modules can be easily added or removed as per requirement. These modules are:

- Webcam Module
- Lane Detection Module
- Object Detection Module
- Robot Module
- Main Module

### 4.4.1 Webcam Module:

Webcam Module captures image from the camera and sends it to others modules. There is a separate module for capturing images because more than one module like Lane Detection and Object Detection requires images from camera hence it is not practical to capture images separately in different modules. Moreover, in future when more modules are added such as traffic sign recognition etc. then same Webcam Module will be used.



```python
import cv2

cap = cv2.VideoCapture(0)

def getImg(display= False,size=[480,240]):
    _, img = cap.read()
    img = cv2.resize(img,(size[0],size[1]))
    if display:
        cv2.imshow('IMG',img)
    return img

if __name__ == '__main__':
    while True:
        img = getImg(True)
```

*Figure 4-3 Retrieving Image from Webcam*

### 4.4.2 Lane Detection Module:

The lane detection module takes the image from Webcam module and calculates the curve in it. This module is broken down into four steps:

- Thresholding
- Warping
- Summation of Pixels
- Calculation of curve value

The image processing part is crucial for this project. In this project the road is differently colored from the environment. The system detects the road by differentiating the color. The color is specified beforehand. The road can be any color but it should be distinctive.

The image processing pipeline is summarized in following figure 4-4

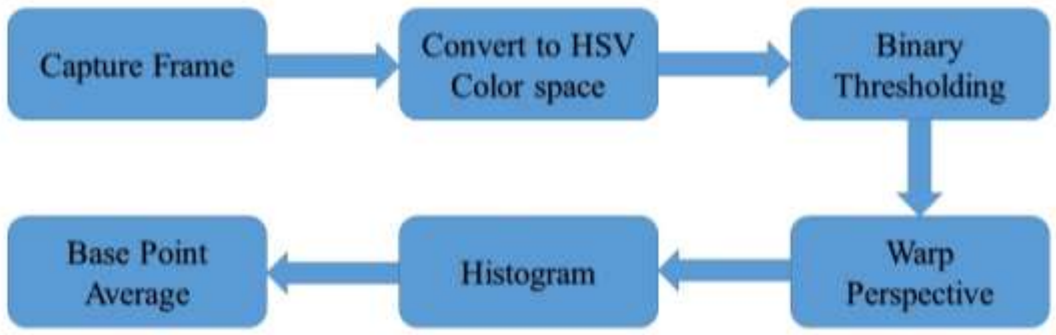


*Figure 4-4 Image Processing Pipeline*

## Step 1: Thresholding

The frame we capture is in RGB format. To easily separate the desired color, we convert it to HSV color space. HSV space is showed in following figure 4-5

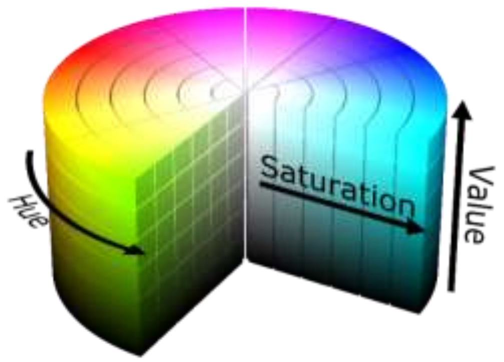


*Figure 4-5 HSV Color Space*

Binary thresholding is done by masking with binary value. The pixels matched with our value is retained and any other pixels are zeroed. Threshold operation effect is shown in following figure 4-6

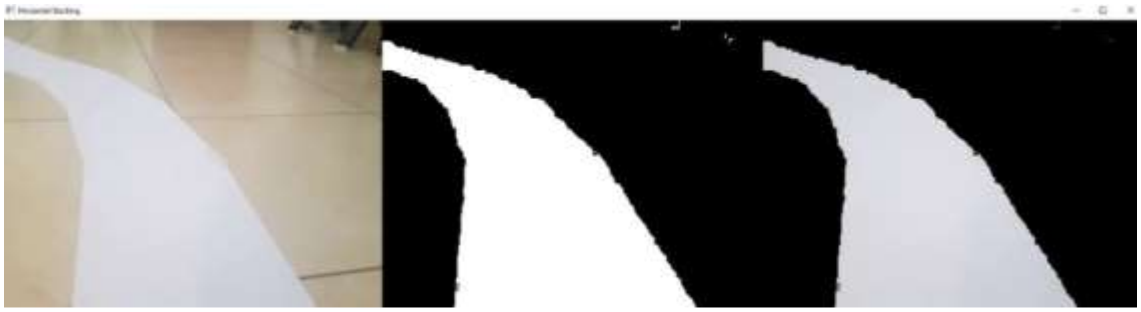


*Figure 4-6  Binary Thresholding*

### Step 2: Warping

For better detection of road curvature angle and correct our camera perspective, we warp detected road. It is also called birds eye view. For example, is shown this figure 4-7

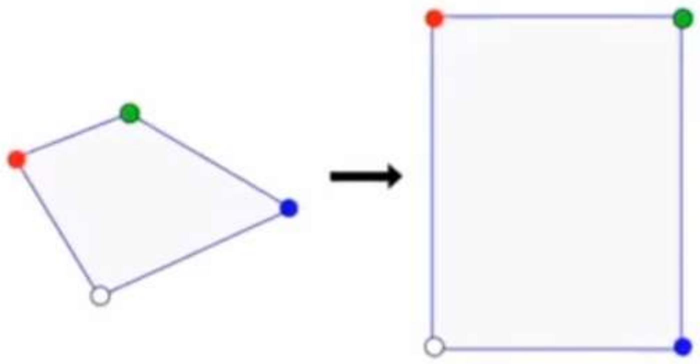


*Figure 4-7 Warp Function*

After warping the image looks like following figure 4-8

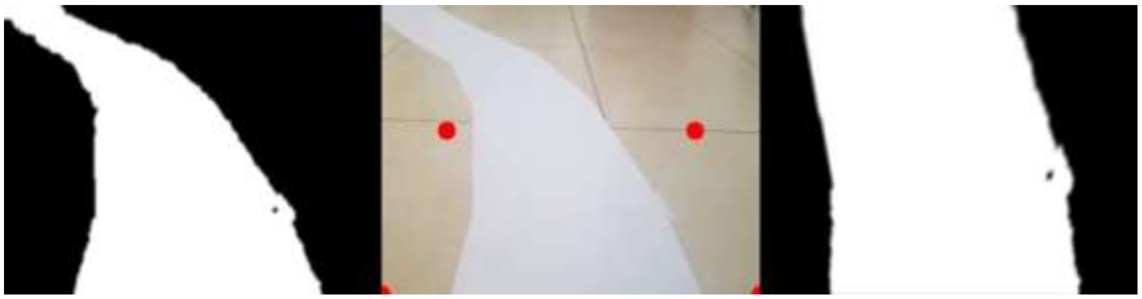


*Figure 4-8 Right most is after warping*

## Step 3: Summation of Pixels

Then histogram method gives us values depending on the weight of the pixels of the road. If the road is curved right then the sum of the pixel values is high on the right side. Movement decision criteria are shown in following figure 4-9

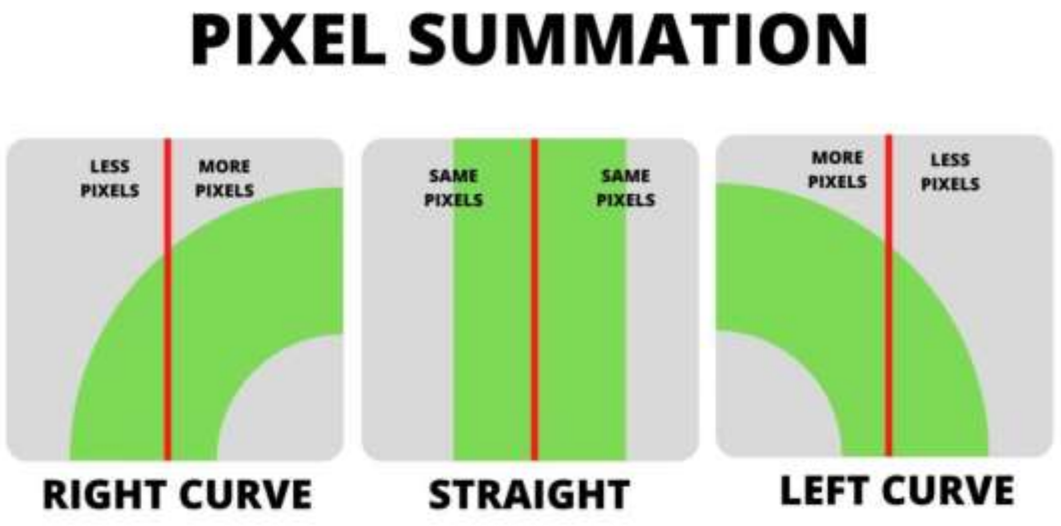


*Figure 4-9 Road Curvature Decision*

On the histogram it looks like the following figure 4-10

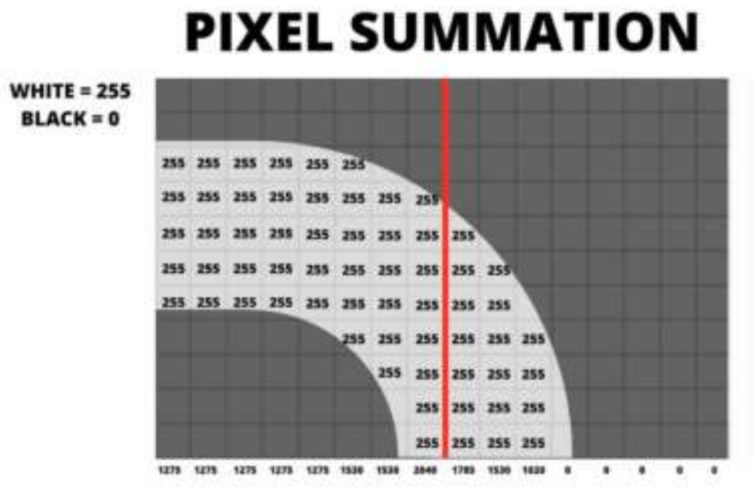


*Figure 4-10 Pixel summations in Histogram for left curve*

Actual results are shown in following figure 4-11

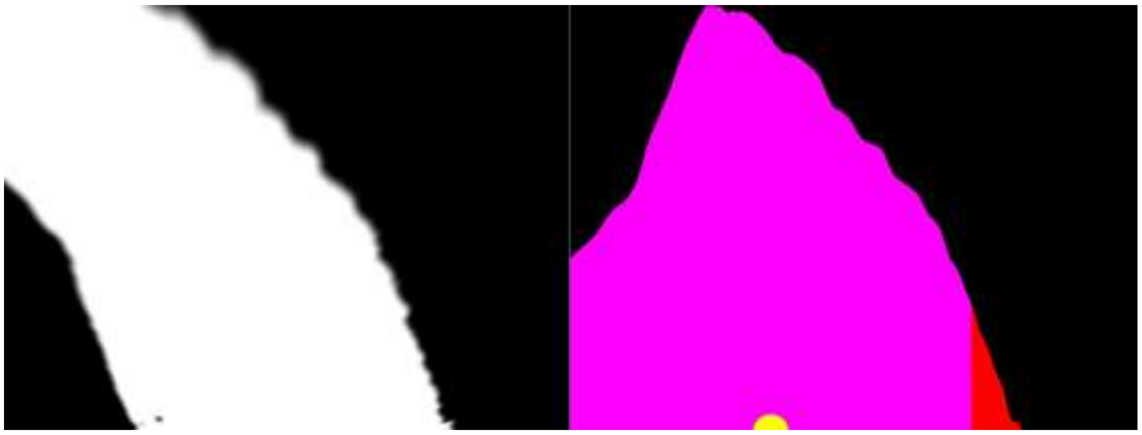


*Figure 4-11 Left side is road image. Right side is histogram and average base point.*

### Step 4: Curve Value

In this step, the curve value is calculated which is then sent to the robot module for decision making.

### 4.4.3 Object Detection Module

Object detection is a critical component of self-driving cars, as it enables the car to detect and identify objects in its environment and take appropriate actions based on that information. TensorFlow provides a powerful and flexible framework for developing object detection models that can be used in self-driving cars.

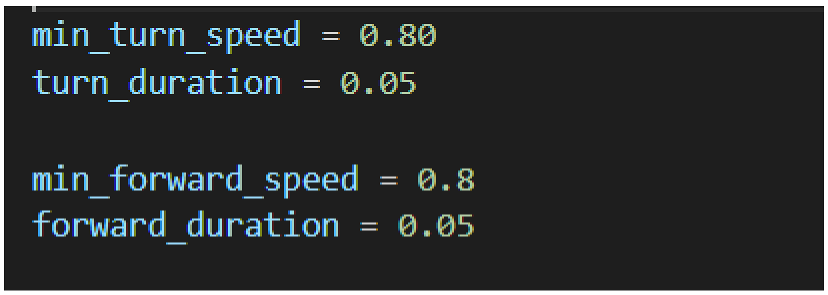
To develop an object detection module using TensorFlow for a self-driving car, the first step is to collect and label a large dataset of images that represent the objects the car needs to detect. Once the dataset is prepared, the next step is to train a CNN using

TensorFlow's high-level API, such as the Keras API, to detect objects in the images. During training, the CNN learns to recognize patterns in the images that correspond to the objects it needs to detect. The model is then evaluated on a separate test dataset to measure its performance and fine-tune the model if necessary.

Once the object detection model is trained, it can be integrated into the self-driving car's software stack and used in real-time to detect objects in the car's environment. The detected objects can then be used to inform the car's decision-making process and enable it to take appropriate actions, such as braking or changing lanes. Overall, TensorFlow provides a powerful and flexible platform for developing object detection modules that can be used in self-driving cars and other applications where accurate and reliable object detection is critical.

### 4.4.4 Robot Module

This module is responsible for moving the motors in the specified direction. The robot module includes commanding the motor drivers, which are electronic circuits that control the speed and direction of the motors. The motor drivers receive commands from the Raspberry Pi and use that information to adjust the motors' speed and direction. The minimum speed and turn speed values are also defined in this module. The optimal working values for our project are:



```
min_turn_speed = 0.80
turn_duration = 0.05

min_forward_speed = 0.8
forward_duration = 0.05
```

*Figure 4-12 Turn Speed and duration values used*

Also, based on the curved value, the motor is moved in that direction. Once we have the curve value, the decision-making process depends on the curve value. It is showed in following figure 4-13.

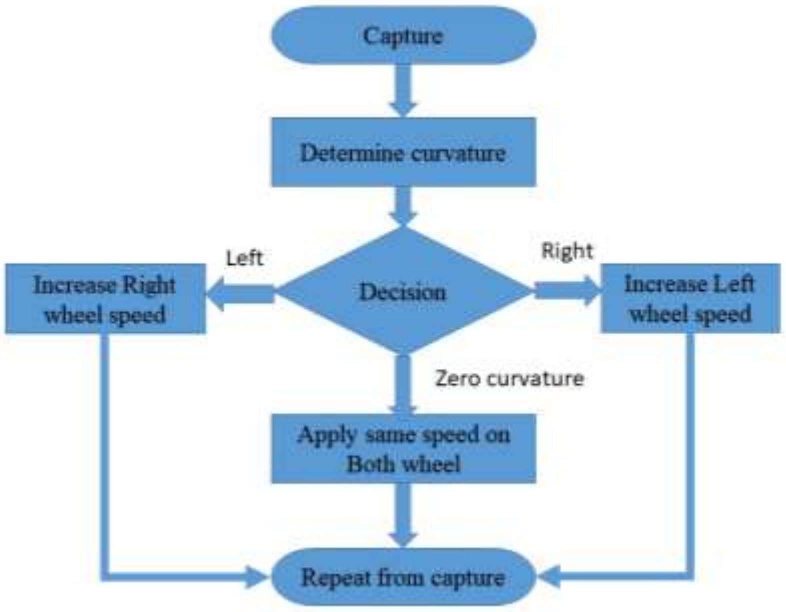
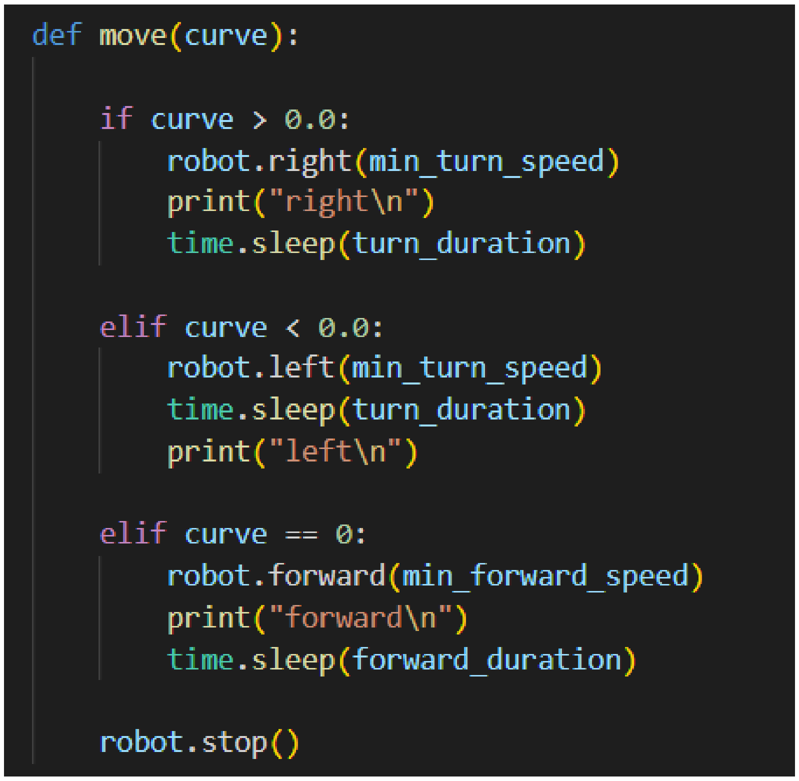

Capture
Determine curvature
Left
Right
Increase Right wheel speed
Decision
Increase Left wheel speed
Zero curvature
Apply same speed on Both wheel
Repeat from capture

*Figure 4-13 Decision Making Process Flowchart*



```
def move(curve):

    if curve > 0.0:
        robot.right(min_turn_speed)
        print("right\n")
        time.sleep(turn_duration)

    elif curve < 0.0:
        robot.left(min_turn_speed)
        time.sleep(turn_duration)
        print("left\n")

    elif curve == 0:
        robot.forward(min_forward_speed)
        print("forward\n")
        time.sleep(forward_duration)

    robot.stop()
```

*Figure 4-14 Decision making on direction of car based on Curve Value*

### 4.4.5 Main Module

The main module is the central control unit. All the separate modules need to combined in one file which has the main function and integrates the code to simultaneously run all files. All the above-mentioned modules are connected to the main module and the results are obtained by running this module.

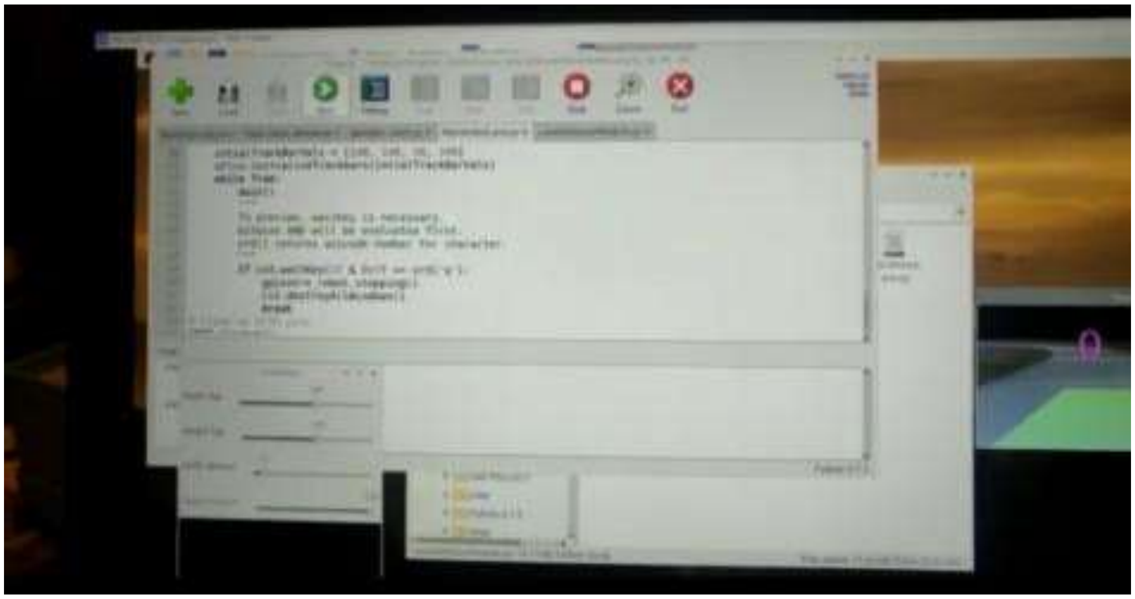
# CHAPTER 5: RESULTS

## 5.1 Results

The car is working properly on the custom-made path in controlled environment. The road is differently colored than the environment.

The car follows the road. If the road take a curve it also follows the curve. We can achieve a high-speed run when the road is mostly straight. White color is tested correctly. There are some limitations in this setup. The car’s speed is reduced voluntarily to synchronize image processing time to motor actuation time.

The code simulation is as follows ;



```
    initialTrackBarVals = [illegible]
    utlis.initializeTrackbars(initialTrackBarVals)
    while True:
        main()
        '''
        To prevent, waitKey is necessary.
        bitwise AND will be evaluated first.
        ord() returns unicode number for character.
        '''
        if cv2.waitKey(1) & 0xFF == ord('q'):
            [illegible]
            cv2.destroyAllWindows()
            break
```

*Figure 5-1 Car moving in forward direction*

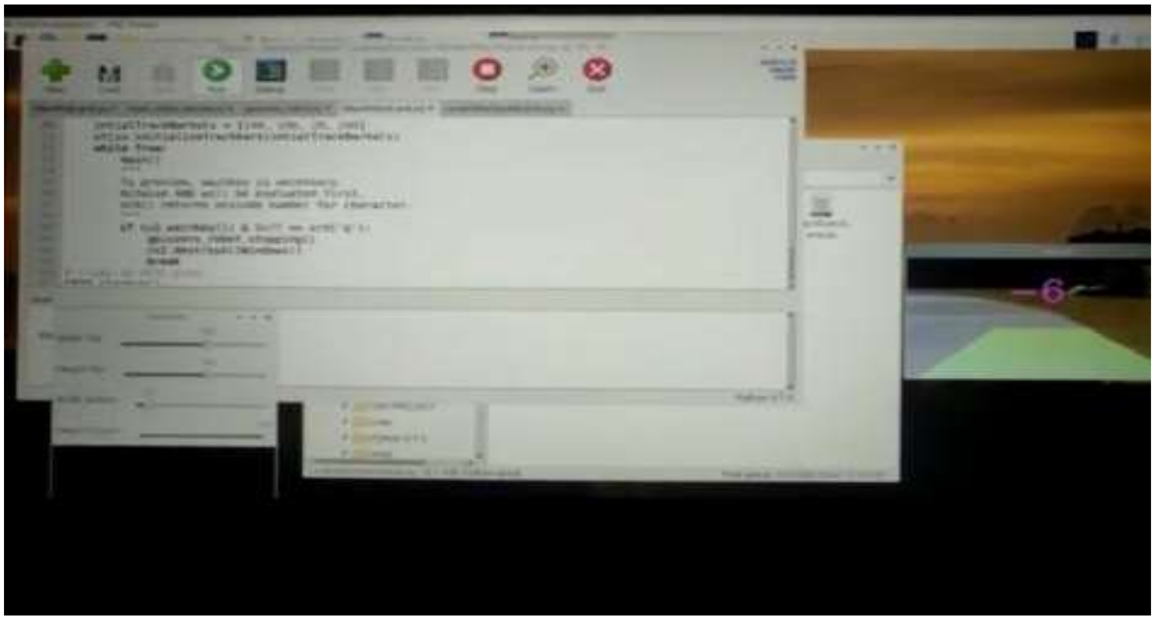


*Figure 5-2 Car moving in left direction*

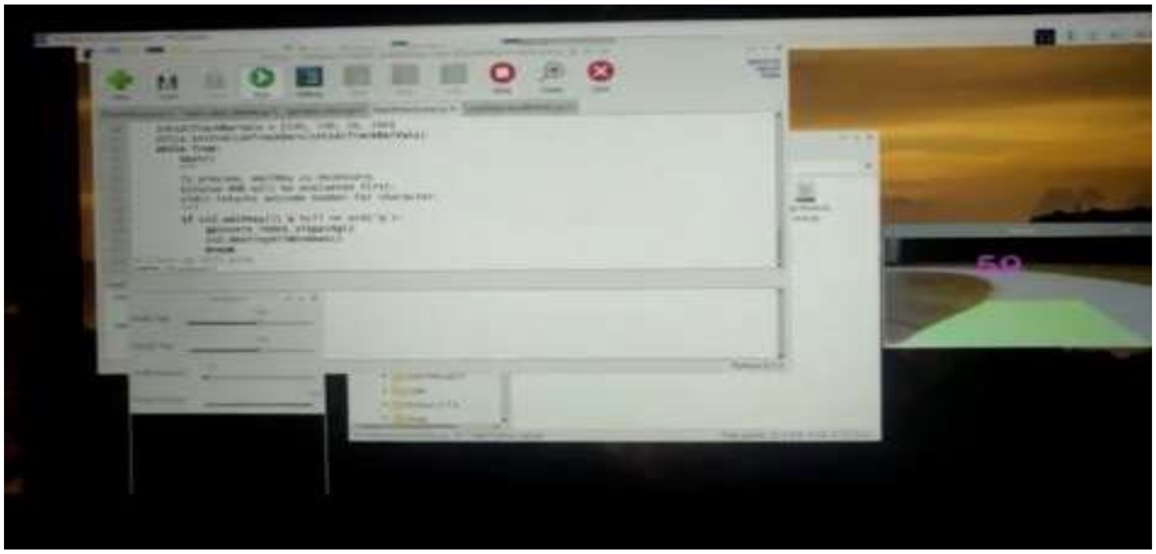


*Figure 5-3 Car moving in right direction*

Actual car functional figures are attached ;

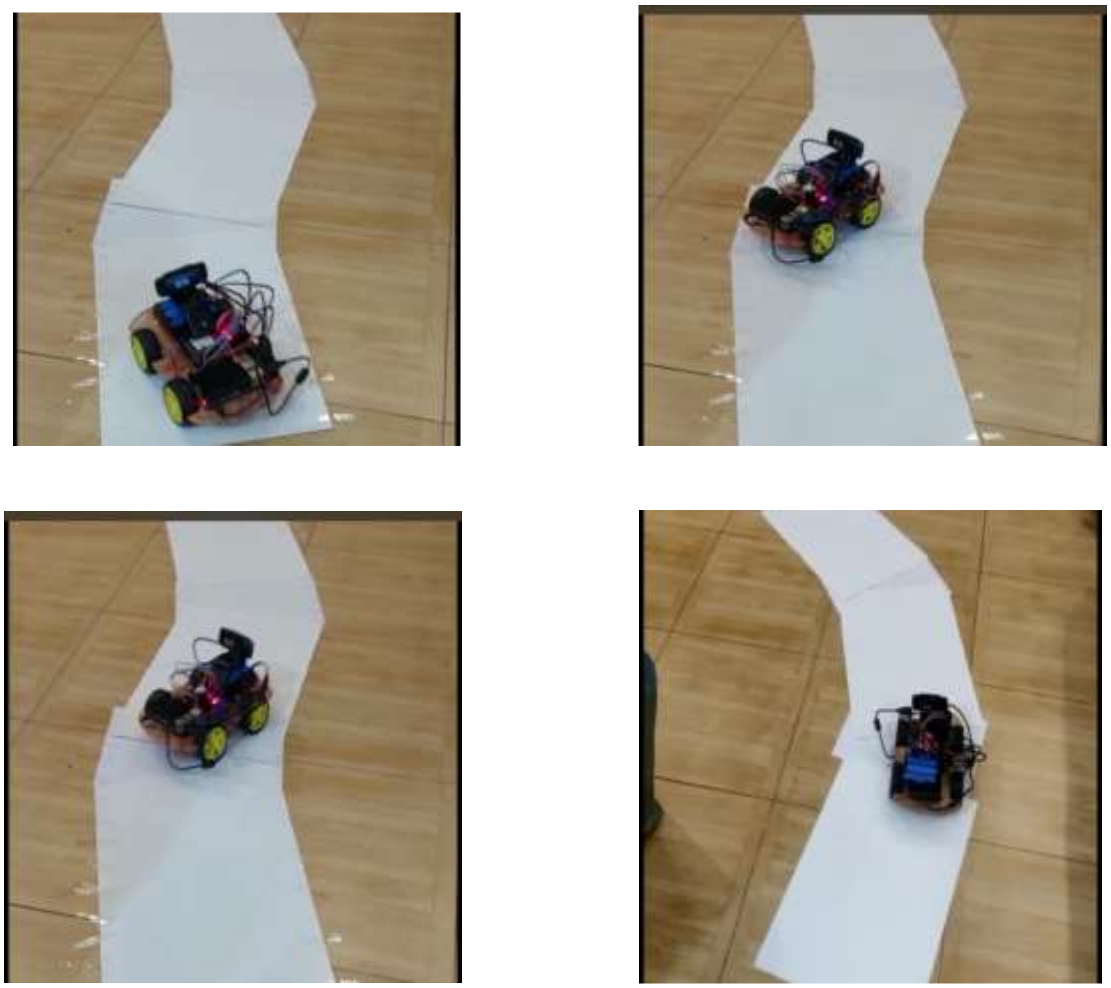


*Figure 5-4 Actual Car Functionality*

## 5.2 Real Time Simulation

A more practical approach is to implement the proposed solution on real-time vehicle. We have also implemented real time simulation of lane detection algorithm which displays the curve of the actual road and gives the curve angle, guiding the car in that direction. However, the steering control of the vehicle is the future scope of this project. The results are shown below:

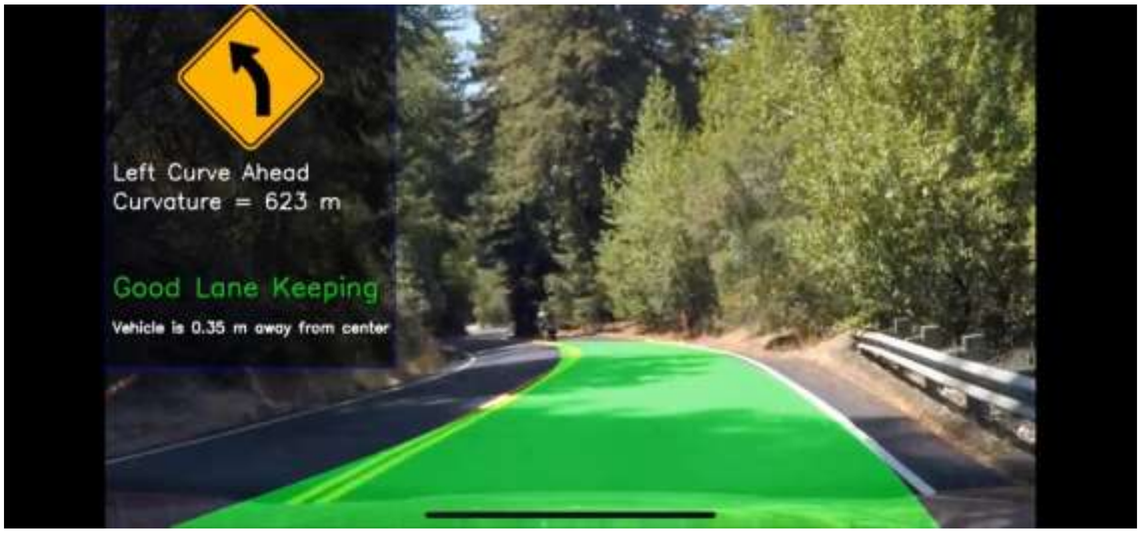


*Figure 5.5 Navigation of left curve on the road*

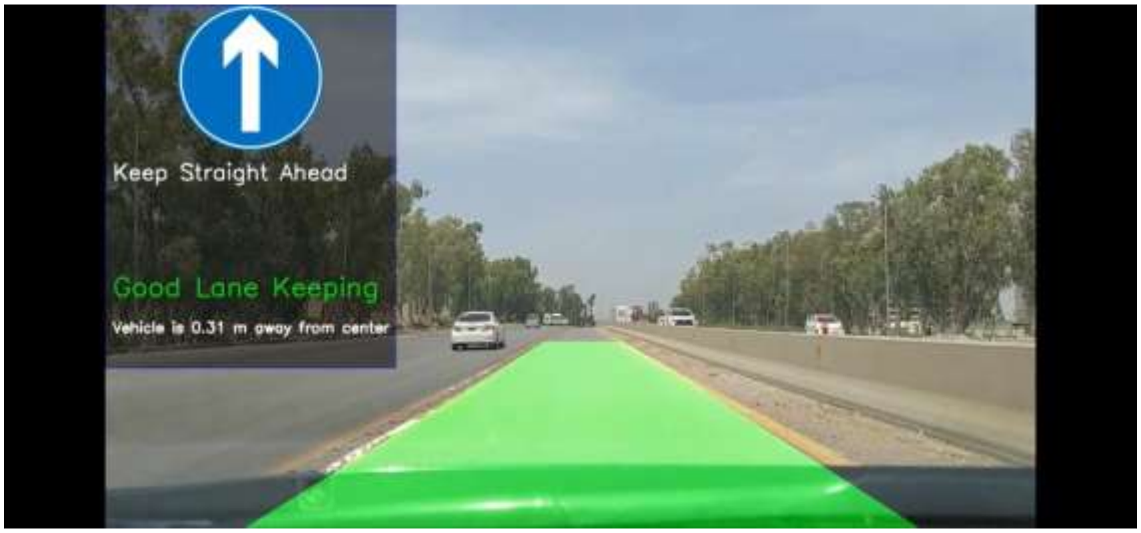


*Figure 5.6 Navigation of straight line on the road*

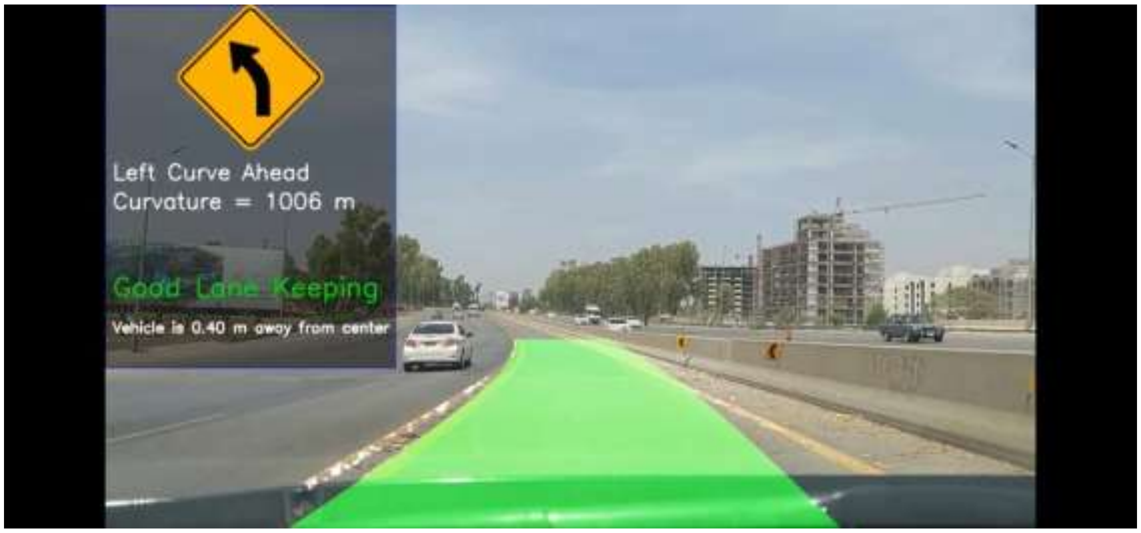


*Figure 5.7 Navigation of left curve on the road*

# CHAPTER 6:

# CONCLUSION AND FUTURE WORK

# CHAPTER 06

# CONCLUSION

## 6.1 Conclusion

In this thesis, a prototype of an autonomous vehicle using raspberry pi and lane detection has been discussed. It provides the detailed description of the hardware and software requirements of the project. It includes the necessary components such as chassis, Raspberry Pi, motor driver, power bank, webcam, ultrasonic sensor, and jumping wires, and explains their function also outlining the software requirements, including the operating system Raspberry Pi OS, OpenCV, and Python. It is important to have a clear understanding of these technical requirements to successfully develop the project.

We developed a prototype of a Raspberry Pi self-driving car that can detect road and drive along the lane using computer vision techniques. The system is designed to be modular, allowing for the integration of new sensors and the updating of software. The project does not include the development of a fully autonomous vehicle capable of navigating in all driving conditions, such as heavy rain, snow, or off-road environments. The project also does not include the legal and regulatory aspects of autonomous driving. The project is limited to a prototype that can operate on a closed track and is not intended for use on public roads.

In conclusion, a Raspberry Pi self-driving vehicle using lane detection is a promising application of computer vision and machine learning technology. By using a camera and image processing techniques, the system is capable of identifying and tracking lanes on the road, allowing the vehicle to stay in its designated lane and avoid collisions.

The use of a Raspberry Pi as the processing unit for this self-driving vehicle is a cost-effective solution, which makes it accessible to hobbyists and small-scale projects. Furthermore, the open-source nature of the Raspberry Pi platform allows for customization and flexibility in the development process.

Although the system may still have limitations in complex driving scenarios, the potential of this technology is significant, and it could lead to more advanced and sophisticated autonomous driving systems in the future. With continued research and development, it may become possible to achieve full autonomy with minimal cost and hardware requirements, making self-driving vehicles accessible to a wider range of people and industries.

## 6.2 FUTURE WORK

Future milestones that need to be achieved to commercialize this project are the following.

### 6.2.1 Night Vision:

Night vision is an important feature in autonomous vehicles, as it can help improve the safety and reliability of these vehicles when driving in low-light conditions. The project can be further enhanced to get a view in the dark, so that driving in the night could be safer. Technologies like infrared sensors, LIDAR and thermal ranging sensors can be used for night vision.

### 6.2.2 Complete Autonomous:

Assisted driving vehicles can be enhanced to become fully autonomous by incorporating additional sensors, algorithms, and computing power. Adding more sensors like cameras, LIDAR, radar, and ultrasonic sensors or improving existing sensors, increasing computing power to improve autonomous vehicle decision-making accuracy can improve autonomous vehicle capabilities. By incorporating these enhancements, assisted driving vehicles can gradually become more capable of autonomous driving, with the goal of eventually achieving full autonomy. However, it is important to note that achieving full autonomy is a complex and ongoing process, and it may take years of research, development, and testing to achieve this goal. Autonomous driving technology has the potential to revolutionize transportation in many different industries beyond just cars..

### 6.2.3 Diverse Automobiles:

Autonomous driving technology has the potential to revolutionize transportation in many different industries beyond just cars. has the potential to transform many different industries by improving safety, efficiency, and productivity. As the technology continues to develop, we can expect to see more and more applications of autonomous driving in various industries beyond just cars. Autonomous driving can be applied to trucks, delivery vehicles, public transportation, agriculture, mining and construction to improve safety and efficiency, reduce labor costs, and increase productivity. It can also help reduce accidents caused by human error.

## References and Work Cited

1. Chaubey, A. (2022, January 4). Future of Self-Driving Cars ; Ft. Machine Learning - students x students. *Medium*. https://studentsxstudents.com/how-machine-learning-can-be-used-in-autonomous-vehicles-4f2b8a64d9c4
2. *Keyboard Controlled Robot with Raspberry Pi 4*. (n.d.). Little Bird Electronics Guide. https://littlebirdelectronics.com.au/guides/150/keyboard-controlled-robot-with-raspberry-pi-4
3. Instructables. (2021). Object Detection on Raspberry Pi. *Instructables*. https://www.instructables.com/Object-Detection-on-Raspberry-Pi/
4. *Basic Recipes — GPIO Zero 1.6.2 Documentation*. (n.d.). https://gpiozero.readthedocs.io/en/stable/recipes.html
5. Macharla, M. (2020). Self Driving Car based on Raspberry Pi and OpenCV. *IoTEDU*. https://iot4beginners.com/self-driving-car-based-on-raspberry-pi-and-opencv/
6. Ijraset. (n.d.). *Research Paper on AI in Driving*. IJRASET. https://www.ijraset.com/research-paper/research-paper-on-ai-in-driving
7. Fathy, M., Ashraf, N., Ismail, O., Fouad, S., Shaheen, L., & Hamdy, A. (2020). Design and implementation of self-driving car. *Procedia Computer Science*, *175*, 165–172. https://doi.org/10.1016/j.procs.2020.07.026
8. Martínez-Díaz, M., & Soriguera, F. (2018). Autonomous vehicles: theoretical and practical challenges. *Transportation Research Procedia*, *33*, 275–282. https://doi.org/10.1016/j.trpro.2018.10.103