Autonomous Self-Driving Car using Raspberry Pi

A Thesis/Project Submitted in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

By

MD. Al - Amin

CSE 055 06842

Supervised by: **Adnan Ferdous Ashrafi**

Senior Lecturer

Department of Computer Science and Engineering

STAMFORD UNIVERSITY BANGLADESH

September 2021

Abstract

Self-Driving car is the most trending technology which already implemented in Tesla cars, as initially, we can learn about the technology by using this system. For this, we are using OpenCV, Machine learning technology. This system contains Raspberry Pi as the core system, which having functionalities like, Traffic light detection, Vehicle detection, pedestrian detection, Road sign detection to make the car as autonomous. Every process is done using the Raspberry Pi with Python programming. The Raspberry Pi which is interfaced with the Car with 4 motors and one servo motors. Through the Driver IC (L298N). Based on the driving parameter, the signal is sent to raspberry to operate the Car. Currently, self-driving cars are already being implemented in foreign countries however these cannot be implemented in Bangladesh. Reason being these existing approaches uses GPS, Sensors. The problem with GPS is that these display roads on the map that might or might not exist and also these roads in Bangladesh might not be a concrete road. In our project, we planned of using a special pattern which will be deployed on the road. These patterns are a special pattern that is used for detection of the pathway and it detects the type of road. Hence using this technology, we can implement a self-driving car in Bangladesh. Our prototype would use a modelled car which has a Raspberry pi to process the captured images from the camera and send it remotely on remote computer process it and send back. Similarly, we have various sensors around the car to detect the surrounding obstacles. The camera will be able to capture specific pattern on the road. The pattern is like a pathway for the modelled car that makes it easier to drive on roads in Bangladesh. Our prototype uses a hybrid combination of the existing technology as well as the newly implemented methodology of detecting special pattern marked on the road for providing better results.

Approval

The project report “**Auto self-driving car using Raspberry**” submitted by MD. Al - Amin ID: CSE 055 06842 to the Department of Computer Science & Engineering, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc.) in Computer Science & Engineering and as to its style and contents.

Board of Examiner’s Name, Signature and Date:

……………………………...	……………………………..	……………………………..
(Board Member 1)	(Board Member 2)	(Board Member 3)
Date:	Date:	Date:

Supervisor’s Signature and Date:

……………………………...

Supervisor Name

Date:

Declaration

We hereby, declare that the work presented in this Project is the outcome of the investigation performed by us under the supervision of **Adnan Ferdous Ashrafi,**

Senior lecturer, Department of Computer Science & Engineering, Stamford University Bangladesh. We also declare that no part of this Project and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Signature and Date:

……………………………...

Student Name:

Date

Dedicated to…...

Beloved parents, Teachers

And CSE Department

Stamford University Bangladesh

Acknowledgements

First of all, I'm thankful to the almighty God, The Merciful for blessing me with the determination and courage to face any difficulty during the completion of this work. I would also like to show our gratitude to our supervisor Senior Lecturer **Adnan Ferdous Ashrafi** for sharing his pearls of wisdom with me during the course of this project work and also for giving me the golden opportunity to do this wonderful project on Auto Self Driving Car by Raspberry During the course of this work, I have come across so many things and learned a lot of wonderful things. I am thankful to him for his continues support for my study, related research, for his patience, motivation, and immense knowledge. I could not have imagined having a better supervisor and mentor than Adnan Ferdous Ashrafi. Without his continuous support and encouragement this work would not be possible. I am immensely grateful to my parents and friends who helped me finishing this work and giving all the support that I needed to finish this work.

Table of Contents

List of Figures

List of Tables

List of listings

1 Introduction

A self-driving car (also known as an autonomous car or a driverless car) has no human input and can sense surrounding without any human interactions. Variety of sensors are combined and are used to identify the pathway, obstacles, pedestrians etc. from the surrounding. Benefit of having a driverless car is having reduced costs due to less wastage of fuel, increased safety, increased mobility, increased customer satisfaction etc. Safety benefits means we will be having reduced traffic collisions, lower accident rate reducing injuries and related costs such as, insurance. Automated cars could increase traffic flow by having proper routine mobility from source and destination, providing mobility for children, the elderly, disabled, and the poor who cannot operate the non-autonomous vehicles. Travelers could relieve their stress from driving and navigation issues when they go to an unknown city, reduce needs for parking space, lower fuel consumption, reduce crime and facilitate transportation as service like convert existing vehicles such as taxi, train buses to fully automate for the people. Between manually driven vehicles which are referred to as (SAE Level 0) and fully autonomous vehicles referred to as (SAE Level 5) we have wide range of vehicles that are classified in these SAE range. These are known as semi-automated vehicles. These were developed before fully automation could come in existence. These were iterative approach done for semi automating a car for example some parts get automated. These semi-automated vehicles could take some properties of fully automated vehicles, while still keeping driver in charge of the vehicle they operate. Since the cars rely on a preprogrammed code primarily. The traffic light, sudden pedestrians contact on road, is secondary data they need to process. Hence, they tend travelling slower for processing these extra scenarios. The vehicle sometime might have difficulty when determining certain objects such as light debris, trash, when humans such as police officers are signaling the vehicle to stop, spotting potholes on the road is also sometimes difficult hence avoiding them becomes not possible. Advantages could include higher speed limits with smoother rides since these have better control of vehicle and can view larger distance then a human high and can increase the roadway capacity and minimized(reduce) traffic congestion caused due to decreased need for safety gaps between vehicles travelling at higher speeds. Currently on highways drivers usually keep between 40 to 50 m (130 to 160 ft) distance away from the car in front of their pathway. These increases in highway capacity sometime are one of the main significant reason for impact in traffic congestion, particularly in the urban areas and more affected in highway congestion in some places. For the authorities to manage the traffic flow usually leads to increase the traffic congestion, with the extra data and predicting the driving behavior of people, we can combine these two details for reducing the traffic congestion the road with less need for traffic police on the roads and even for the road signage. Manually driven vehicles on online surveys are reported to be used only 4–5% time, while being parked and unused for the remaining 95–96% of the time. Autonomous vehicles, on the other hand, be continuously used even after it has travelled from some source to some destination for a given person. This could lead to reduce the need for parking space.

1.1 *Motivation*

In this research the goal to be able to make a Car with Raspberry Pi. This car will be able to detect an object and road sign, identify the pathway, obstacles, pedestrians etc. Our Goal is to make a both of them autonomous. Sensor will also be used to control the car by using Raspberry.

Another goal of this research project to build a low cost but productive car. The overall goal is to build an autonomous car which can be used in the future by modifying its functionality. After completing the project, we will find a car which can detect an object, road sign, identify the pathway, pedestrians etc.

1.2 *Sensors & Modules*

i. Ultrasonic Sensor (HC-SR04)
ii. Raspberry Pi Camera Module
iii. Motor Driver
iv. Mini DC Motors
v. Servo Motors
vi. Dc Motor Chassis
vii. Raspberry Pi 4

1.3 *Ultrasonic Sensor (HC-SR04)*

Ultrasonic sensing is one of the best ways to sense proximately and detect levels of high reliability. Our technical support gets emails all of the time about how our sensors work and what environments our sensors work (Or don’t work) in this guide was created as an introduction to ultrasonic sensing, its principle, and how ultrasonic sensors work in our applications.

1.3.1 Ultrasonic sensor

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transduce to send and receive ultrasonic pulses that relay back information about an object’s proximity. High-frequency sound waves reflect from boundaries to procedure distinct echo pattern.

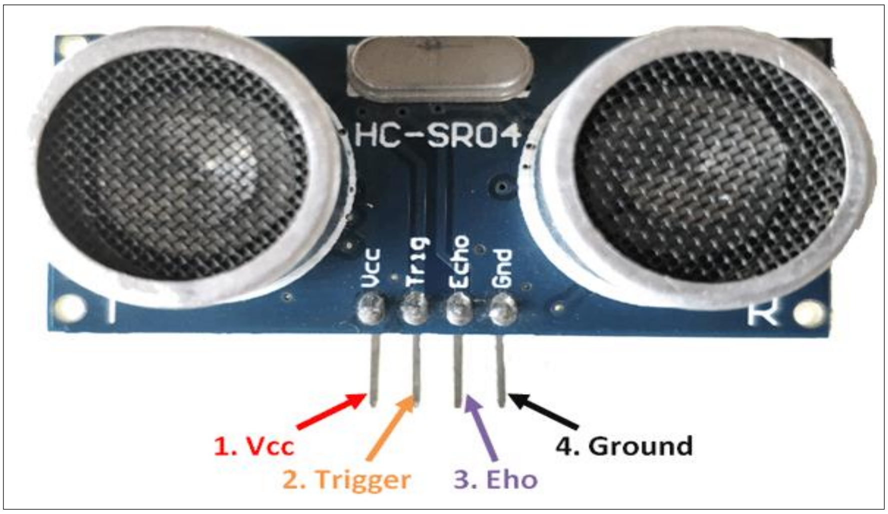


Figure 1.1: Ultrasonic Sensor HC-SR04

1.3.2 Ultrasonic sensor work

Ultrasonic sound vibrates at a frequency above the range of human hearing. Transducers are the microphone used to receive and send the ultrasonic sound. Our ultrasonic sensor, like many others, use a single transduce to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

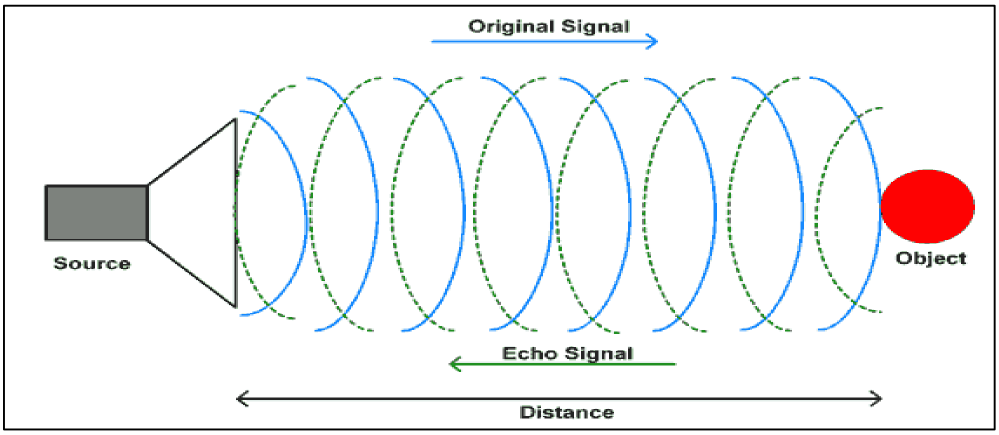


Figure 1.2: How ultrasonic sensor work

An ultrasonic sensor is as sensor which measures the distance of respective object by sending the sound wave of specific frequency. This sound wave is reflected after the collision with respective object and this wave is received by ultrasonic receiver. Distance is measured by calculating sending and receiving time of this sound wave. You may also like to read distance measuring using ultrasonic sensor and Raspberry pi.

1.3.3 Use an ultrasonic sensor

Ultrasound is reliable in any lighting environment and can be used inside or outside. Ultrasonic sensors can handle collision avoidance for a robot, and being moved often, as long as it isn't too fast. Ultrasonic are so widely used, they can be reliably implemented in grain bin sensing applications, water level sensing, drone applications and sensing cars at your local drive-thru restaurant or bank. Ultrasonic rangefinders are commonly used as devices to detect a collision.

Ultrasonic Sensors are best used in the non-contact detection of:

- Presence
- Level
- Position
- Distance

Non-contact sensors are also referred to as proximity sensors.

Ultrasonic are Independent of:

- Light
- Smoke
- Dust
- Color

Material (except for soft surfaces, i.e., woo, because the surface absorbs the ultrasonic sound wave and doesn't reflect sound.)

1.3.4 Advantages of Ultrasonic sensor

- The ultrasonic sensor has high frequency, high sensitivity and high penetrating power therefore it can easily detect the external or deep objects.
- These sensors easily interface with micrometer or any type of controller.
- These sensors have greater accuracy then color methods for measuring the thickness and depth of parallel surface.
- These sensors could easily sense the nature, shape and orientation of that specific objects which is within the area of these sensors.
- These sensors are easy to use, not dangerous during operation for nearby objects, person, equipment or material.

1.3.5 Disadvantages to Ultrasonic Sensors

- During use of ultrasonic tester for testing it is very important to known the operational manual, is words it required careful attention for experienced technician.
- During the development inspection procedure of equipment using ultrasonic tester extensive knowledge is required is testing technician.
- When these sensors are used for inspection purpose then these should be water resistive otherwise, they could be damaged.

- When these sensors are interfaced with microcontroller or any controller then experienced person or programmer is required

1.4 *Raspberry Pi Camera Module*

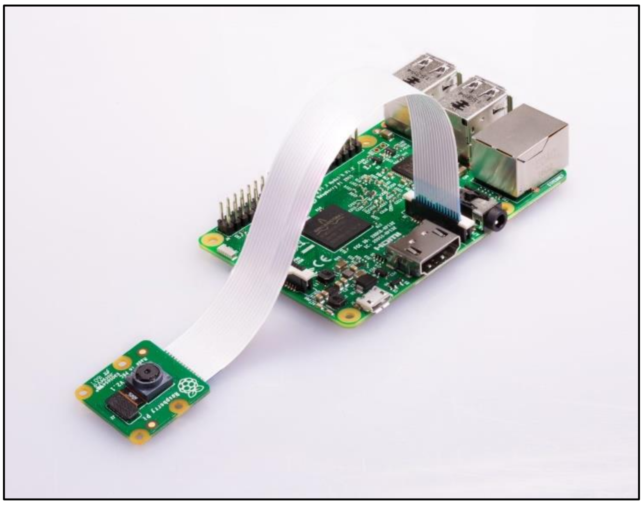


Figure 1.3: Pi Camera module

The Raspberry Pi Camera v2 is a high quality 8-megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464-pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras. The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable. The high-quality Sony IMX219 image sensor itself has a native resolution of 8 megapixel, and has a fixed focus lens on-board. In terms of still images, the camera is capable of 3280 x 2464-pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video

1.5 *Motor Driver*

This L298N Motor Driver Module is a high-power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control. The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit.

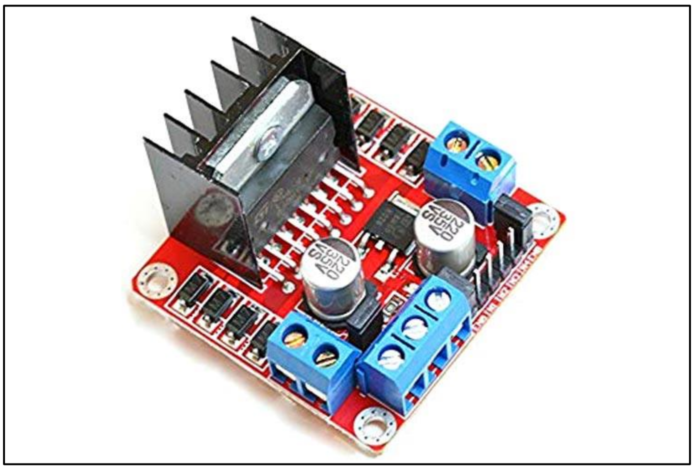


Figure 1.4: Dc motor driver l298n

78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry. ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B. Internal circuit diagram of L298N Motor Driver module is given below:

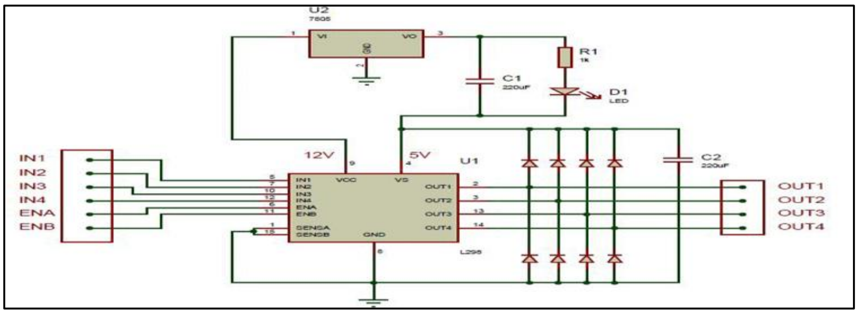


Figure 1.5: L298 mechanical drawing

1.5.1 L298 Module Features & Specifications:

- Driver Model: L298N 2A
- Driver Chip: Double H Bridge L298N
- Motor Supply Voltage (Maximum): 46V
- Motor Supply Current (Maximum): 2A
- Logic Voltage: 5V
- Driver Voltage: 5-35V
- Driver Current:2A
- Logical Current:0-36mA
- Maximum Power (W): 25W
- Current Sense for each motor
- Heat sink for better performance

- Power-On LED indicator

1.6 *Mini DC Motors*

- Operating Voltage: 3, 4.5, 5 & 12 Vdc
- No Load Speed: 90 RPM
- No Load Current: 190mA (250mA max)
- Max. Torque: 800 gf.cm
- Gear Ratio: 48:1

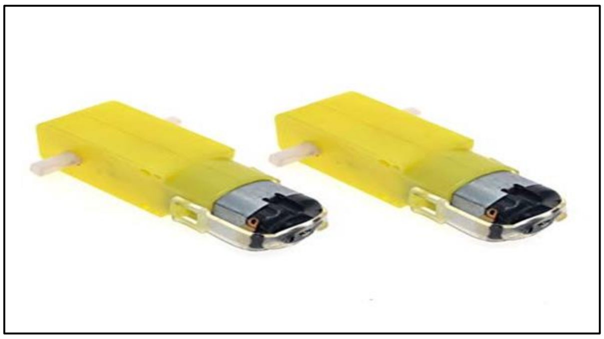


Figure 1.6: Mini DC Motors

1.7 *Servo Motors*

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a servo mechanism. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. For this tutorial, we will be discussing only about the DC servo motor working. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc.

Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity. The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.

Figure 1.6: Servo Motor

1.7.1 Servo Motor Working Mechanism

It consists of three parts:

- Controlled device
- Output sensor
- Feedback system

It is a closed-loop system where it uses a positive feedback system to control motion and the final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here reference input signal is compared to the reference output signal and the third signal is produced by the feedback system. And this third signal acts as an input signal to the control the device. This signal is present as long as the feedback signal is generated or there is a difference between the reference input signal and reference output signal. So, the main task of servomechanism is to maintain the output of a system at the desired value at presence of noises.

1.7.2 Servo Motor Working Principle

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, we use gear assembly to reduce RPM and to increase torque of the motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in terms of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with the potentiometer and as the motor rotates so the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

1.7.3 Interfacing Servo Motors with Microcontrollers:

Interfacing hobby Servo motors like s90 servo motor with MCU is very easy. Servos have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU. An **MG995 Metal Gear Servo Motor** which is most commonly used for RC cars humanoid bots etc. The picture of MG995 is shown below:

Figure 1.7: Servo Motor2

Figure 1.8: Servo Motor3

The color coding of your servo motor might differ hence check for your respective datasheet. All servo motors work directly with your +5V supply rails but we have to be careful on the amount of current the motor would consume if you are planning to use more than two servo motors a proper servo shield should be designed.

1.7.4 Controlling Servo Motor:

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU.Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically, servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.

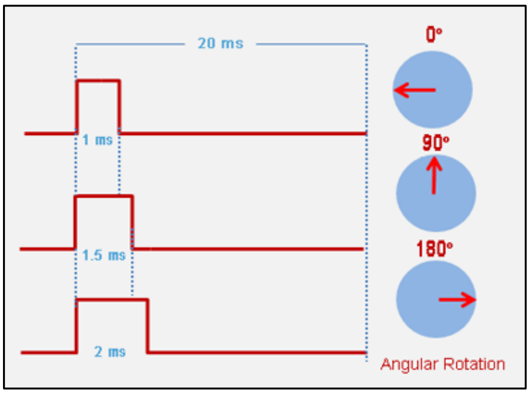


Figure 1.9: Controlling Servo Motor

Servo motor can be rotated from 0 to 180 degrees, but it can go up to 210 degrees, depending on the manufacturing. This degree of rotation can be controlled by applying the Electrical Pulse of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. The pulse of 1 ms (1 millisecond) width can rotate the servo to 0 degrees, 1.5ms can rotate to 90 degrees (neutral position) and 2 ms pulse can rotate it to 180 degree.

All servo motors work directly with your +5V supply rails but we have to be careful about the amount of current the motor would consume if you are planning to use more than two servo motors a proper servo shield should be designed.

1.8 *Dc Motor Chassis*

Make your vehicle dreams a reality with the 4-wheel drive DC motor car chassis kit.

This kit gives you everything you need to build the shell of a 4-wheel-drive Mobile Platform Robot! You get the metal plates that make up the chassis, four DC drive motors with matching wheels. You’ll fill in the rest with a power supply, microcontroller board, and motor controller. The differential drive (four separately driven wheels) allows for a near zero turning radius while the high-strength aluminum alloy body plus high-quality high-speed motors make it suitable for flat indoor surfaces. Also, it’s adorably small!

Kit includes:

- 4x Drive Motors & 4x wheels (drive with 3-6VDC, 200-400 mA run, 1.5A hard stall)
- Anodized aluminum frame and all mounting hardware for assembly

Figure 1.10: Dc Motor Chassis

1.9 ***Raspberry Pi 4***

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

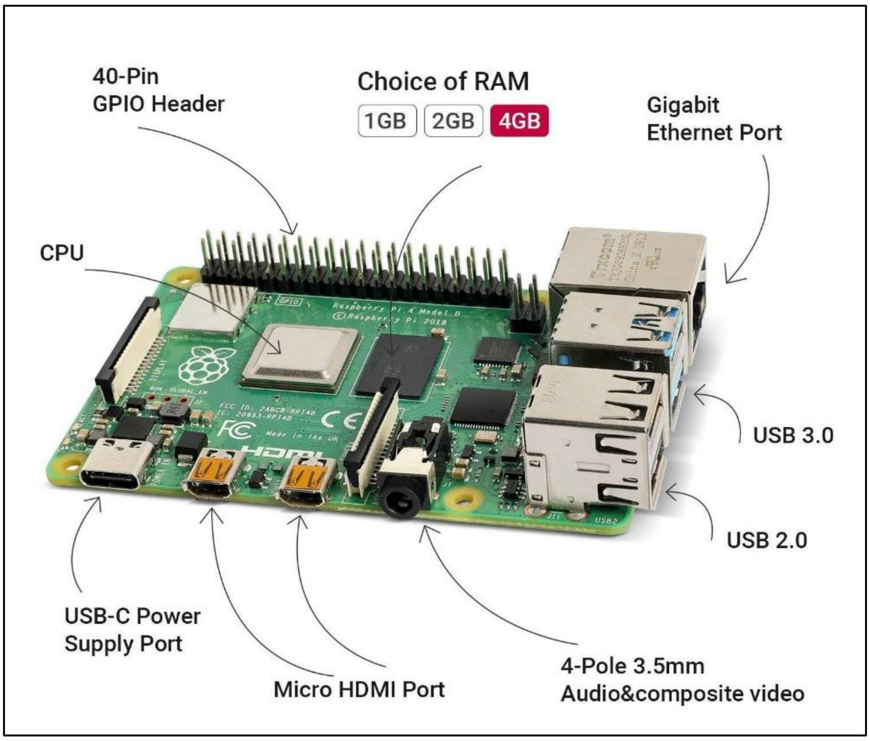


Figure 1.11: Raspberry Pi 4

1.9.1 Raspberry Pi 4 pin out

1. A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. A 40-pin GPIO header is found on all current Raspberry Pi boards (unpopulated on Pi Zero and Pi Zero W). Prior to the Pi 1 Model B+ (2014), boards comprised a shorter 26-pin header.

2. This section is solely dedicated to the Raspberry Pi 4 GPIO Pin out:

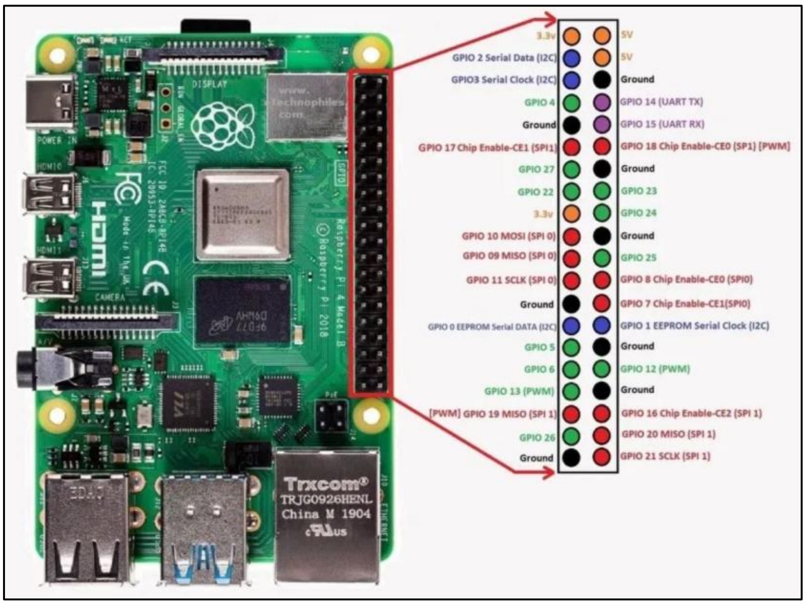


Figure 1.12: Raspberry Pi 4 pin out

Raspberry Pi GPIO stands for General Purpose Input Output pins. These pins are used to connect the Raspberry pi board to external input/output peripheral devices.

1.9.2 Power Pins on Raspberry Pi 4:

The raspberry pi 4 model B board consists of two 5V pins, two 3V3 pins, and 7 ground pins (0V).

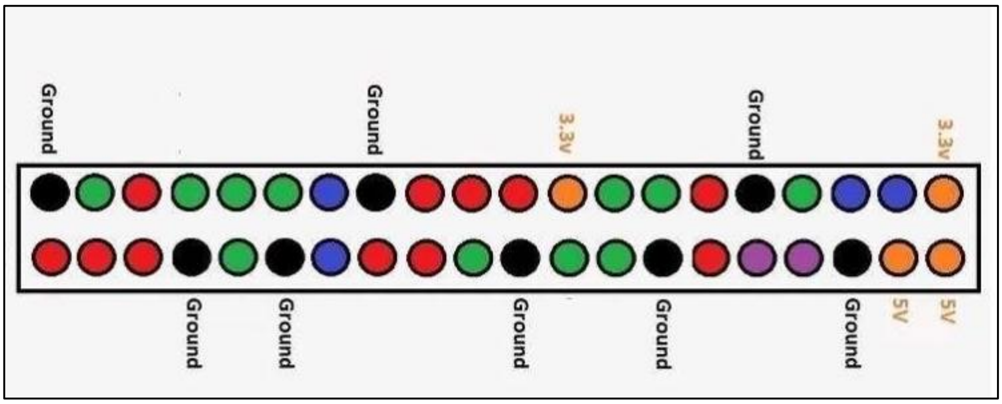


Figure 1.13: Power Pins on Raspberry Pi 4

5V: The 5v pin outputs the 5 volts coming from the USB Type-C port.

3.3V: The 3v pin is used to provide a stable 3.3v supply to external components.

GND: The ground pin is commonly referred to as GND.

1.9.3 R-Pi 4 Global Input/Outputs pins:

A pin that can be set as an input or output and is controlled in run time is called a GPIO pin

A GPIO pin set as **input** allows the signal transmitted by any external device (connected to this pin) to be received by the Raspberry Pi.

Input voltage between 1.8V and 3.3V is read as HIGH by the Raspberry pi. And when the input voltage is lower than 1.8V, it is read as LOW.

Note: Do not connect an external device with an output voltage above 3.3V to any of the GPIO pins, or else it will fry your Raspberry Pi board.

A GPIO pin set as **output** delivers HIGH/3.3V or LOW/0V.

Apart from Input/Output, the GPIO pins can also perform a variety of other functions like PWM. Some of these functions/pins are:

1.9.4 PWM (pulse-width modulation) pins:

PWM stands for "Pulse Width Modulation". It means that an analog value is being modulated on a digital signal. Software PWM is available on all pins. Hardware PWM is available on these pins only: **GPIO12, GPIO13, GPIO18, GPIO19V**

SPI pins on RPi 4:

SPI (Serial Peripheral Interface) is a type of serial communication protocol. It is used by the Raspberry Pi for master-slave communication to quickly communicate between one or more peripheral devices.

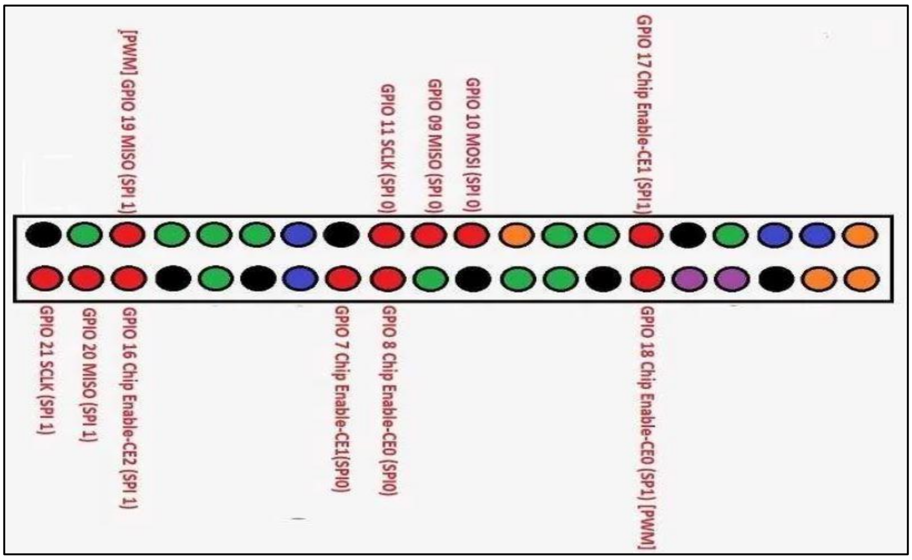


Figure 1.14: SPI pins on Raspberry Pi 4

The data is synchronized using a clock (**SCLK** at GPIO11) from the master (RPi).The Pi sends this data to an SPI device using **MOSI** (Master Out Slave In) pin. And when the SPI device needs to communicate back to the Raspberry Pi, it sends the data back through the **MISO** (Master In Slave Out) pin.

5 pins are required for SPI communication:

GND: Connect the GND pin from all the slave components and the Raspberry Pi 4 board together.

SCLK: Clock for SPI communication.

MOSI: It stands for Master Out Slave In. This pin is used to send data from the master to a slave.

MISO: It stands for Master In Slave Out. This pin is used to receive data from a slave to the master.

CE: It stands for Chip Enable. We need to connect one CE pin per slave (or peripheral devices) in our circuit. By default, we have two CE pins but we can configure more CE pins from the other available GPIO pins.

SPI pins on Raspberry Pi:

SPI0: GPIO9 (MISO), GPIO10 (MOSI), GPIO11 (SCLK), GPIO8 (CE0), GPIO7 (CE1)

SPI1: GPIO19 (MISO), GPIO20 (MOSI), GPIO21 (SCLK), GPIO18 (CE0), GPIO17 (CE1), GPIO16 (CE2)

I2C Pins on RPi 4:

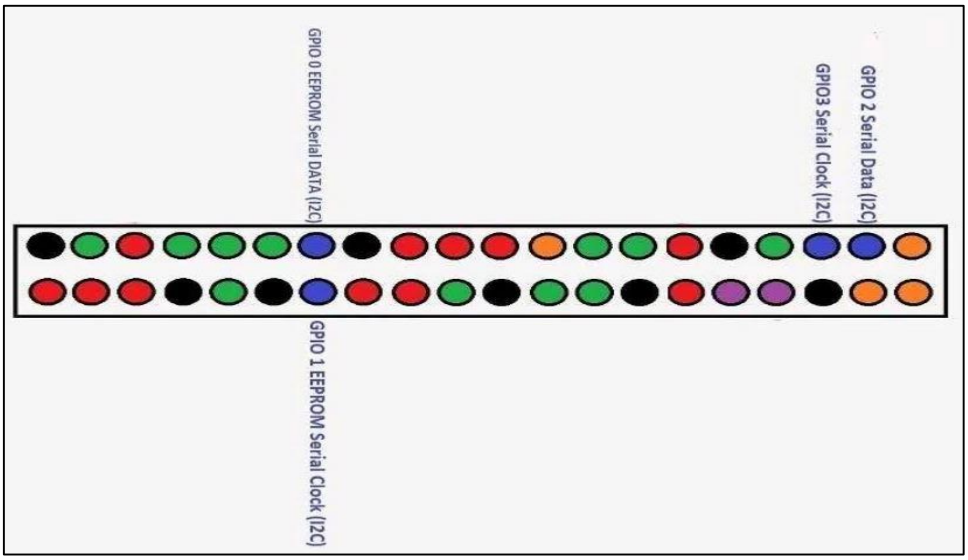


Figure 1.15: Raspberry Pi 4 I2C pins

I2C pins on the Raspberry Pi board are used to communicate with peripheral devices that are compatible with **Inter-Integrated Circuit** (a low-speed two-wire serial communication protocol).

This serial communication protocol requires master-slave roles between both, the board and the peripheral devices.

I2C protocol requires two connections: **SDA (Serial Data)** and **SCL (Serial Clock)**. They work by transmitting data using the SDA connection, and the speed of data transfer is controlled via the SCLK pin.

Data: (GPIO2), Clock (GPIO3)

EEPROM Data: (GPIO0), EEPROM Clock (GPIO1)

UART pins on RPi 4:

The UART (Universal Asynchronous Receiver / Transmitter) is an asynchronous protocol that provides a way to communicate between two microcontrollers or devices.

TX pin transmits the serial data to the RX pin of another device and RX pin receives the serial data coming from TX pin of the other device.

TX : GPIO14

RX : GPIO15

1.10 *Chapter Summary*

Well, there are many things you can do with Raspberry Pi 4 pins. I'll repeat it here: you can never be too cautious when manipulating the pins! A mistake can destroy your board in less than a second. But if you pay attention and double check everything, there is no reason you'll burn anything.

Now, if you feel lost with so much information and don't know where to start, here's a list of steps you can take from there:

- Get some simple examples and do them, read the value from a button, etc. For that you'll need to create a small circuit on a breadboard, and you'll use the pins with their primary function (GPIO).
- Once you're familiar with how basic circuits work (GND, Vcc, and communication pins), try to get a more complex sensor, for example an I2C accelerometer, so you can measure whether your board is on a flat surface or not.
- After you know how to communicate with one sensor, try to communicate between your Raspberry Pi board and another Raspberry Pi/Arduino/Computer, using all 3 protocols: UART, I2C, SPI. You'll learn a lot by doing that.
- And then, there is no secret! Find yourself a personal project, and you'll learn more stuff along the way.

2 Related work

2.1 *Combinations like Arduino and Raspberry-pi by Luiz.*

The concept of self-driving cars has been around for a while now. Different controller combinations like Arduino and Raspberry-pi have been used to distribute the work load. This is because different real-time tasks like image processing distance calculation, localization, decision making etc. are needed to be done continuously with precision and accuracy, hence having a threat of hardware failure [1][8]. For localization and mapping purpose, a dynamic feature extraction technique for mapping of point cloud for generation of 3D cognitive maps has been used with the help of scene recognition and alignment without any initial pose estimation requirement. It surpasses the typical approach of localization through a particle filter [2]. The 2D neural network based map learning techniques have also proven to be effective for path planning. In this approach different images of the environment are trained with their position as the label. The car then clicks a picture of the environment dynamically through a camera module and finds its current location. But this technique gives high time and space complexity which is not affordable [4]. Work has been done in path planning using trajectory generation and evaluation techniques which have proven to be useful on curved roads and mountain passes to obtain an optimal solution provided the environmental information is available beforehand [3]. Exploration of car designs which work on solar power has also been done. It solves the problem of continuous energy supply required by the car and its in-built hardware. The overtaking mechanism depending on the measurements obtained from the environment, along with its mathematical computation has been introduced by Luiz G. Hafemann, Robert Sabourin, Luiz S. Oliveria [6]. PSO based search algorithm to find single source multiple destination shortest paths using an particle encoding decoding scheme for exploitation of sub optimality features of the shortest path is provided [7]. Ultrasonic distance sensor and a buzzer is used in building a system to execute obstacle detection and alerting. The system detects the obstacle within a certain distance threshold. This system can be integrated in cars to avoid collision[5].

2.2 *Autonomous Self-driving by H. Mujtaba*

In another paper, a method to make a self-driving robot car is presented [9]. The different hardware components and their assembly are clearly described. A novel method to determine the uneven, marked or unmarked road edges is explained in details relying upon OpenCV. Using ultrasonic sensors, the collisions with obstacles is avoided. The algorithm mentioned in the paper has been successfully implemented on a small autonomous car.

2.3 *Autonomous car*

In their project deals with building an autonomous car that is capable of travelling safely and intelligently avoiding the risk of human errors. This raspberry Pi based project is able to detect the obstacles & traffic light. It is able to compare the data processed with the data provided to it and is able to take an intelligent decision whether to stop or continue on its present path. Important components involved in this project are - the hardware platform which includes raspberry pi board, all the hardware like pi camera and the ultrasonic sensor for improved efficiency & the camera used along with an ultrasonic sensor to provide necessary data from the world for real time processing and application. Second being the cloud platform which will be basically used to train our raspberry pi board for real time applications. Cloud helps us to test as well as train better tracking and decision models & helps in providing the offline computing and storage capabilities for vehicle. Basically it will be used to train the processor to differentiate between positive(green signal) and negative (red signal) images using various thousands of such signal images as an example .The third and most important part includes the algorithms for perception ,control, localization and recognition.[8]

3 System Design

The project aims to build a monocular vision autonomous car prototype using Raspberry Pi as a processing chip. A Pi camera along with an ultrasonic sensor is used to provide necessary data from the real world to the car. The car is capable of reaching the given destination safely and intelligently thus avoiding the risk of human errors. Many existing algorithms like obstacle detection are combined together to provide the necessary control to the car

The project aims to build a monocular vision autonomous car prototype using Raspberry Pi as a processing chip. A Pi camera along with an ultrasonic sensor is used to provide necessary data from the real world to the car. The car is capable of reaching the given destination safely and intelligently thus avoiding the risk of human errors. Many existing algorithms like obstacle detection are combined together to provide the necessary control to the car

3.1 *Pin Definition*

In the Figure 3.1, the pin definition of Raspberry Pi is shown and in the Table 3.1 a detailed pin description is given.

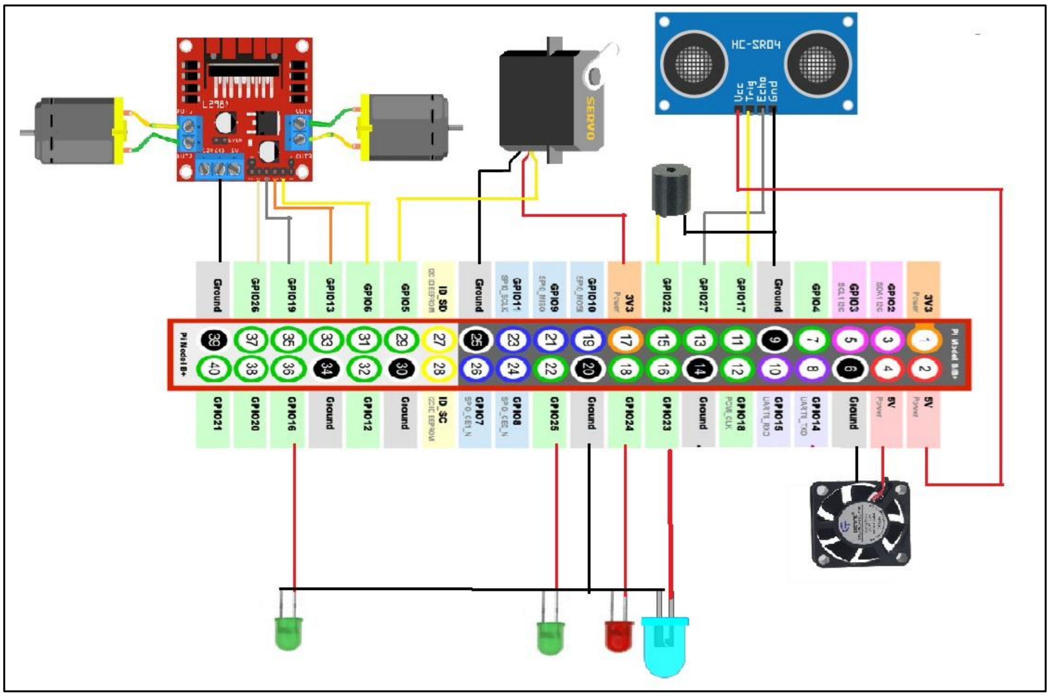


Figure 3.1: Pin Connection diagram

3.2 *Obstacle detection sub-system algorithm*

Image processing sub-system result It is important to detect the LANE from the image of road. And check their position in the form of pixel coordinates for the decision of turn. Following picture shows the image captured by the camera.

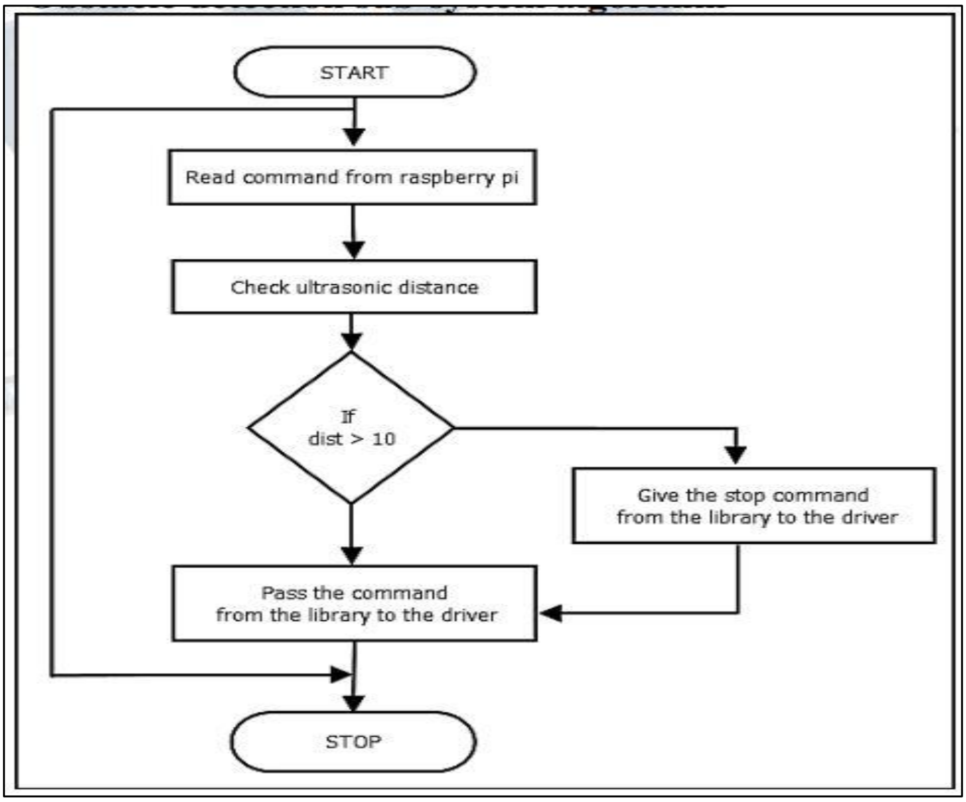


Figure 3.2: Obstacle detection sub-system algorithm

Table 3.1: Pin Description of RaspberryPi to motor driver

COMMAND	GPIO pin 1	GPIO pin 2
STOP	LOW	LOW
FORWARD	HIGH	HIGH
LEFT	HIGH	LOW
RIGHT	LOW	HIGH

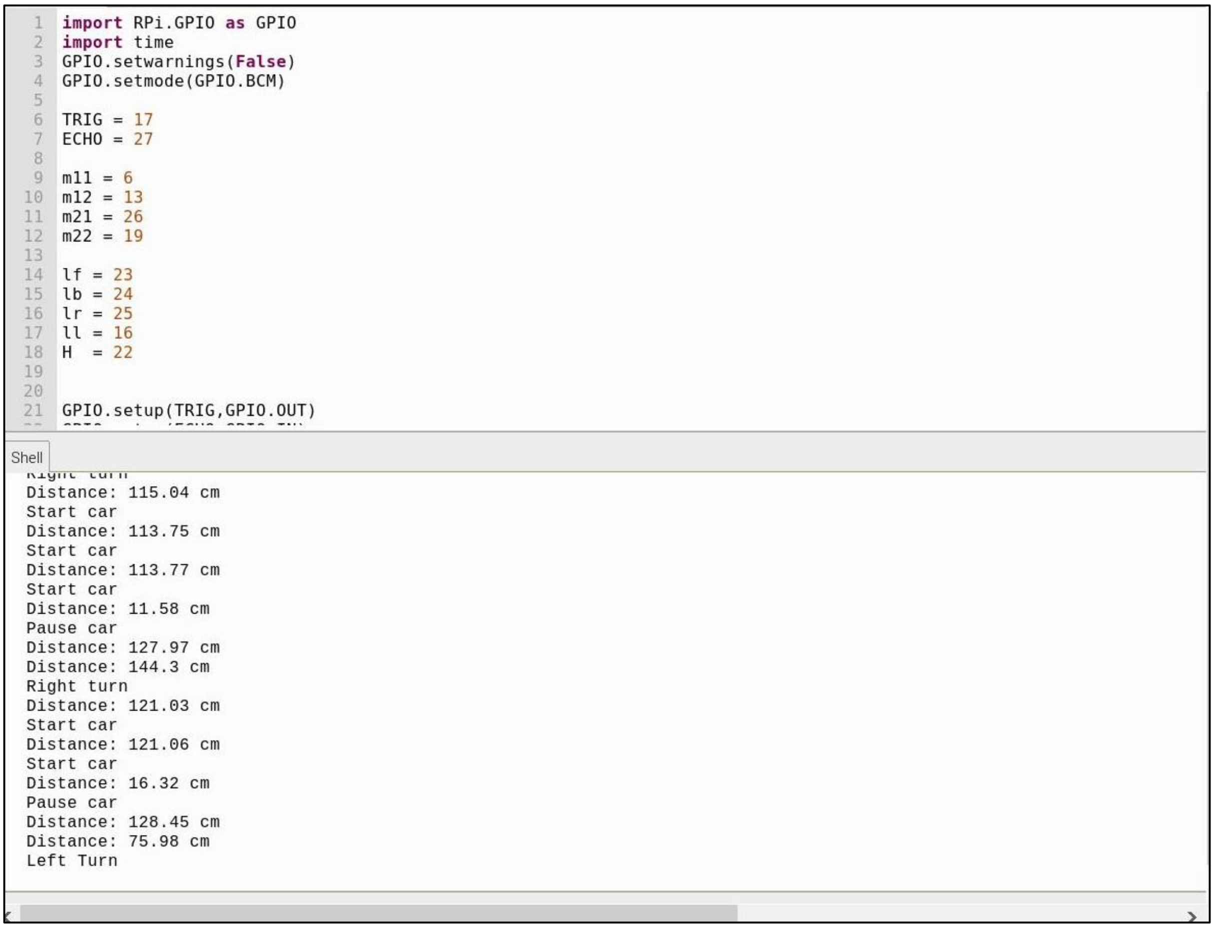
Table 3.2: Parameters of RaspberryPi 4

Serial	Properties
1.	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
2.	2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
3.	GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
4.	Gigabit Ethernet
5.	2 USB 3.0 ports; 2 USB 2.0 ports.
6.	Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
7.	2 × micro-HDMI ports (up to 4kp60 supported)
8.	2-lane MIPI DSI display port
9.	2-lane MIPI CSI camera port
10.	4-pole stereo audio and composite video port
11.	H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)

12.	OpenGL ES 3.1, Vulkan 1.0
13.	Micro-SD card slot for loading operating system and data storage
14.	5V DC via USB-C connector (minimum 3A*)
15.	5V DC via GPIO header (minimum 3A*)
16.	Power over Ethernet (PoE) enabled (requires separate PoE HAT)
17.	Operating temperature: 0 – 50 degrees C ambient
18.	A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

3.3 *Ultrasonic sensor*

We can describe this figure, where ultrasonic sensor pin and measurement results TRIG =17 mean GPIO 17 pin and ECHO = 27 mean GPIO 27 pin. And down part of this image measurement distance.



```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

TRIG = 17
ECHO = 27

m11 = 6
m12 = 13
m21 = 26
m22 = 19

lf = 23
lb = 24
lr = 25
ll = 16
H  = 22


GPIO.setup(TRIG,GPIO.OUT)
```

Shell

```
Distance: 115.04 cm
Start car
Distance: 113.75 cm
Start car
Distance: 113.77 cm
Start car
Distance: 11.58 cm
Pause car
Distance: 127.97 cm
Distance: 144.3 cm
Right turn
Distance: 121.03 cm
Start car
Distance: 121.06 cm
Start car
Distance: 16.32 cm
Pause car
Distance: 128.45 cm
Distance: 75.98 cm
Left Turn
```

Figure 3.3: Ultrasonic sensor measurement distance

3.4 *Obstacle*

In this figure we can see two part, left and right part. When autonomous car ride left side on the road and see a man on left side then stop the car and slide right side from obstacle then crossing obstacle again move their main position on their road and continue running.

In this alternative side , when autonomous car ride right side on the road and see a man on right side then stop the car and slide left side from man then crossing obstacle again move their main position on their road and continue running.

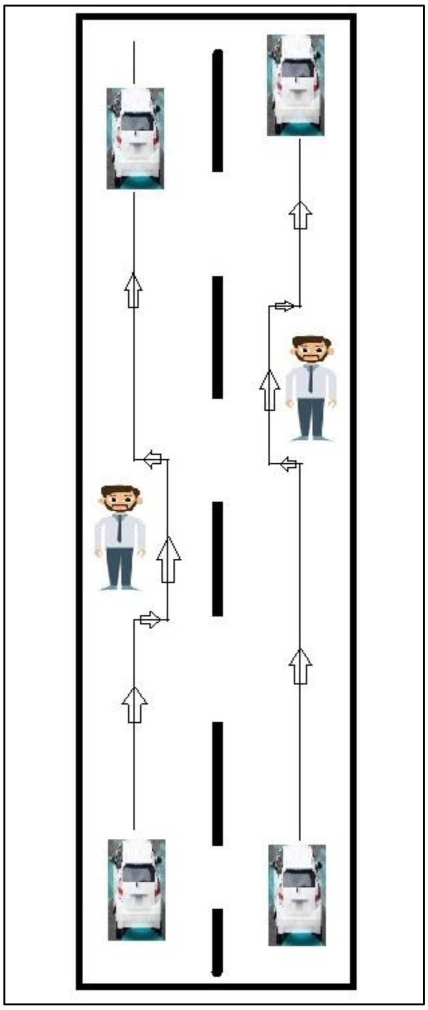


Figure 3.4: Obstacle position ride car

3.5 *Left side obstacle*

In this figure we can describe two part one part code obstacle size and another part code there position, when (xmax-xmin)>0.40 and (ymax-ymin)>0.40: there coordinator measurement then autonomous car decision whit I do? Then again measurement x and y coordinator if (xmin<0.3<xmax) and (ymin<0.3<ymax): when car turn right side 0.9 (personal desition) sec.

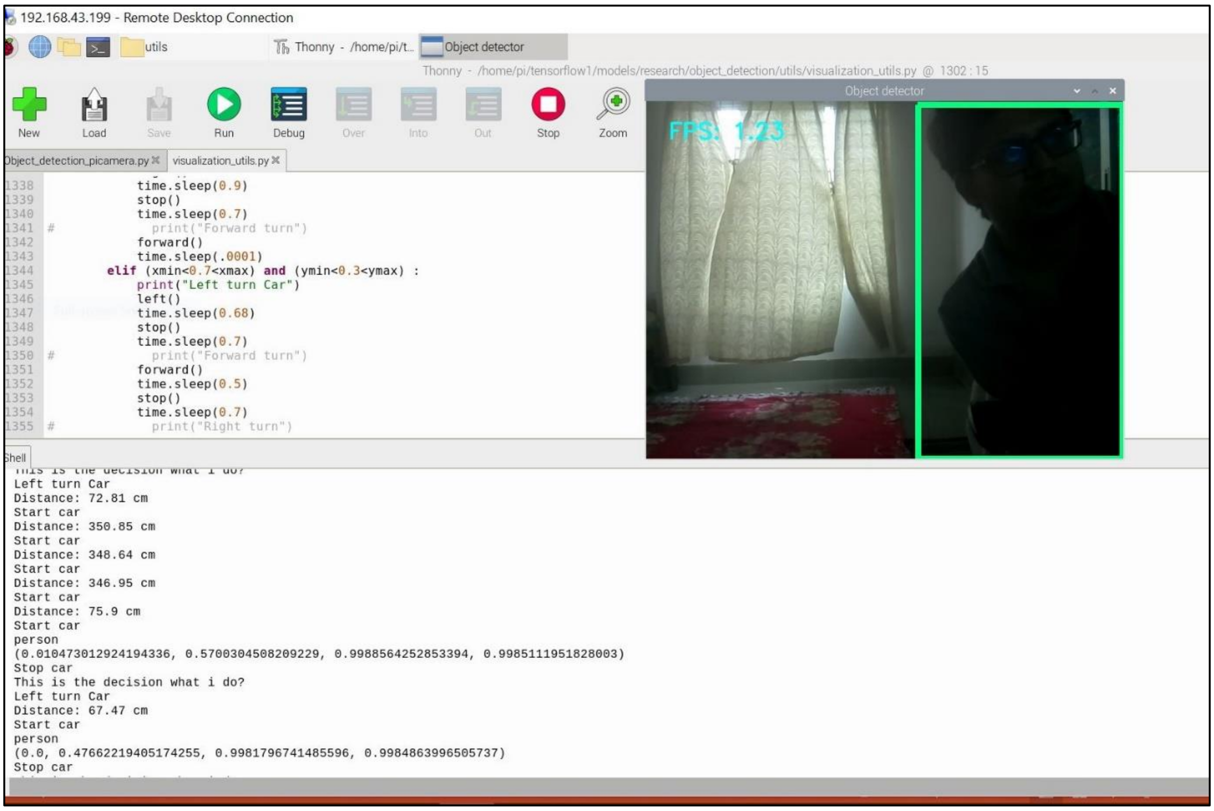


Figure 3.5: When person stay left side

3.6 ***Right side obstacle***

In this figure we can describe two part one part code obstacle size and another part code there position, when (xmax-xmin)>0.40 and (ymax-ymin)>0.40: there coordinator measurement then autonomous car decision whit I do? Then again measurement x and y coordinator if (xmin<0.3<xmax) and (ymin<0.7<ymax): when car turn left side 0.9 (personal desition) sec.

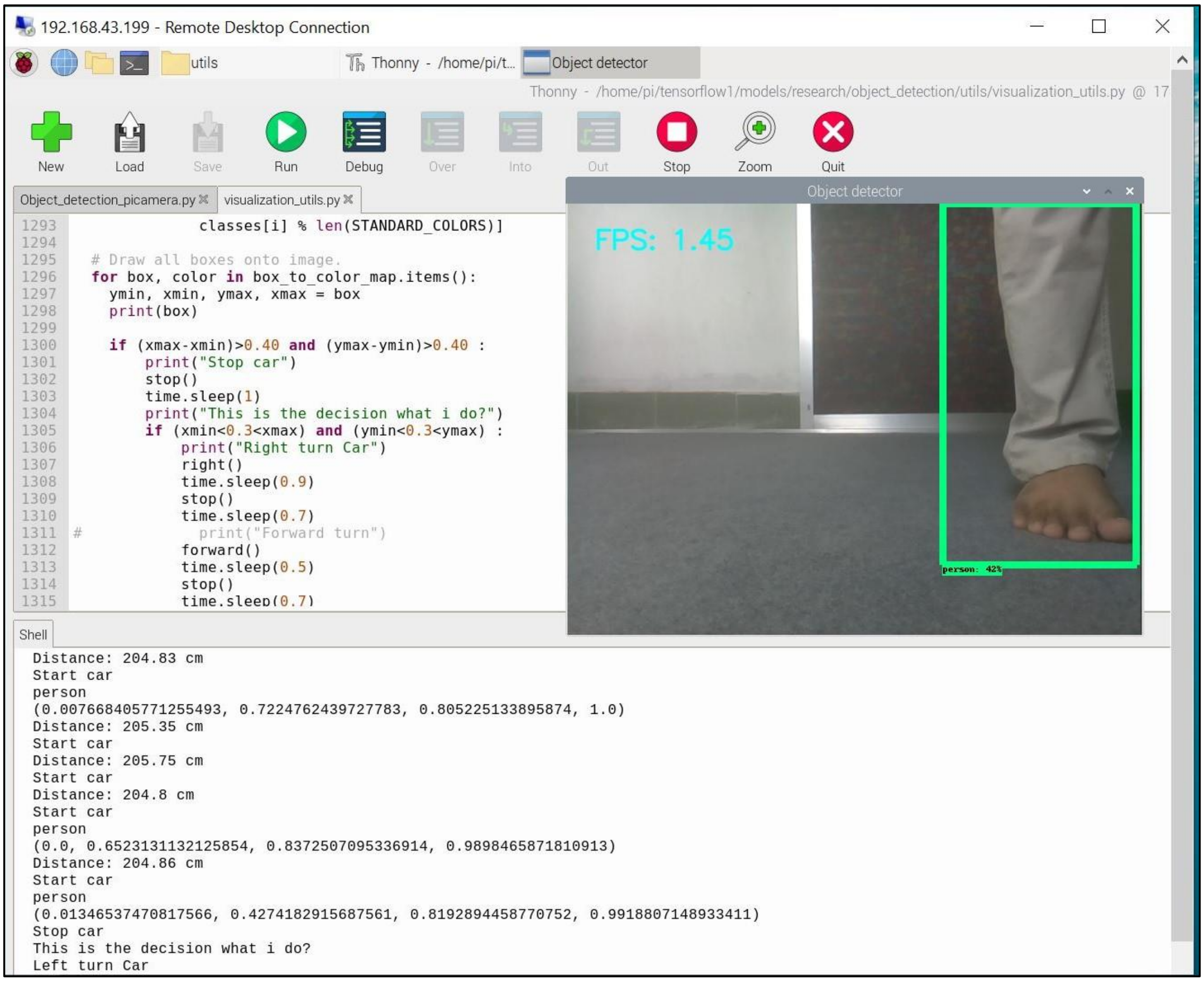


Figure 3.6: When person stay right side

If we consider there are four coordinator. Another two coordinator concept will be describe two part, One part code obstacle size and another part code there position, when (xmax-xmin)>0.40 and (ymax-ymin)>0.40: there coordinator measurement then autonomous car decision whit I do? Then again measurement x and y coordinator if (xmin<0.7<xmax) and (ymin<0.3<ymax): when car turn right side 0.9 (personal desition) sec.

Another coordinator Then again measurement x and y coordinator if (xmin<0.7<xmax) and (ymin<0.7<ymax): when car turn right side 0.9 (personal desition) sec.

3.7 *Traffic Signal detected*

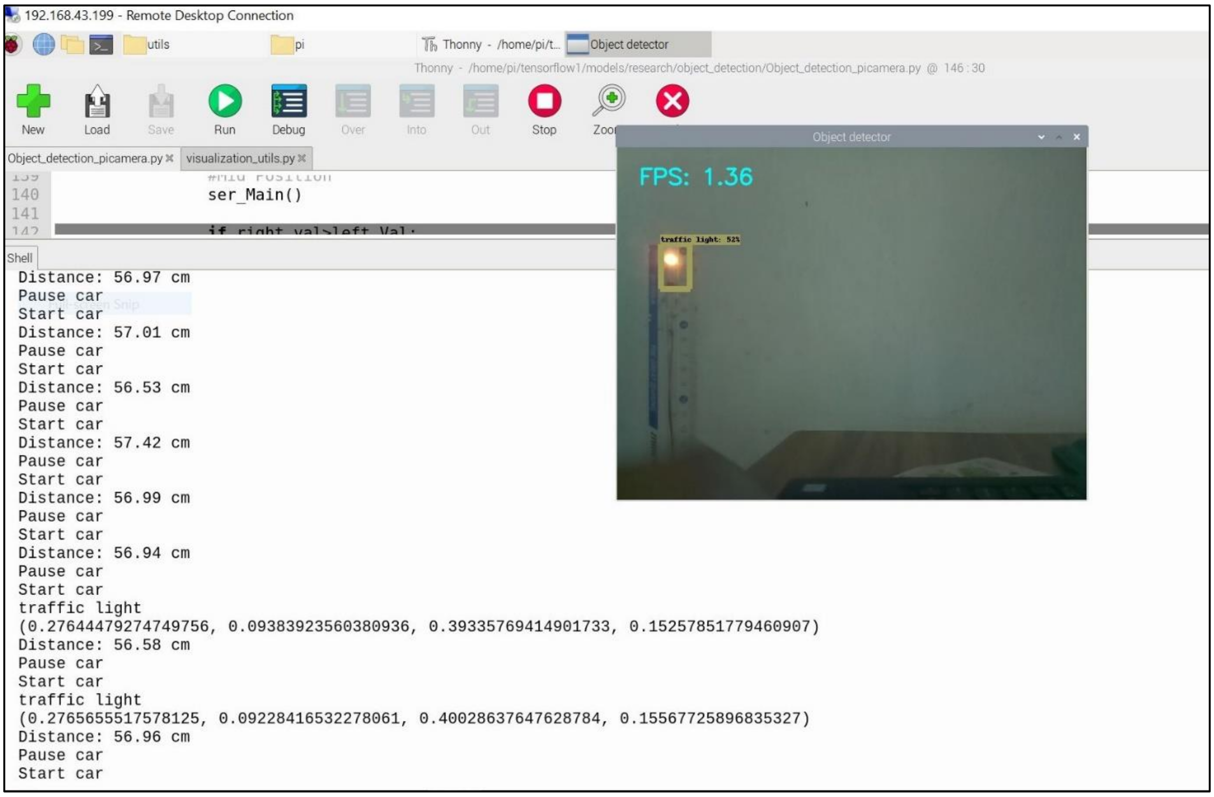


Figure 3.7: Traffic Light detection

It can be many object detect in this autonomous self-driving car. Such as chair, table, laptop, bed, etc.

4 Implementation

Our prototype model shows some work on both the application that we have discussed in this paper. The following set of figures shows the prototype Mobile Robot (Vehicle) used in the construction of the model. Our main focus was on Following Vehicle, which detects and avoids obstacles, coordinate with environment, get route and follow the route.

4.1 *Autonomous self-driving car using Raspberry Pi 4*

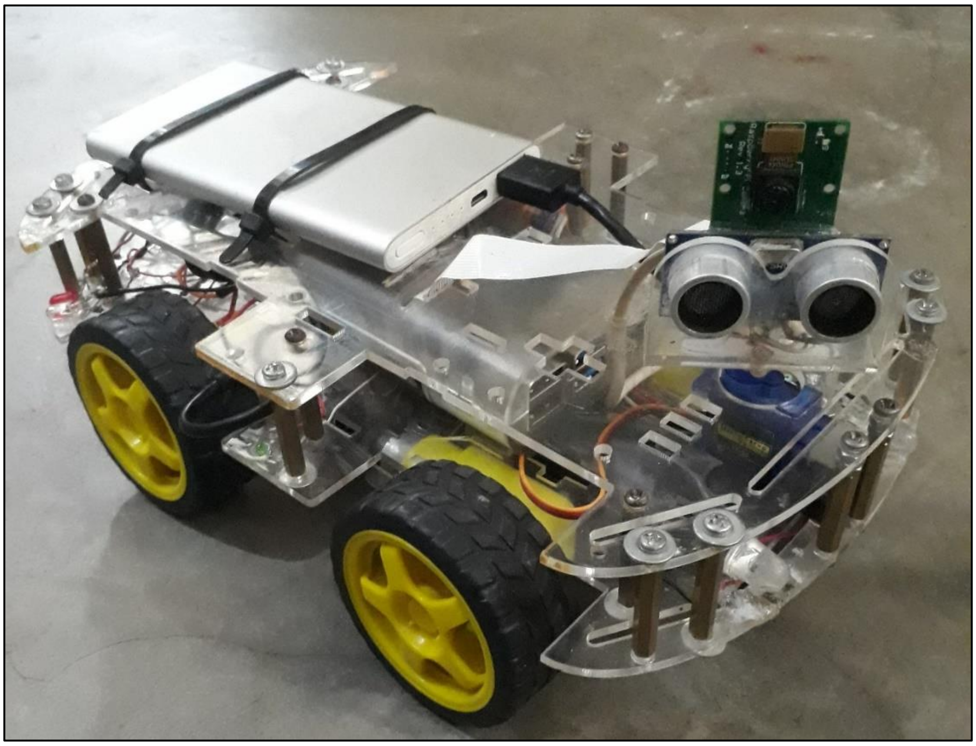


Figure 4.1: Top view autonomous car

We connected required components needed for the motor driver and then there after we connected the output of the raspberry to the motor driver, which would be used to output specify power to modeled car to control its speed. Once the sensors were implemented in the model, the camera which was going capture the video footage, to detect the pattern was connected to the Raspberry pie. The processing of the image was done remotely on the external system. Hence using the radio waves, we send the data from the raspberry pi to the system, process the data and then send the required output back to the raspberry pie. The system will process on determining what obstacles are detected and what it should when it detects an obstacle in the environment. The obstacle could be another vehicle or pedestrians crossing the road. The remote system can also determine what speed the modeled car should travel, what direction it should travel following the specific pattern (pathway) provided for the modeled car on the road.

4.2 ***Sensor Connection***

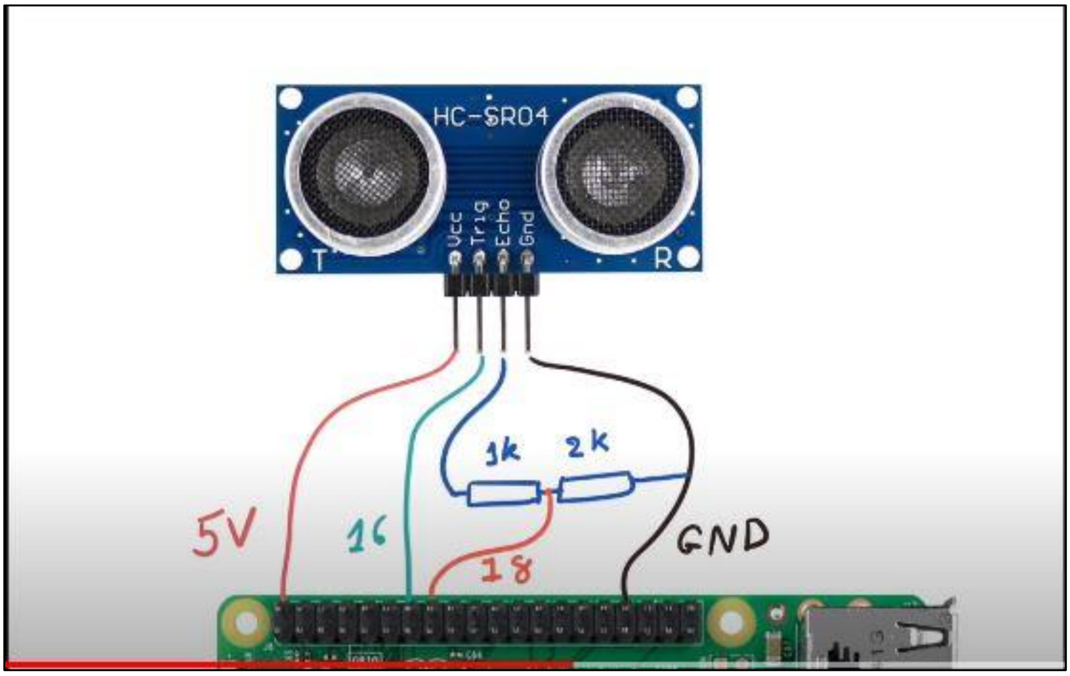


Figure 4.2: Ultrasonic sensor connection

In this figure we can found Ultrasonic sensor Four pin Vcc, Trig, Echo, Gnd. Vcc(+ve) pin connect 5v power pin (2number pin). Trig pin connect GPIO 17, and Echo connect 1K register then GPIO 27, and Gnd connect 2K register then raspberry pi Ground pin, Then 1K and 2K register connected.

4.3 ***Real-Time Object Detection Using Tensorflow***

Object Detection using Tensorflow is a computer vision technique. As the name suggests, it helps us in detecting, locating, and tracing an object from an image or a video. Let us gain a deeper understanding about how object detection works, what is Tensorflow, and more.

i. Object detection
ii. Object detection work
iii. Tensorflow
iv. Tensorflow object detection API
v. Object Detection Using Tensorflow
vi. Real-Tim Object detection using Tensorflow

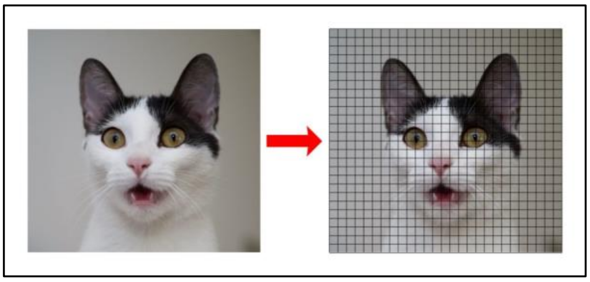
4.3.1.1 Object detection

Object detection is a computer vision technique in which a software system can detect, locate, and trace the object from a given image or video. The special attribute about object detection is that it identifies the class of object (person, table, chair, etc.) and their location-specific coordinates in the given image. The location is pointed out by drawing a bounding box around the object. The bounding box may or may not accurately locate the position of the object. The ability to locate the object inside an image defines the performance of the algorithm used for detection. Face detection is one of the examples of object detection.

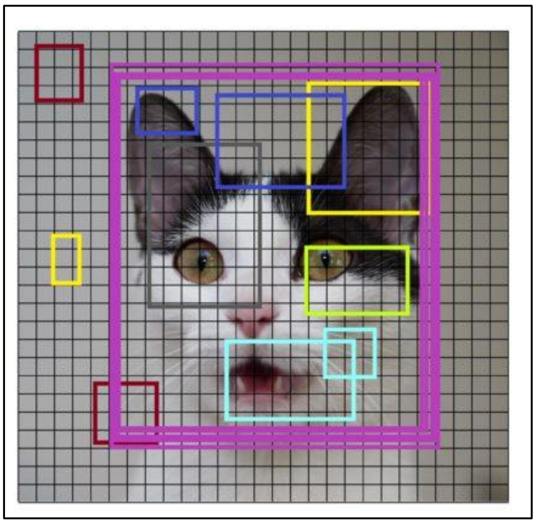
These object detection algorithms might be pre-trained or can be trained from scratch. In most use cases, we use pre-trained weights from pre-trained models and then fine-tune them as per our requirements and different use cases.

4.3.1.2 Object detection work

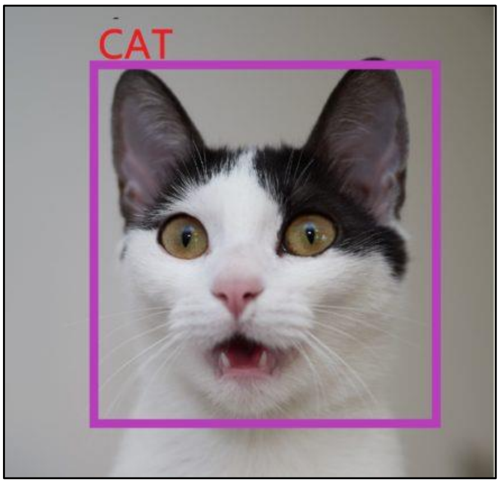
Generates the small segments in the input as shown in the image below. As you can see the large set of bounding boxes are spanning the full image



Feature extraction is carried out for each segmented rectangular area to predict whether the rectangle contains a valid object.



Overlapping boxes are combined into a single bounding rectangle (Non-Maximum Suppression)



4.3.1.3 Tensorflow

Tensorflow is an open-source library for numerical computation and large-scale machine learning that ease Google Brain TensorFlow, the process of acquiring data, training models, serving predictions, and refining future results.

- Tensorflow bundles together Machine Learning and Deep Learning models and algorithms.
- It uses Python as a convenient front-end and runs it efficiently in optimized C++.
- Tensorflow allows developers to create a graph of computations to perform.
- Each node in the graph represents a mathematical operation and each connection represents data. Hence, instead of dealing with low-details like figuring out proper ways to hitch the output of one function to the input of another, the developer can focus on the overall logic of the application.

The deep learning artificial intelligence research team at Google, Google Brain, in the year 2015 developed TensorFlow for Google's internal use. This Open-Source Software library is used by the research team to perform several important tasks.

TensorFlow is at present the most popular software library. There are several real-world applications of deep learning that makes TensorFlow popular. Being an Open-Source library for deep learning and machine learning, TensorFlow finds a role to play in text-based applications, image recognition, voice search, and many more. DeepFace, Facebook's image recognition system, uses TensorFlow for image recognition. It is used by Apple's Siri for voice recognition. Every Google app that you use has made good use of TensorFlow to make your experience better.

4.3.1.4 Tensorflow object detection API

The TensorFlow Object Detection API is an open-source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models.

- There are already pre-trained models in their framework which are referred to as Model Zoo.
- It includes a collection of pre-trained models trained on various datasets such as the
- COCO (Common Objects in Context) dataset,
- the KITTI dataset,
- And the Open Images Dataset.

As you may see below there are various models available so what is different in these models. These various models have different architecture and thus provide different accuracies but there is a trade-off between speed of execution and the accuracy in placing bounding boxes.

Tensorflow bundles together Machine Learning and Deep Learning models and algorithms. It uses Python as a convenient front-end and runs it efficiently in optimized C++.

Tensorflow allows developers to create a graph of computations to perform. Each node in the graph represents a mathematical operation and each connection represents data. Hence, instead of dealing with low-details like figuring out proper ways to hitch the output of one function to the input of another, the developer can focus on the overall logic of the application.

The deep learning artificial intelligence research team at Google, Google Brain, in the year 2015 developed TensorFlow for Google's internal use. This Open-Source Software library is used by the research team to perform several important tasks.

TensorFlow is at present the most popular software library. There are several real-world applications of deep learning that makes TensorFlow popular. Being an Open-Source library for deep learning and machine learning, TensorFlow finds a role to play in text-based applications, image recognition, voice search, and many more. DeepFace, Facebook's image recognition system, uses TensorFlow for image recognition. It is used by Apple's Siri for voice recognition. Every Google app that you use has made good use of TensorFlow to make your experience better.[8]

4.4 *Configuration Code*

Sample configuration code is presented in

Listing 4.1: Wi-Fi Connection with raspberry

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=BD
network={
     ssid="wifi username"
     psk="wifi password"
}
```

Sample configuration code is presented in

Listing 4.2: Ultrasonic Sensor

```
def measure():
        GPIO.output(TRIG, False)
        time.sleep(0.01)
        GPIO.output(TRIG, True)
        time.sleep(0.00001)
        GPIO.output(TRIG, False)

        while GPIO.input(ECHO)==0:
                pulse_start = time.time()

        while GPIO.input(ECHO)==1:
                pulse_end = time.time()

        pulse_duration = pulse_end - pulse_start
        distance = pulse_duration * 17150
        distance = round(distance, 2)

        print("Distance:",distance,"cm")
        return distance
```

Listing 4.3: Dc Motor Declaration

```
def measure():
    import RPi.GPIO as GPIO
    import time
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    m11 = 6
    m12 = 13
    m21 = 26
    m22 = 19
    GPIO.setup(m11, GPIO.OUT)
    GPIO.setup(m12, GPIO.OUT)
    GPIO.setup(m21, GPIO.OUT)
    GPIO.setup(m22, GPIO.OUT)
```

Dc motor driver pin to raspberry pi pin m11 mean motor driver IN1 pin and 6 mean raspberry pi GPIO 6, m12 = motor driver IN2 and 13 mean GPIO 13, m21= motor driver pin IN3 and 26 mean GPIO 26, m22 = motor driver IN4 and 19 mean GPIO19.

Listing 4.4: Dc Motor Stop, Forward, Backward, Left and Right

```
def def Pause():
    GPIO.output(m11, 0)
    GPIO.output(m12, 0)
    GPIO.output(m21, 0)
    GPIO.output(m22, 0)
def forward():
    GPIO.output(m11, 0)
    GPIO.output(m12, 1)
    GPIO.output(m21, 0)
    GPIO.output(m22, 1)
def back():
    GPIO.output(m11, 1)
    GPIO.output(m12, 0)
    GPIO.output(m21, 1)
    GPIO.output(m22, 0)
def left():
    GPIO.output(m11, 1)
    GPIO.output(m12, 0)
    GPIO.output(m21, 0)
    GPIO.output(m22, 1)
def right():
    GPIO.output(m11, 0)
    GPIO.output(m12, 1)
    GPIO.output(m21, 1)
    GPIO.output(m22, 0)
    GPIO.output(lr, 1)
```

Listing 4.5: Object position detection

```
if (xmax-xmin)>0.40 and (ymax-ymin)>0.40 :
        print("Stop car")
        stop()
        time.sleep(1)
        print("This is the decision what i do?")
        if (xmin<0.3<xmax) and (ymin<0.3<ymax) :
            print("Right turn Car")
            right()
            time.sleep(0.9)
            stop()
            time.sleep(0.7)
```

4.5 *Chapter Summary*

In The project has verified the above pictures taken show the working of the model where it is able to detect the special pattern provided and also is able to detect the obstacles in the surrounding.

The model is thereby able to do all required task which as we stated above.

5 Conclusion

In this paper, a method to make a self-driving robot car is presented. The different hardware components and their Assembly are clearly described. This method to determine the uneven, marked or unmarked road is explained in details relying upon Open CV. Using ultrasonic sensors, the collisions with obstacles is avoided. The algorithm mentioned in the paper has been successfully implemented on a small autonomous car.

5.1 *Limitation*

- The low quality build of the car wheel
- The pi camera was not good and the signal was choppy.
- The rechargeable batteries life was short.
- We had to use a power bank and it was heavy.
- We needed to slow the motor in order to get a better result out of training.

5.2 *Future Work*

The work could be enhanced by improving the algorithm and high resolution Camera by adding advanced machine learning to it. Using advanced algorithm we can improve Image processing algorithms. Multi layered processors can be used for fast processing. The present obstacle detection algorithm just detects the obstacle and stops, but in future it can be improved by the avoiding the obstacle, and go through another way using advanced obstacle detection algorithm.

References

[1] "GitHub - EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi: A tutorial showing how to set up TensorFlow's Object Detection API on the Raspberry Pi", *GitHub*, 2021. [Online]. Available: https://github.com/EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi. [Accessed: 06- Sep- 2021].

[2] 2021. [Online]. Available: https://www.digikey.com/en/maker/projects/how-to-perform-object-detection-with-tensorflow-lite-on-raspberry-pi/b929e1519c7c43d5b2c6f89984883588. [Accessed: 06- Sep- 2021].

[3] "Running TensorFlow Lite Object Recognition on the Raspberry Pi 4", *Adafruit Learning System*, 2021. [Online]. Available: https://learn.adafruit.com/running-tensorflow-lite-on-the-raspberry-pi-4?view=all. [Accessed: 06- Sep- 2021].

[4] "Raspberry Pi 4 GPIO Pinout, Specs, Schematic (Detailed board layout)", *eTechnophiles*, 2021. [Online]. Available: https://www.etechnophiles.com/raspberry-pi-4-gpio-pinout-specifications-and-schematic/#raspberry-pi-4-schematicofficial. [Accessed: 06- Sep- 2021].

[5] *Technoarete.org*,2021.[Online].Available:https://www.technoarete.org/common_abstract/pdf/IJERECE/v5/i4/Ext_56874.pdf. [Accessed: 06- Sep- 2021].

[6] *Ijert.org*, 2021. [Online]. Available: https://www.ijert.org/research/Autonomous-Self-Driving-Car-using-Raspberry-Pi-Model-IJERTCONV7IS08081.pdf. [Accessed: 06- Sep- 2021].

[7] R. Sirolkar, R. Datar, G. Behere and A. Dhongade, "Self-Driving Autonomous Car using Raspberry Pi", *Ijert.org*, 2021. [Online]. Available: https://www.ijert.org/self-driving-autonomous-car-using-raspberry-pi. [Accessed: 18- Sep- 2021].

[8] W. Dimbour and S. Manou-Abi, "Asymptotically $$\omega $$ ω -Periodic Functions in the Stepanov Sense and Its Application for an Advanced Differential Equation with Piecewise Constant Argument in a Banach Space", *Mediterranean Journal of Mathematics*, vol. 15, no. 1, 2018. Available: 10.1007/s00009-018-1071-6.

[9] H. Mujtaba, "Real-Time Object Detection Using TensorFlow - Great Learning", *GreatLearning Blog: Free Resources what Matters to shape your Career!*, 2021. [Online]. Available: https://www.mygreatlearning.com/blog/object-detection-using-tensorflow/#:~:text=Object%20Detection%20using%20Tensorflow%20is,an%20image%20or%20a%20video. [Accessed: 19- Sep- 2021].