# Autonomous Car with Collision Avoidance


*Group Members*

**BASIT NADEEM (150180)**
**M. ABDULLAH SUHAG (150270)**
**WALEED AHMAD (150378)**


**BACHELOR OF ELECTRICAL ENGINEERING**

**(2015-2019)**


**Project Supervisor**

**MS. MADEEHA UZMA**

**(LECTURER)**

**DEPARTMENT OF ELECTRICAL ENGINEERING**

**FACULTY OF ENGINEERING**

**AIR UNIVERSITY, ISLAMABAD**

# Autonomous Car with Collision Avoidance

**Final Year Project**

**(2015-2019)**

**DEPARTMENT OF ELECTRICAL ENGINEERING**

# Autonomous Car with Collision Avoidance

*Submitted by:*


| | |
|---|---|
| **BASIT NADEEM** | **(150180)** |
| **M. ABDULLAH SUHAG** | **(150270)** |
| **WALEED AHMAD** | **(150378)** |


**Project Supervisor**

___________________________________

**Ms. Madeeha Uzma**

**Lecturer**

**Head of Department**

___________________________________

**Dr. Shahryar Saleem**

**Assistant Professor**

# Acknowledgement

Firstly, we would like to thank ALLAH Almighty, who has blessed us with countless blessings, to defeat all challenges that came our way amid the finishing of our project. While completing our work, we were guided and at times practically helped by our worthy instructors. We acknowledge their efforts and show our gratefulness to them. We are highly thankful to our parents for their prayers, continuous encouragement, and financial support. We thank our peers who encouraged us to steer through difficult but temporary setbacks; during the project work. We are appreciative to our supervisor Ms. Madeeha Uzma who bailed us out in the finishing of our task. We are truly obliged and value his noteworthy by and large direction in our task. We are additionally earnestly appreciative to the department lab engineers for their direction and backing.

# Abstract

Autonomous means of transport are the near-future brilliant autos expect to be driver less, proficient keeping away from imperfection. To reach to this outcome, automaker have begun working to understand the potential and reveal the outcomes of achieving this result. It also reveals the difficulties and their solutions. Unmanned ground vehicles are capable of driving itself without on-board operators. A UGV, with sensors installed for interacting with environment are now-a-days an interesting area of research. This UGV will be able to sense its environment and take decision according to it. Our proposed design was to implement a prototype of driverless car in controlled environment. The project poses particular challenges in terms of controllability and design considerations.

Results are obtained using image processing. The document provides an insight on final year project, autonomous car. The report discusses its objective, its entire working and entire system of autonomous car. It briefs the mechanical design, electronics and electrical design.

# Nomenclature

$ASIRF$ Association for Safe International Road travel

$DARPA$ Defense Advanced Research Projects Agency

$GPIO$ General Purpose Input-Output

$HSV$ hue, saturation, value

$OpenCV$ Open Source Computer Vision

$PWM$ Pulse Width Modulation

$RPi$ Raspberry Pi

$UAV$ Unmanned Air Vehicle

$UGV$ Unmanned Ground Vehicle

$US$ Ultrasonic Sensor

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1
# Introduction

What is autonomous car? Autonomous car are the visions of unmanned and autonomous machines. A car which is capable to drive its way to it distention without a driver through uncontrolled environments.

All things considered, ongoing years have seen impressive improvement towards progress of autonomous and unmanned vehicles Car makers, for example, Ford, BMW, Tesla, and different organizations, for example, NVIDIA are putting billions of dollars in self-ruling vehicle driving exploration and make an autonomous car can capable of making its way through uncontrolled environments. Such vehicles are a connected utilization of progressively complex man-made brainpower and mechanical technology abilities. [1]

## 1.1 Unmanned Ground Vehicle (UGV)

A UGV is a vehicle that works on ground and without a locally open human manager. By and large, the UGV have various sensors prepared on it for watching nature, and after that taking activities or settle on choices as indicated by it independently and pass the information to a human director at a substitute region. By design of UGV, we are conserned with number of omponents: stage, sensors, control frameworks, direction interface, correspondence connections, and frameworks coordination highlights.

## 1.2 Motivation of Project

Almost **1.25 million** individuals pass on in street crashes every year, by and large 3,287 passing per day. An extra 20-50 million are harmed or debilitated and the greater part of all street traffic passing happen among youthful grown-ups ages 15-44. More than 85 percent of these accidents are caused by driver negligence. Report by Association

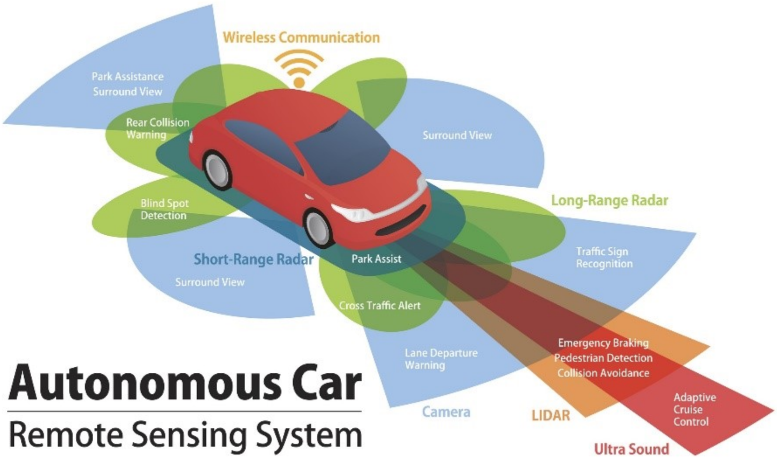


Figure 1.1: Autonomous car environment

for Safe International Road travel (ASIRF).
This motivate us try to manufacture an autonomous car which is capable to sense variable environments prevailing in our world. Try to make it decisions which are safe for human life and make there ride more comfort by saving their precious time. [2]

## 1.3 Evolution of Autonomous Car

Self-governing mechanical technology have experienced a great deal of development over the most recent couple of decades. One of the innovations that has be most talked and contributed on is self-driving Autonomous Car. The exploration in independent field began in mid-60s and first independent and really self-governing vehicles showed up during the 1980s with Carnegie Mellon University. However, the genuine kick begin to the improvement of driverless self-governing vehicle was given by Defense Advanced Research Projects Agency (DARPA). [3]
DARPA, aim to check your autonomous robot and test its safety protocols on extreme conditions and make shore that the car is safe for human life that human can relay it life on machine.
Firstly, there autonomous robots are used for mining process so that it saves lives of labors but as technology advanced their purpose form mining shifted toward autonomous


Figure 1.2: Evolution of UGV

car and machines so, that its saves time and lives of human. [4]

## 1.4 Problem Associated

The issues with independent frameworks in reality are various. A robot must have some approach to see its condition. A robot must have some approach to balanced out its very own position, yet should be set up to keep up this state regardless of whether any given sensor may not work. A robot must have some technique for changing its position, yet ought to in all likelihood observe unequivocally how its position is truly moving. A completely independent robot must most likely settle on choices to change its position dependent on the world it sees; it ought to be noted, in any case, that this task does exclude CAR PARKING out of extension. Another issue of this generally new field is the nonattendance of data sharing among vehicles and controller. Independent vehicles are perplexing frameworks that depend intensely on innovations, for example, OPEN CV, GPS, high-definition maps, and artificial insight for route and impact shirking. This implies each self-governing vehicle is continually gathering and examining an extremely high volume of information, which as indicated by Intel is about 2.6 terabytes every hour. Which is a great deal for every hour we need to deal with that information which it a ton in itself. [5]

## 1.5 Objectives

With the potential for human slip-up ousted, self-driving vehicles will diminish events of accidents realized by driver screw up, crushed driving or occupied drivers. When driverless vehicles become typical on our boulevards, it is normal that mishaps are probably going to fall by an astounding 90% It spares our time make our ride a lot simpler. Self-governing vehicles are, to all plans and purposes, programming on wheels. The innovation associated with a driverless vehicle of things to come will be to such an extent that every vehicle can be advanced to guarantee fuel utilization is as productive as could reasonably be expected. [6]To such an extent that new-age vehicles are relied upon to help diminish discharges by 60%.

## 1.6 Applications

As the field of mechanical technology progresses and the establishment of learning manufactures, extends that are on the front line of the field will turn out to be progressively further developed. This can be straightforwardly found in the multifaceted nature and scale that was found in the DARPA. The Challenge began with an objective of structure a framework that could self-sufficiently explore on open landscape, to expecting a high achievement rate in the development of a self-sufficient route framework that could drive through an urban setting while at the same time obeying transit regulations, maintaining a strategic distance from people on foot and other true obstructions. This progress happened quickly once a strong assemblage of information had been produced. The application of autonomous car is numerous. Some of these applications are as follow. [7]

- Save time and safe task achievements
- Reduce life risk
- Military use
- Industrial task

- Move at rough terrains
- Avoid hurdles and obstacle
- Work without on-board human presence and interference

# Chapter 2
# Literature review

As the field of apply autonomy progresses and the establishment of learning constructs, extends that are on the cutting edge of the field will turn out to be progressively further developed. This can be legitimately found in the multifaceted nature and scale that was found in the DARPA Urban Challenge when contrasted with the underlying DARPA Grand Challenge. [8] The Challenge began with an objective of structure a framework that could independently drive with no human driver. A self-sufficient route framework that could drive through a urban setting while at the same time obeying transit regulations, dodging people on foot and other certifiable hindrances. This change happened quickly once a strong group of information had been produced.
The implementation of UGV (autonomous car) requires information about the sensor and electro-mechanical components used in the project. For this reason, research was conducted on the basis of the previous study-oriented work that was done on the autonomous vehicles. [9] Therefore, the full driving assignment is too perplexing an action to be completely formalized as a detecting acting mechanical autonomy framework that can be expressly explained through model-based and learning-based methodologies so as to accomplish full unconstrained vehicle self-governance. In our university this project is done by department of Mechatronics (session 2009) which was based on Arduino which include obstacle avoidance and land mines.
The previously work on the autonomous cars is based on the Arduino micro-controller which was not so compatible. Therefore, raspberry Pi controller with electro-mechanical system in which servo motor and DC motors are included. Therefore, the main features of the project are:

- Programming to be done on raspberry pi micro-controller.
- DC and servo motors are used.

- Light weight and portable design.
- Pi-camera to be used for image processing.

# Chapter 3
# Design Procedure

## 3.1 Overview

The design procedure of the unmanned ground vehicle required rigorous knowledge on all aspects of road algorithms and scenarios. Mainly the course of action was to try to make the car work autonomously on road and interact with environment. This would be done by 3 modules.

- Input module
- Output module
- Control module

The input module consists of the actions the user wants the car to perform. Now since the car is working autonomously, the instructions given by the user to it are directly through the computer/processor used for it. But the instructions are sent via a WiFi channel accessing the processor remotely. Therefore, the communication was done by a remote server which the user holds and operates through local network.

The output module consists of the output which the car has to produce. Here the output of the car is basically in the form of the movement of the motors after deciding from the sensor values and processing it. Therefore, the output module mainly consists of the motors along with their drives.

Lastly, the most important module of the car is the control module which will be placed at its supporting base. The control module is like the brain of the project. All the actions which the user wants the car to perform are firstly passed form the input module to the control module. Here the information is processed and then passed on to the output module. The control module consists of all the electronic circuitry along with the micro-controller and sensor units. All the circuits would be driven by a battery.

All these modules then have to be placed on the car's body. The body consists of a base with wheels and moving mechanism.
The design process of the project was divided into 2 major phases.

- Mechanical modeling
- Electrical modeling

## 3.2 Mechanical modeling

The first task towards the accomplishment of the car was to devise a suitable mechanical model which would consist of the main body on which the motor and control unit are mounted with proper moving mechanism using motors. The central base would hold the electronic circuit, modules and battery.
The challenging task was to select the type of material that would be most suitable in terms of the following aspects:

- Lightweight
- Durability
- Strength
- Machine ability
- Cost and availability

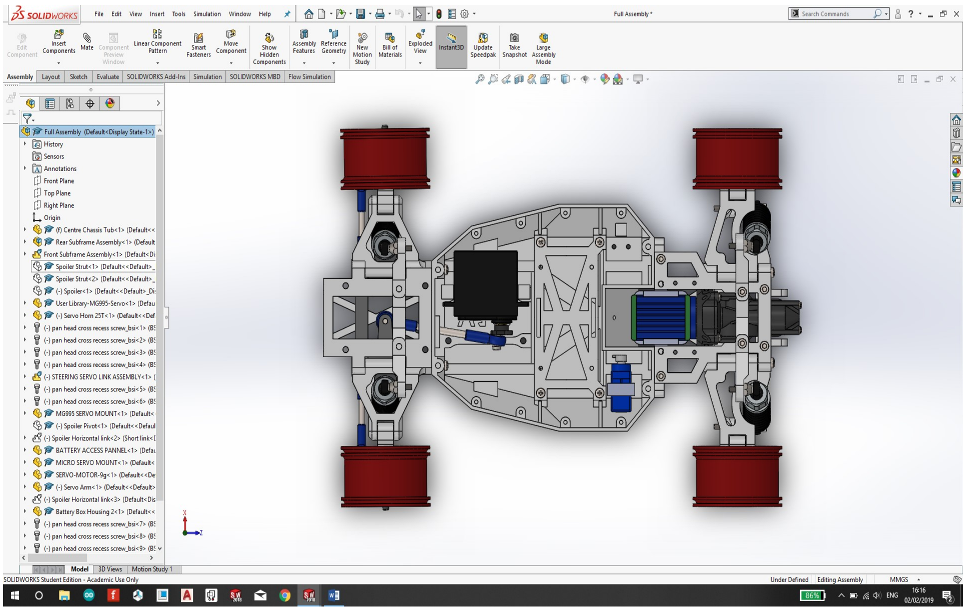

Figure 3.1: Car chassis

This chassis of the car consists of:

- Servo motor for steering mechanism.
- 9V battery for mechanism of moving.

## 3.3 Electrical Modeling

After the completion of mechanical modeling, the next step was to research on all the required electronics for the project. The electrical modeling is basically the interacting unit of input, output and control modules as described in article 3.1

The electrical model was decided and divided into following components:

- Processor
- Motors
- Motor driver
- Battery

### *3.3.1 Processor*

A very high-speed microcontroller was required for the programming of the motor control and image processing and making decision according to the environment. The processor used in our project is Raspberry Pi 3 Model B+ which is considered as the brain of the project. It is the control unit by which input unit interacts and the decision making is done in it and output unit is given commands to perform as per decision produced by it. The Raspberry Pi is a progression of little single-board PC. The model we used is Model B+, with processor Broadcom BCM2837B0, with ethernet, USB 2.0 ports and Wi-Fi module installed in it.

Figure 3.2: Raspberry Pi

### *3.3.2 Motors*

Motors are the part of the output module of the car as mentioned in article 3.1. the design required is to have small sized motors that can produce enough RPM to bear the weight of car. The most appropriate option in the light of above requirement was to use DC motor for movement of chassis and servo motor for steering movement.

### *3.3.3 DC motor*

A DC motor is rotating electrical motor which changes over direct stream electrical energy into mechanical energy. The most generally perceived sorts rely upon the forces conveyed by alluring fields. Practically a wide extent of DC engine have some inside fragment, electromechanical, to unpredictably change the course of current stream in part of the motor.

We have used **Brushed DC motor**. It produces torque direct from DC control gave to the motor by using internal reward, stationary magnets and turning electromagnets. The brushes reach the commutator. At the point when a DC voltage is connected to the brushes, that voltage is moved into the commutator which, thusly, powers the winding. This electrical info creates an attractive field with armature.

The left-out armature is driven from the left stator magnet, towards the magnet on the right. Moreover, the right half is driven a long way from the right magnet, towards the left.

By constantly alternating the polarity of the magnetic field around the armature, the motor shaft is made to continuously rotate.

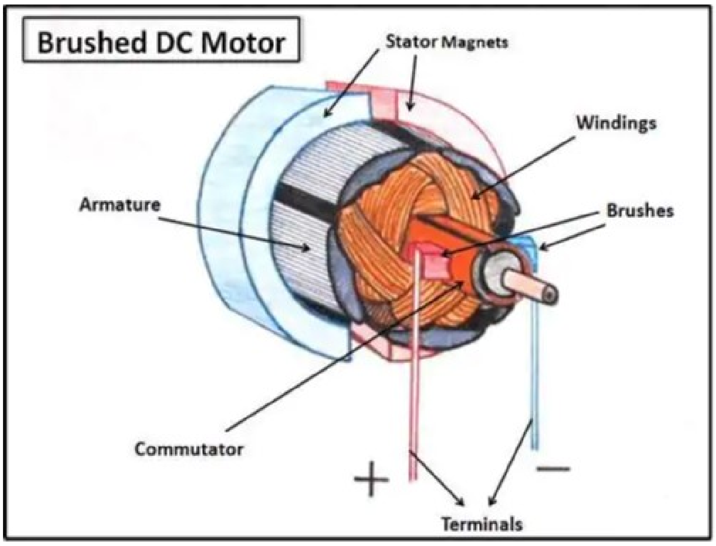


Figure 3.3: DC motor

### *3.3.4 Construction of DC motor*

Exactly when the circle is powered, an attractive field is made. The Armature continues turning. Exactly when the armature ends up being on a dimension plane balanced, the torque ends up zero. Now, the commutator turns around the heading of current through the curl, switching the attractive field. The procedure at that point rehashes, causing ceaseless turn.

### *3.3.5 Rating*

- Name: Gear Motor
- Voltage: 9V
- Output speed: 150 rpm
- No-load current: 200mA
- Rated speed: 100 rpm
- Rated torque: 3 Kg-cm

### *3.3.6 Servo motor*

To produce steering movement, an additional motor was attached at the front tires to rotate the car. Servo motor was used because of its compact size and less weight. Servo motors are mostly 3 wire motors. Two wires used for sending power to the motor and the remaining one wire is used to control the rotation of the motor directed by a control circuit.

The servo used in the chassis is the servo manufactured by **Kyosho Perfex KS-12MG** micro servo motor. The advantage of this motor is light-weight.

To completely see how the servo functions, you have to investigate the hood. Inside, it comprises of a little DC engine, potentiometer, and a control circuit. The motor is associated by riggings to the control wheel. As the engine turns, the potentiometer's


Figure 3.4: Servo motor

obstruction changes, so the control circuit can decisively supervise how much improvement there is and in which course.

Servos works on electrical pulse, or pulse width adjustment (PWM). There is a base pulse, a most extreme pulse, and impartial pulse. A servo motor can turn 90° in either direction. The PWM sent to the motor chooses position of the motor turns, which makes the rotor will swing to its position. The motor sees a pulse every 20 ms.

For example, a 1.5ms will make the rotor to swing to the 90° position. Now, less than 1.5ms will take it to the CCW direction toward the 0° position, and greater than 1.5ms will turn the rotor to CW direction toward the 180°.

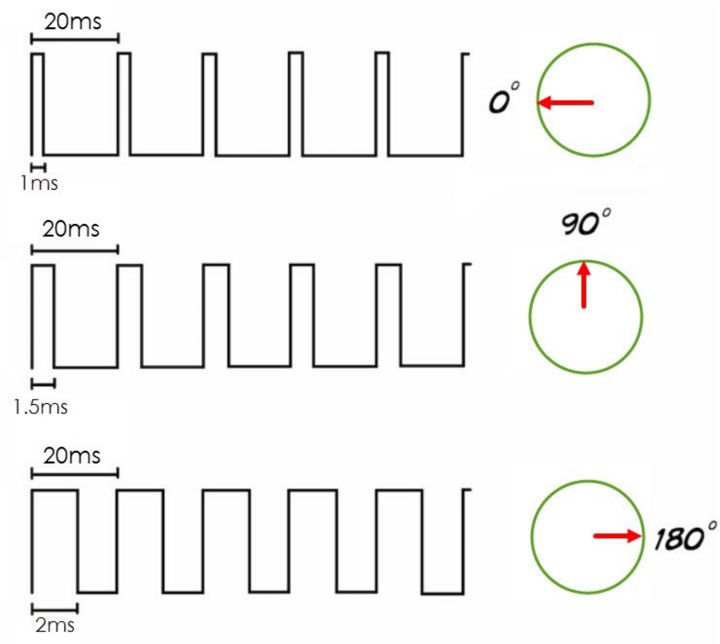


Figure 3.5: Servo Duty cycle

### *3.3.7 Rating*

Servo Technology: Analogue

Moment of force: 2.60 kg/cm

Servo speed: 0.09 s

Weight: 15 g

Torque: 3.5 Kg.cm

Gear Type: Metal

**Servo Duty cycle:**

Dc = 0.5/20×100 = 2.5% = 0°

Dc = 1.5/20×100 = 7.5% = 90°

Dc = 2.5/20×100 = 12.5% = 180°

### *3.3.8 Motor driver*

The motor driver is utilized for the DC motor. Motor driver utilized in our undertaking is L298N. L298 IC is most normally used to drive motors dynamically or motor controllers. it can be controlled by Arduino, RPi etc. They get input signals from mini-scale controllers and run the motors connected to their yield sticks. L-298 motor driver can control two diverse DC motors in the meantime. Also, it can control a solitary stepper motor too. L298 has two (PWM) pins. When we want to run in PWM mode, these PWM pins are utilized. What's more, the motor can likewise be kept running at full speed by setting jumpers on the PWM pins. We can pivot the motor in either clockwise or counter clockwise course by changing the flag's extremity at its information. It has various genuine activities for example mechanical technology, entryways lock frameworks and so on.


Figure 3.6: L298 Motor Driver

L298 is a high present and high voltage IC. It gains TTL power flags and works with various burdens like motors, solenoid, transfers and so forth. It has two explicit pins for empowering or impairing the specific gadget joined at its yield.

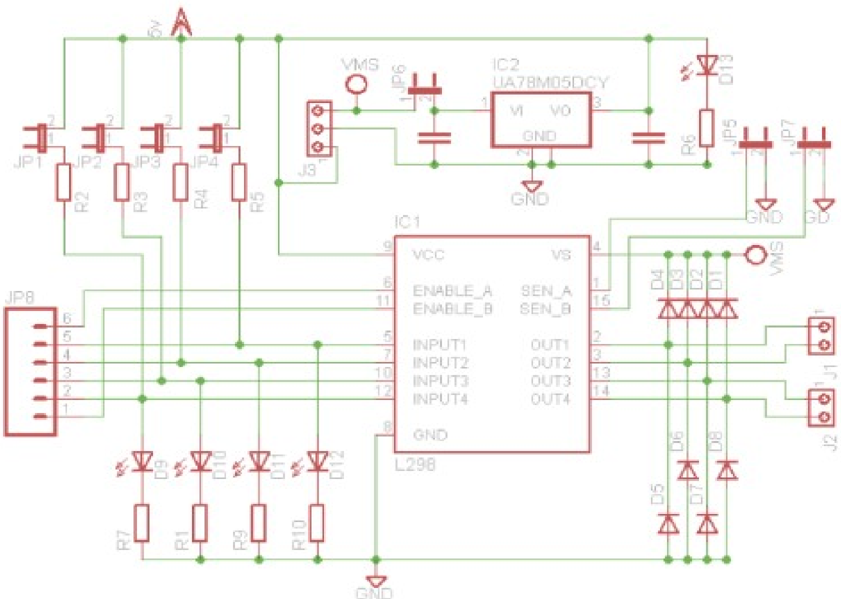


Figure 3.7: Schematics of Motor Driver

### *3.3.9 Battery*

**Lithium Polymer** batteries ("LiPo" batteries), are a present day age sort of battery currently utilized in numerous gadgets. They have been getting in noticeable quality in the radio control industry all through the latest couple of years, and are by and by the most standard choice for anyone scanning for long run events and high power.
LiPo batteries offer a wide group of focal points, anyway every customer must pick if the favorable circumstances surpass the detriments. For a consistently expanding number of people, they do. Through my eyes, there is nothing to fear from LiPo batteries, in light of the fact that you seek after the standards and treat the batteries with the respect they merit.
The advantages of LiPo battery are following:

- Lighter in weight and can be given any size or shape in manufacture.
- High power battery.
- High capacity and working hours.


Figure 3.8: LiPo Battery

### *3.3.10 Ultrasonic Sensor*

Ultrasonic sensing is an amongst the most ideal approaches to detect vicinity and distinguish levels with high unwavering quality. This sensor is an apparatus that checks the partition to an article by US waves. A ultrasonic sensor uses a transmitter to send and get ultrasonic pulses to measure closeness. High-rehash sound waves reflect from limits to pass on verifiable reverberation structures.


Figure 3.9: Ultrasonic Sensor

Our US sensor, similarly as different others, use a single transducer to send a heartbeat from trigger stick and to get at the reverberation stick. The sensor chooses the partition to a goal by assessing time btw the producing and getting of the ultrasonic heartbeat from trigger to resound by reflecting of wave.

US sensor can be used in-house or out of the it because if its rigidness. Ultrasonic sensors can oversee influence keeping away from for a robot, and being moved routinely, as long as it isn't extravagantly quick.

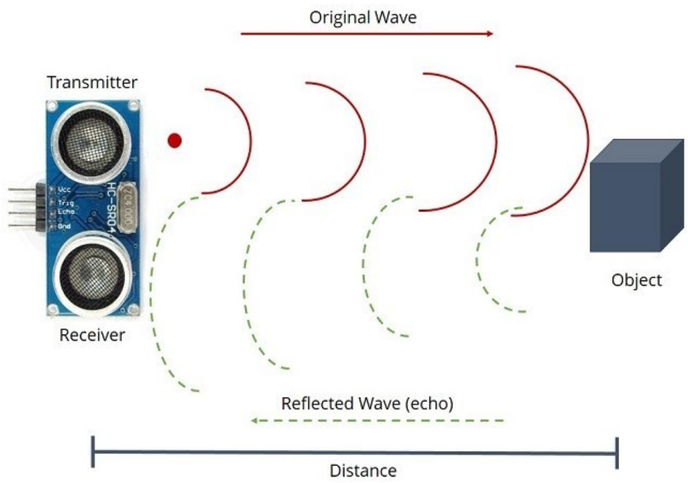


Figure 3.10: Sensor working

**For Distance calculations:**

L = ½ x T x Constant (34300 cm/s)

Constant is used as speed of sound for the ultrasonic wave to travel in the air.

**Interfacing Ultrasonic with Raspberry Pi:**

- Raspberry Pi Input-Output Pins (GPIO) can have voltage up to 3.3Volts.
- So, resistors are installed in the circuit to get 3.3 Volts.

$$3.3/5=\mathrm{R}_2/(R_2+1k)$$

$$0.66(1\mathrm{k}+\mathrm{R}_2=R_2)$$

$$0.66+0.66\mathrm{R}_2=R_2$$

$$0.66=0.34\mathrm{R}_2$$

$\mathrm{R}_2=1.94$

- R1 is fixed for 1k ohms and R2 is calculated by voltage divider to 2k ohm.

### *3.3.11 Ultrasonic Algorithm*

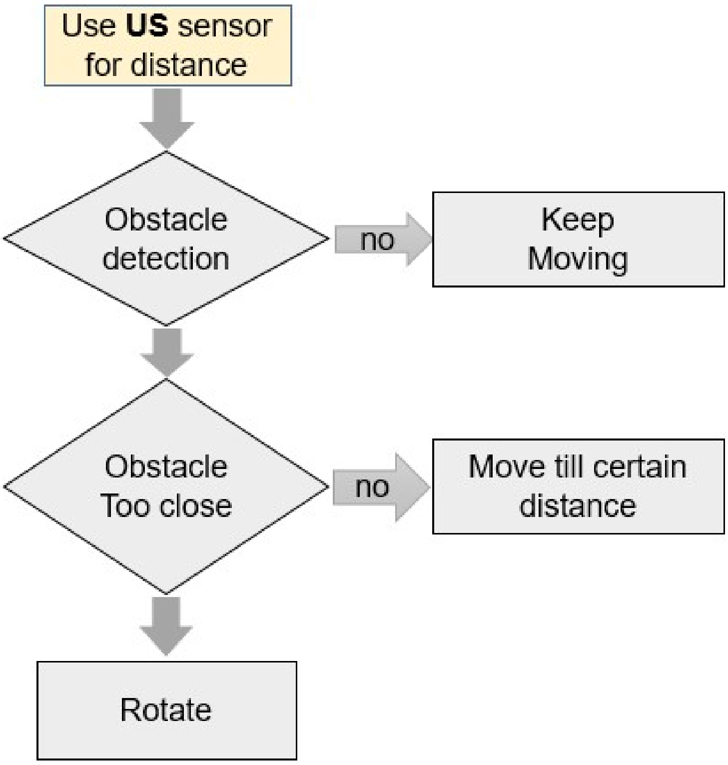


Figure 3.11: US Algorithm

If(cm>50)

“move forward”

If(cm>45cm<50)

“stop”

If(cm<45)

“turn the car 45◦/90◦”

### *3.3.12 Pi-Camera*

The Pi-Camera is a module usually used with the Pi controller. This board is used to feed in HD video stream into the system or pictures.
We have used Pi Camera for image processing input in our project. All the algorithms are run according to the video provided by camera to processor. And decisions are made according to it.
Features:

- On-board lens for imaging.
- 8-MP camera with resolution 3280 x 2464-pixel
- Supports video formats and resolutions
- Weight just over 3g
- Associates with the Raspberry Pi board effectively.
- Camera v2 is supported by Raspberry OS.

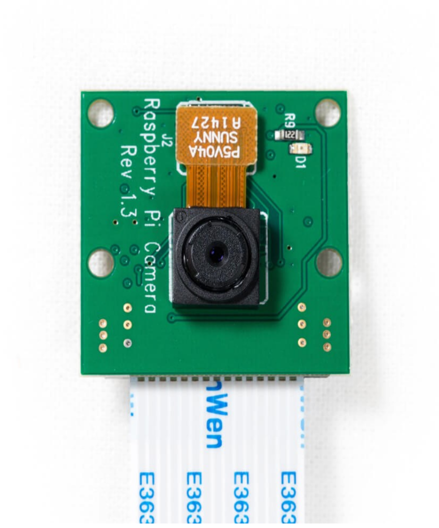


Figure 3.12: Pi Camera

# Chapter 4
# Image Processing Algorithms

Different Algorithms are used for image processing technique. These algorithms are used as an input from the camera real view and goes on to the control unit, Raspberry pi in this scenario, for controlling purposes and the calculated output is forwarded to the output unit. This calculated output decides the operation performed by all the motors and driver. All these algorithms are run in OpenCV.
This output is calculated through the following algorithms:

- **Haar-Cascade Classifier** for stop sign detection.
- **Hough transform** for lane detection.
- **HSV image conversion** for traffic light detection.

## 4.1 OpenCV

OpenCV (Open Source Computer Vision Library) is an AI programming library. OpenCV was endeavored to give a normal structure to PC vision applications and to breath life into the utilization of machine recognition in the business things. As an open source stage, OpenCV is make basic associations to use and change the coding. Initially, it was developed by Intel, later it got supported with others. OpenCV was originally in C/C++ but now it contains bindings of different languages i.e. python, JAVA etc.
**Python** is a valuable comprehensively programming language started by Guido van Rossum, which ended up being outstanding in a nutshell time generally in perspective on its code clarity. Its enables the designer to communicates the contemplations in lesser lines. However, another huge segment of the Python is that is it might be viably extended with C and C++. This gives us two inclinations: first, our code is as brisk as exceptional C/C++ code and second, it is incredibly easy to code in Python. This is the methods by which OpenCV-Python work.

The library of **Numpy** makes the assignment progressively easier. Numpy is an exceptionally upgraded brary for numerical undertakings. it adds array values to the OpenCV objects. Other than that, couple of various libraries like can also be associated with Numpy library to work along with it.

The library has in excess of 2500 person-trained estimations, which fuses a total game plan of both extraordinary and cutting edge PC vision and AI figuring's. These estimations can be used to identify and see faces,understand objects, bunch human exercises in accounts, camera tracking improvements, extract 3Ds shapes of articles, solidify pictures to make a high destinations picture of a whole scene, and so on. OpenCV has in excess of 47,000 client organize and assessed number of downloads outflanking 18 million.

Figure 4.1: OpenCV

### *4.1.1 Applications*

Applications are:

- 2D and 3D highlight Toolkit.
- Sense of self movement estimation.
- Facial acknowledgment System.
- Signal Recognition.

## 4.2 Haar-Cascade Classifier

A Haar-Cascade is in a general sense a classifier which is used to distinguish the thing for which it has been prepared for, from the source. The Haar- Cascade is set up by superimposing the positive picture over a great deal of negative pictures. The arrangement is usually done on a server and on various stages. Better results are procured by using superb pictures and extending the proportion of stages for which the classifier is readied.
Article Detection by Haar incorporate based course classifiers is a persuading object acknowledgment methodology. It is an AI based methodology,working on a number of positive and negative pictures. Now we will work with the **Stop Sign detection**. At first, computation will require incredible arrangement of the Positive pictures and negative pictures to set up the classifier. By then we need to expel features from it. For this, Haar features showed up in the picture are used. Every piece is a single regard procured by withdrawing aggregate of pixel under white square shape from entire of pixels under Dark square shape.

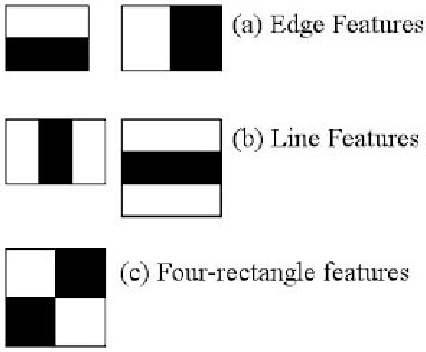


Figure 4.2: Haar-cascade working

The potential size and area of each section is used to describe the functional packages. To calculate each component you need to find all the pixels under the white and dim square images. The main picture is shown to solve this. In any case, the importance of your look is only 4 pixels in the proposed pixel calculations. For that, we apply each section to all the pictures. For each segment, it will take into account that many will be able to view a number of positive and negative images. Surely bulls will be. We emphasize that at least with the botch ratio, they suggest that they highlight the most demanding face and face images.

This is what if a window succeeds in the main stage. We do not think about the remaining moments. Apply for a second time by accidentally consciously, and continue with the strategy. The window that passes all stages is a hint.

For **Stop-sign detection**, first we need to load an XML classifier file of stop-sign classifier. XML is an extension of classifier file. Then image is converted to gray scale. Now we need to find the stop signs in an image. If it is found, it return the position of sign. Once we get the locations of sign, now we need to create ROI (Region of Interest) for signs. Cv2.rectangle function is found for making rectangle on region of interest. These ROI are the detected stop signs.

Results are as under:


Figure 4.3: Stop sign detection

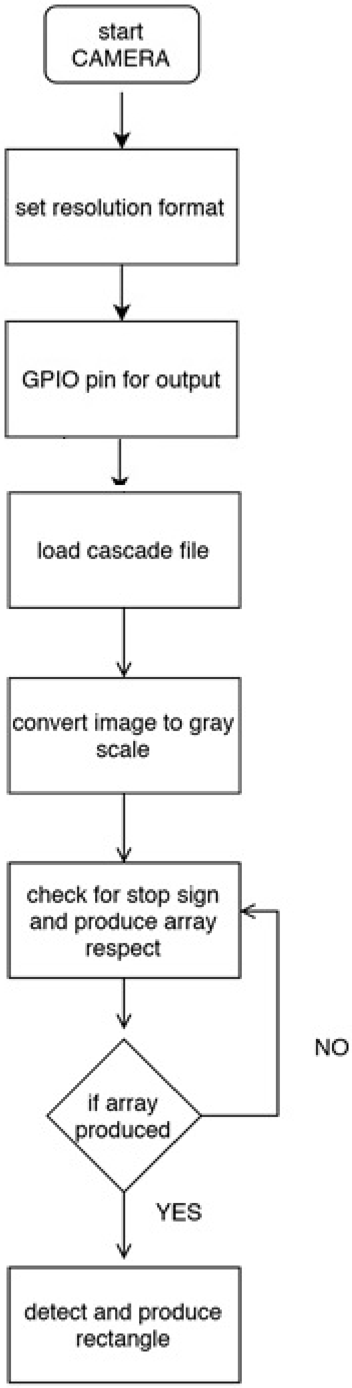


Figure 4.4: Haar-cascade Flow Chart

## 4.3 Hough Transform

The **Hough transform** is an **object extract mechanism** which is used in image processing, analysis and computer vision. Motivation behind this strategy is to discover defective cases of the articles inside a specific area of shapes by a casting a ballot system. This casting a ballot plan is done in a parameter space, object applicants are gotten as the nearest maxim in a collector field, completely computed by the calculation of the Hough Transform.

In robotized examination of advanced pictures, a subproblem routinely rises of recognizing essential shapes. Generally speaking, an **edge detector** can be used as a pre-made step to obtain picture centers or image pixels on a perfect curve in the center of the image. On account of flaws in either the image data or the edge identifier, regardless, there might miss spotlights or pixels on the perfect curves. Thusly, it is much of the time non-inconsequential to hoard the isolated edge features to a fitting plan of shapes. The inspiration driving the Hough change by making possible to group edges into thing hopefuls by playing out an unequivocal throwing a poll procedure over a ton of parameterized picture objects.

At first, hough trasnform was used for straight lines only which were given by y=mx+c equation, where (c, m) is the point. Notwithstanding, vertical lines represent an issue. With more study and working on it, people started using another equation with this algorithm for vertical lines.

$$\mathbf{r = x.\cos\theta + y.\sin\theta}$$

here, r is the detachment from the commencement on the straight line, and θ (theta) is the angle between the origin and straight line.

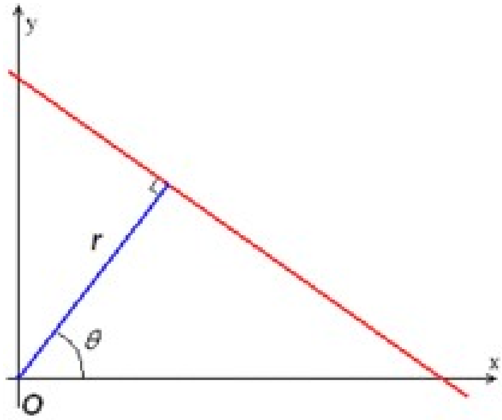


Figure 4.5: Vertical lines in Hough space

It can be coordinated with each line of view. (R, θ) plane is intended as a Hough plot for a straight line game plan in two estimates.

Given a line point in the plane, by then the course of action of every single straight line experiencing that point relates to unusual (r, θ) sinusoidal bending. Many things that give a straight line structure create sinusoids passing through this line (r, θ). Thus, recognition of collinear recommendations for recognition of symphonic species can be changed.

The capacity utilized is **cv2.HoughLinesP()**. It has two new contentions.

- **minLineLength**
- **maxLineGap**

See the results below:

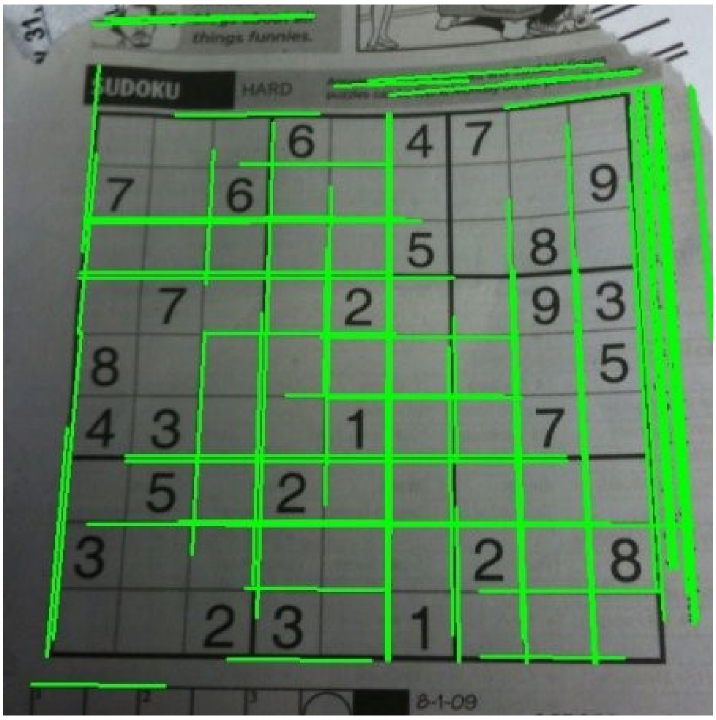


Figure 4.6: Results of Hough Transform

### *4.3.1 Canny Edge Detector*

The Canny edge marker is an edge affirmation director that utilizes a multi-compose tally to recognize a wide degree of edges in pictures. Canny in like manner conveyed a computational speculation of edge disclosure clearing up why the strategy works.
Vigilant edge discovery is a technique to isolate accommodating fundamental information from different vision objects and altogether decrease the proportion of data to be taken care of. It has been commonly associated in different PC vision frameworks. Watchful has found that the necessities for the use of edge area on grouped vision structures are commonly relative. Thusly, an edge acknowledgment answer for area these requirements can be completed in a wide extent of conditions.

To satisfy these necessities Canny used the investigation of assortments – a strategy which finds the limit which improves a given pragmatic. The perfect limit in Canny's marker is depicted by the aggregate of four exponential terms, yet it will in general be

approximated by the primary subordinate of a Gaussian.
Among the edge revelation systems grew as of not long ago, shrewd edge distinguishing proof figuring is a champion among the most deliberately portrayed methodologies that gives extraordinary and trustworthy acknowledgment. Inferable from its optimality to meet with the three criteria for edge disclosure and the ease of methodology for execution, it wound up a champion among the most unmistakable computations for edge recognizable proof.
A function is used for this called, **cv2.Canny()**.

Figure 4.7: Canny Edge Detector

#### *4.3.2 Process of algorithm*

The Process of Canny edge recognizable proof figuring can be isolated to 5 special advances:

1. Applying Gaussian blur to even the image in order to clear the noise
2. Apply non-most extraordinary covering to discard misdirecting response to edge acknowledgment.
3. Apply edge point of confinement to pick potential edges.
4. Finalize the discovery of edges by smothering the various edges that are feeble and not associated with solid edges.
5. Discover the force slopes of the picture.

#### *4.3.3 Stages*

It is a multi-organize calculation and we will experience every stage:

1. **Noise Reduction**

   Since the defect of the edge is flawless, the initialization of the work is to discharge the body of the picture with 5x5 Gauss.

2. **Discovering Gradient of the Image**

   Evened picture it is then filtered through both the lower and the vertical divisions to form a lineline to obtain the first sub-dimension path. From each of these photos, we can discover the edge slope and strength for each pixel as pursues: par Inclination bearing is constantly opposite to edges.

$$Edge\_Gradient\ (G) = \sqrt{G_x^2 + G_y^2}$$
$$Angle\ (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Figure 4.8: Equations of Canny

3. **Non greatest Suppression**

   Following the angle measurement and orientation, a complete image output for unloading any unwanted pixel that can not open its edge. To do this, each pixel is checked if the pixel is at the most extreme slope near it. Picture beneath:

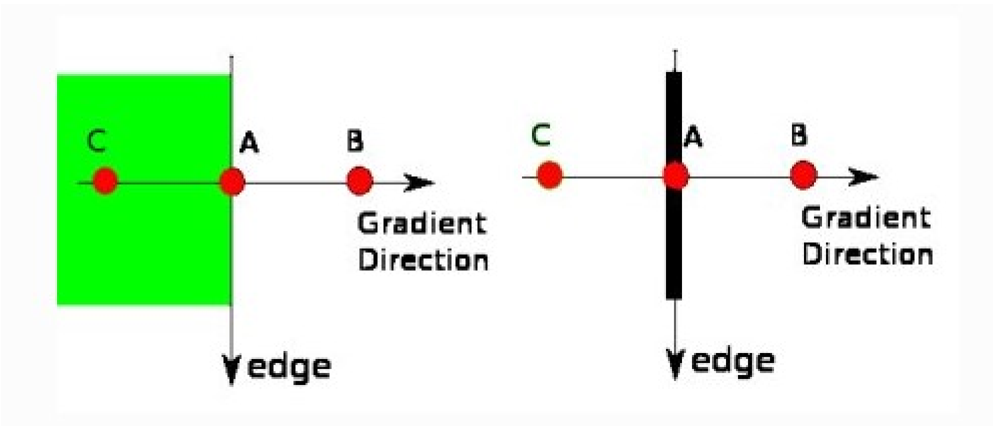


Figure 4.9: Suppression

Point A is on edge. The slope of the bed is an ordinary situation. B point B and C are sloping headers. Thus, from point A, B and C routers are checked to see if there is the most sensitive image in the vicinity. If this is true, it is intended for the next stage, but it is drowned. The result you get to explicitly expose this picture is twice as picture with ”thin edge”.

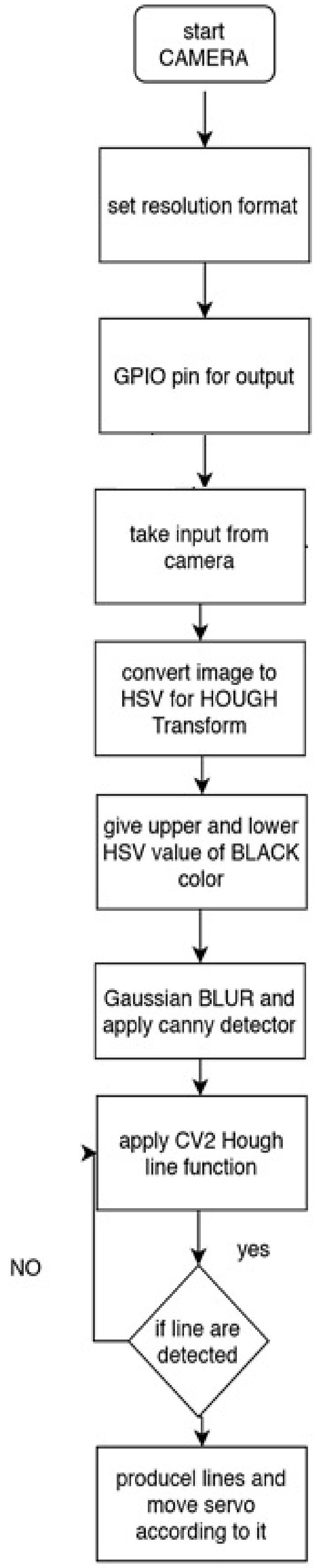


Figure 4.10: Hough transform Flow Chart

## 4.4 HSV Image Conversion

HSV (hue, saturation, value) are elective portrayals of the RGB shading model, structured during the 1970s by computer illustrations specialists to all the more intently line up with the manner in which human vision sees shading making properties. In these models, shades of each tone are masterminded in an outspread cut, around a focal hub of unbiased hues which ranges from dark at the base to white at the top. The HSV portrayal models the route paints of various hues combine, with the immersion measurement looking like different shades of brilliantly hued paint, and the esteem measurement taking after the blend of those paints with fluctuating measures of dark or white paint. HSV is described in a way that resembles how individuals see shading. It relies upon three characteristics: tint, immersion, and esteem. This shading space portrays tones (tint) similar to their shade (drenching or proportion of dull) and their brightness regard.

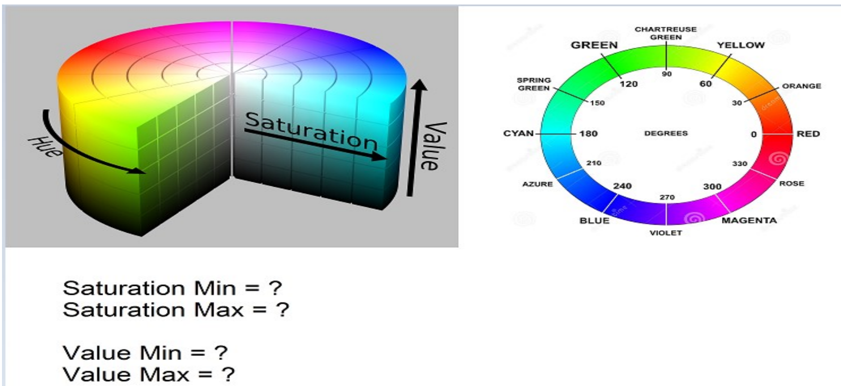


Figure 4.11: HSV scheme

**HSV Image Conversion** is used in our project for traffic light detection. We want to detect **red** and **green** light. Red light for signaling that the car needs to stop until the green light appears. So, we have stored the upper and lower ranges of the colors in our algorithm so that when the image is converted from RGB to HSV, inputting from the camera module, when that range of specific color appears, we have to perform the specific tasks which are defied in our code.
Red color range:
Green color range:

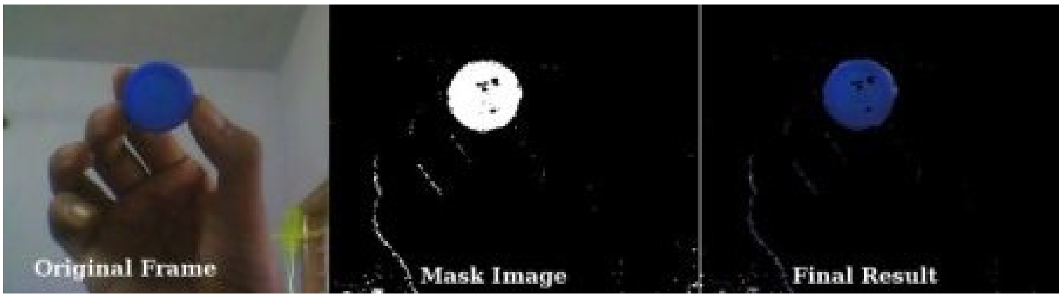


Figure 4.12: Result of HSV masking

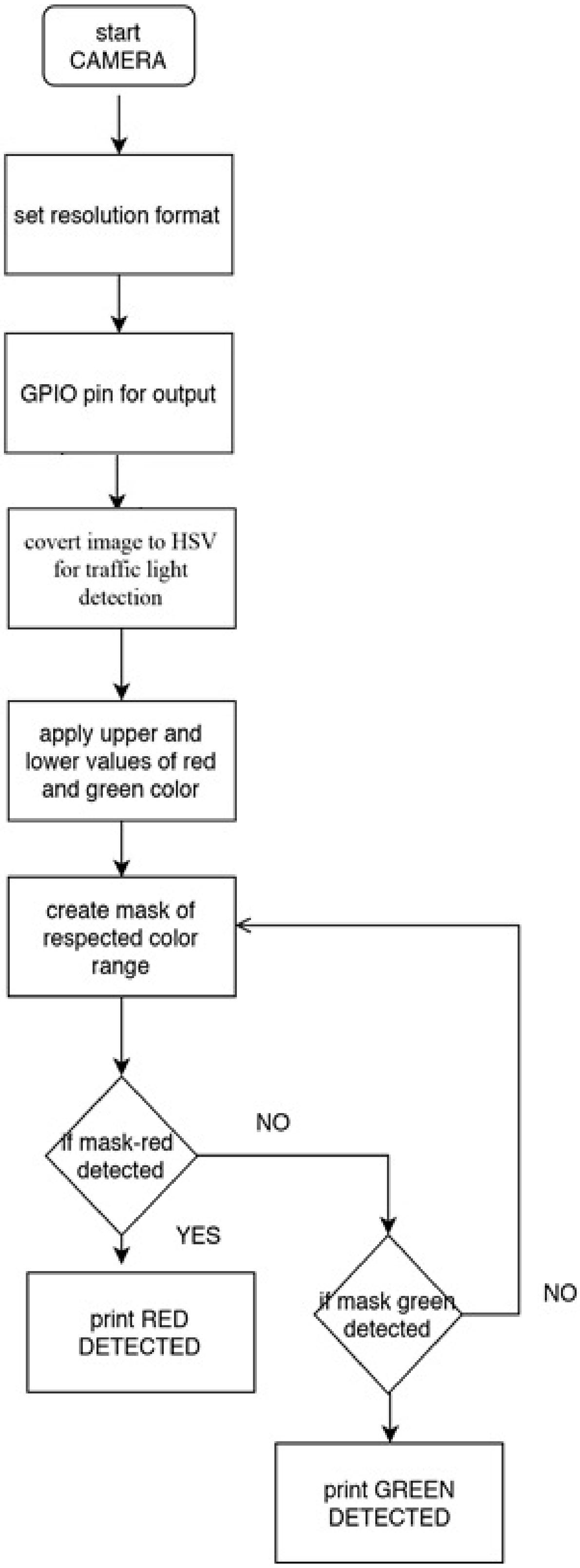


Figure 4.13: HSV image Flow Chart

# Chapter 5
# Result and Discussion

Results are produced after doing this project. We have combined different image processing algorithms to perform our specified task. And all of them are producing some kind of results in any manner.

Ultrasonic sensor used for distance measurement is also working fine and give distance in centimeters (cm).

## 5.1 Lane detection

Lane detection is done using **Hough Transform**. It used for our car to work properly and detect lane in front of it so that it can move in that path. The input from camera is sent to the processor, which runs the specified algorithm on that image giving us the results after detecting the lane of our controlled environment.

The results after processing tells us whether the road is straight, turning left or right. If the road is straight, we have commanded the motors to move straight. DC motor is running on PWM mode and servo motor, which we used for steering purpose, remains at **neutral position** or at 90°.

But if the result gives us that the road is turning left in front of it, now comes the main part. We have commanded our servo motor to take a left turn by using its position to be at 65°. Similarly, if the results give us that the road is turning right, the servo position is changed to 115°. In this way, it takes turn either left or right.

**hough = cv.HoughLinesP(segment, 2, np.pi / 180, 100, np.array([]), minLineLength = 100, maxLineGap = 50)**

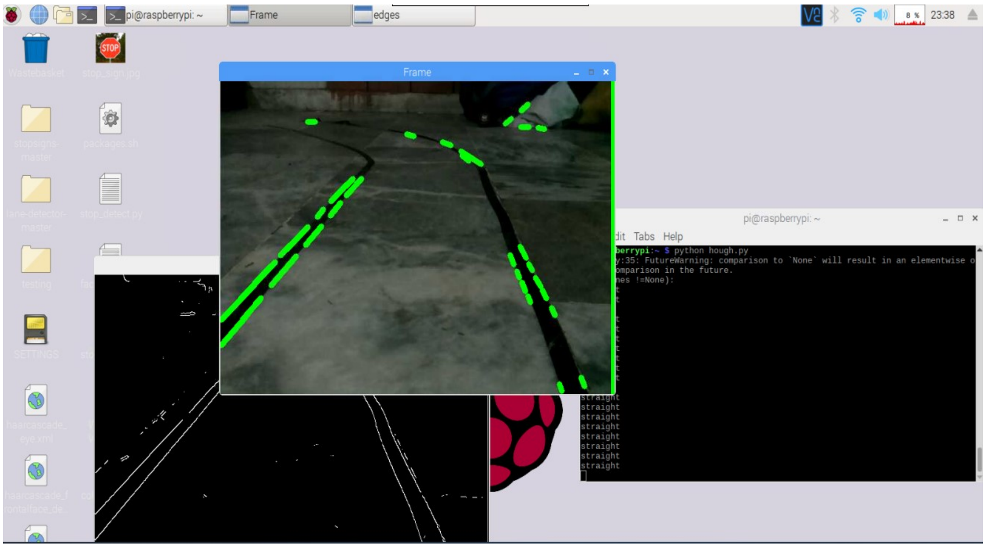


Figure 5.1: Lane Detection

## 5.2 Stop-Sign Detection

Stop sign detection is done using **Haar-cascade Classifier**. This sign is detected on road for telling us to wait for **5 seconds** due to the stop sign.

These classifiers and their working is described in section 4.2 earlier. When the stop sign is detected in front of camera view, it give an array value i.e. [192 167 85 91] to the variable stop. We use this value in our algorithm.

If stop variable gets any array value, we have introduced a delay in our program and all the motor working are stopped for that time of delay i.e. for 5 seconds. And the detection of sign is also highlighted with a blue square around it. Showing the exact detection and location of sign.


Figure 5.2: Stop sign detection-1

As this is a prototype project, by stop sign detection, we have introduced that any sign either left, right or straight, any sign can be detected by using Haar-cascade classifier. We just have to load the specific classifier file in the code and it will work same as this. Here, we have loaded stop sign classifier.

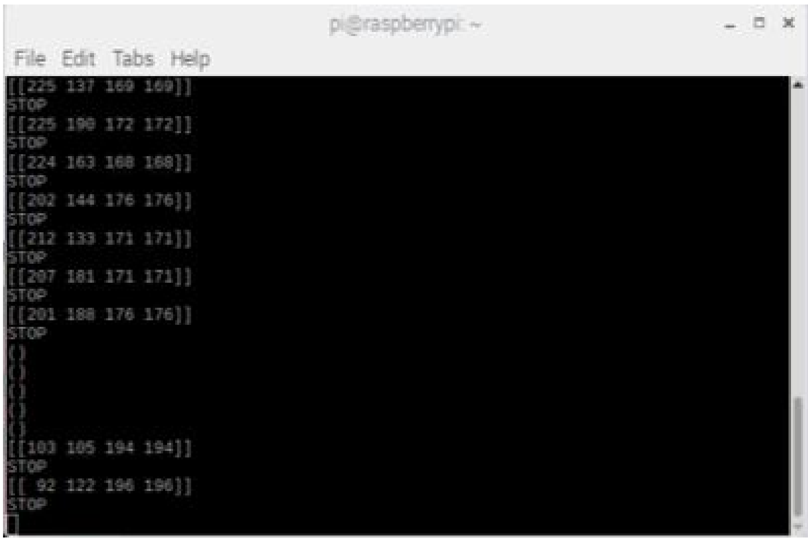


Figure 5.3: Stop sign detection-2

## 5.3 HSV image conversion

HSV image conversion is done for **detecting traffic light signals.**

In this algorithm, the input image from camera module sent to the processor is converted in HSV range. For traffic lights, we need red and green color ranges. So after calculating the red and green color HSV upper and lower ranges, we have stored them into the program in a variable. And the mask of the input image is always showing on the display. Whenever, the camera gets any red or green object in front of it, the upper and lower HSV values stored in the variable are used and mask identifies that object on the screen. When red color is detected in front of camera, the program commands the processor to stop the motors for a while until the green color of traffic light is not detected.

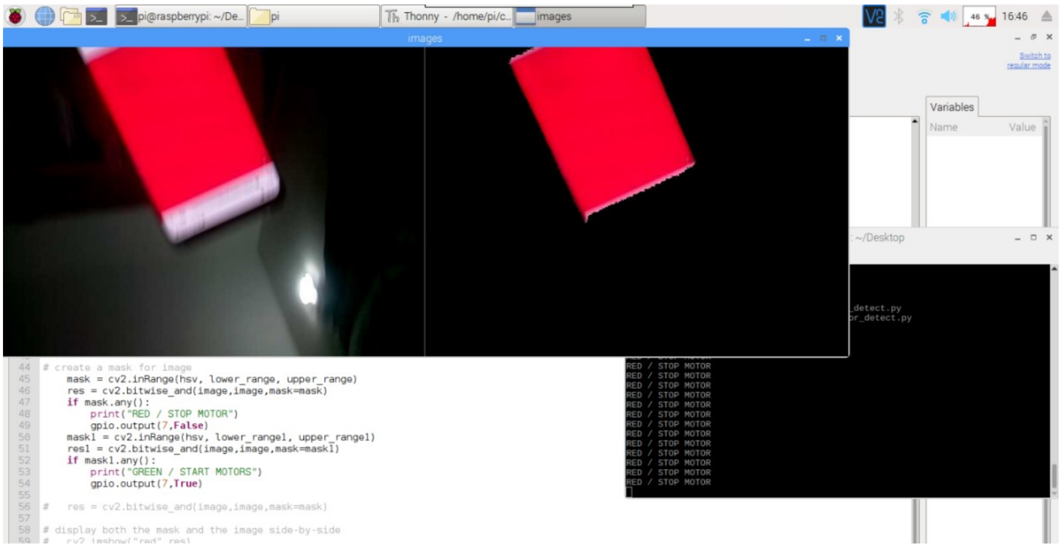

Figure 5.4: Red color detection

When green color is detected, the motor starts to work again in a regular manner.

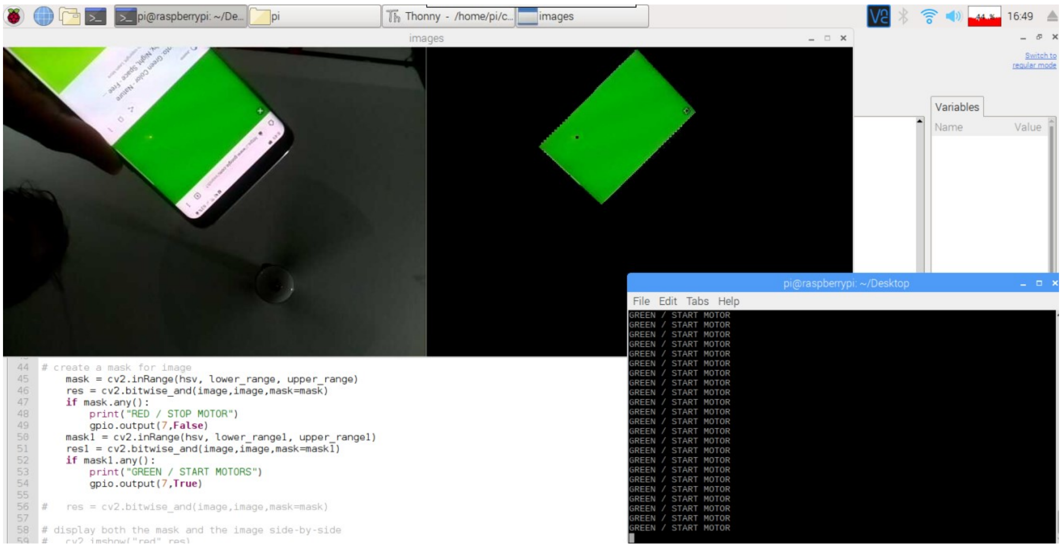


Figure 5.5: Green color detection

# Chapter 6
# Conclusions and Recommendations

## 6.1 Conclusions

1. We initially started with nothing and have implemented a totally driverless car in a controlled environment which can operate itself and take decision according to the environment.

2. All the decisions are done by processor after working on different algorithms in the code which are executed at the same time. The main aim was to make it a collision free vehicle, which can sense the hurdle in front of it and can overtake it with good precision.

3. Image processing was slow at first because of different algorithms working together at the same time, but we are still working on it to make it faster somehow and produce good GPU. So that this project can get any better in near future.

4. The prototype car is fully autonomous; we just have to give it power and all the algorithms work on their own and car works perfectly fine.

## 6.2 Recommendations

1. In future, it can be brought on the road.

2. We can alternate with the speed of image processing algorithms.

3. More accuracy and speed.

# References


[1] C. Urmson *et al.* (2008) Self-driving cars and the urban challenge.

[2] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.* (2016) End to end learning for self-driving cars.

[3] M. Gerla, E.-K. Lee, G. Pau, and U. Lee. (2014) Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. IEEE.

[4] J. Ebken, M. Bruch, and J. Lum. (2005) Applying unmanned ground vehicle technologies to unmanned surface vehicles. International Society for Optics and Photonics.

[5] J. Boeglin. (2015) The costs of self-driving cars: reconciling freedom and privacy with tort liability in autonomous vehicle regulation.

[6] A. Hars. (2015) Self-driving cars: The digital transformation of mobility.

[7] J. Yang and J. Coughlin. (2014) In-vehicle technology for self-driving cars: Advantages and challenges for aging drivers.

[8] J. Allard, D. S. Barrett, M. Filippov, R. T. Pack, and S. Svendsen. (2009, Mar. 3) Systems and methods for control of an unmanned ground vehicle. US Patent 7,499,776.

[9] D. W. Gage. (1995) Ugv history 101: A brief history of unmanned ground vehicle (ugv) development efforts.

# Appendix A

```
from picamera.array import PiRGBArray
import RPi.GPIO as gpio
from picamera import PiCamera
import time
import cv2
import numpy as np
import math
import imutils

gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)

servo = 18
PIN_TRIGGER = 7
PIN_ECHO = 37

gpio.setup(11, gpio.OUT)        #Motor pins
gpio.setup(12,gpio.OUT)
#GPIO.setup(13,GPIO.OUT)
gpio.setup(servo,gpio.OUT)      #servo pin
pq = gpio.PWM(11 , 50)
pq.start(45)            # MOTOR PWM INITIAL

gpio.setup(PIN_TRIGGER, gpio.OUT)
gpio.setup(PIN_ECHO, gpio.IN)
gpio.output(PIN_TRIGGER, gpio.LOW)


camera = PiCamera()
camera.resolution = (320, 240)
camera.rotation = 180
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(320, 240))
time.sleep(0.1)
stop_classifier= cv2.CascadeClassifier('stopsign_classifier.xml')

p = gpio.PWM(servo, 50) # servo pin for PWM with 50Hz
p.start(7.5) # Initialization

theta=0
minLineLength = 5
maxLineGap = 50

#RED COLOR
lower_range = np.array([169,100,100], dtype=np.uint8)
upper_range = np.array([189,255,255], dtype=np.uint8)

#GREEN COLOR
lower_range1 = np.array([52,100,100], dtype=np.uint8)
upper_range1 = np.array([79,255,255], dtype=np.uint8)


for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
        fr = frame.array
        image = imutils.resize(fr, width=200)
        time.sleep(0.1)

        hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
        gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
        blurred=cv2.GaussianBlur(gray, (5,5),0)
        edged = cv2.Canny(blurred, 75,150)
        gpio.output(12, gpio.HIGH)
        gpio.output(11, gpio.HIGH)
```

```
        gpio.output(PIN_TRIGGER, gpio.HIGH)
        time.sleep(0.00001)
        gpio.output(PIN_TRIGGER, gpio.LOW)

        while gpio.input(PIN_ECHO)==0:
                pulse_start_time = time.time()
        while gpio.input(PIN_ECHO)==1:
                pulse_end_time = time.time()

        pulse_duration = pulse_end_time - pulse_start_time
        distance = round(pulse_duration * 17150, 2)
#               print "Distance:",distance,"cm"
        time.sleep(0.2)

        if distance < 50:
                p.ChangeDutyCycle(12.5) #180
                time.sleep(0.8)
                p.ChangeDutyCycle(2.5) #0
                time.sleep(2)
#               p.ChangeDutyCycle(7.5) #90
#               time.sleep(1.8)
#               p.ChangeDutyCycle(2.5) #0
#               time.sleep(1.3)
                p.ChangeDutyCycle(12.5) #180
                time.sleep(0.8)
                p.ChangeDutyCycle(7.5) #90
                time.sleep(0.8)
# Lines
        lines = cv2.HoughLinesP(edged,1,np.pi/180,10,maxLineGap)
        if(lines !=None):
                for x in range(0, len(lines)):
                        for x1,y1,x2,y2 in lines[x]:
#               for line in lines:
#                               x1,y1,x2,y2 = line[0]
                                cv2.line(image,(x1,y1),(x2,y2),(0,255,0),8)
                                theta=theta+math.atan2((y2-y1),(x2-x1))
        threshold=6
        if(theta>threshold):
                print("left")
                p.ChangeDutyCycle(5.5)
                time.sleep(0.5)
        if(theta<-threshold):
                print("right")
                p.ChangeDutyCycle(9.5)
                time.sleep(1)
        if(abs(theta)<threshold):
                print ("straight")
                p.ChangeDutyCycle(7.5) # 0
                time.sleep(1)
        theta=0
# STOP SIGN
        stop=None
        stop = stop_classifier.detectMultiScale(gray, 1.02, 10)
        if stop is not None:
                for (x,y,w,h) in stop:
                        cv2.rectangle(image,(x,y),(x+w,y+h),(255,0,0),2)
                        print("STOP SIGN")
                        gpio.output(12,gpio.LOW)
                        time.sleep(3)
                        gpio.output(12,gpio.HIGH)
```

```
# create a mask for image
        mask = cv2.inRange(hsv, lower_range, upper_range)
        res = cv2.bitwise_and(image,image,mask=mask)
        if mask.any():
                print("RED / STOP MOTOR")
                gpio.output(12,gpio.LOW)
        mask1 = cv2.inRange(hsv, lower_range1, upper_range1)
        res1 = cv2.bitwise_and(image,image,mask=mask1)
        if mask1.any():
                print("GREEN / START MOTORS")
                gpio.output(12,gpio.HIGH)

# RESULTS
        cv2.imshow("Frame", image)
        cv2.imshow("RED / GREEN" , np.hstack([res,res1]))
        cv2.imshow("edges",edged)
        key = cv2.waitKey(1) & 0xFF
        rawCapture.truncate(0)
        if key == ord("q"):
                break
cv2.destroyAllWindows()
```

# Appendix B:

| Equipment | Cost (PKR) |
|---|---|
| Car Chassis | 10,000 |
| Raspberry Pi | 6,000 |
| Pi Camera | 3,300 |
| Battery | 3,800 |
| Motors | 1,800 |
| Ultrasonic Sensors | 500 |
| Miscellaneous | 4,000 |
| Total | 33,200 |

Table 6.1: Cost Analysis

| Progress | Date |
|---|---|
| Literature Review & Topic Selection | February'18 – April'18 |
| Chassis Designing | May'18 – June'18 |
| Language Learning | July'18 – November'18 |
| Testing & Troubleshooting | December'18 – February'19 |
| Documentation | March'19 |
| Final Presentation | April'19 – May'19 |

Table 6.2: Timeline