AUTONOMOUS VEHICLE

Final year Project Report

Submitted to the APJ Abdul Kalam Technological University

in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Electronics and Communication Engineering

by

AHMED RISHAD B A (TVE19EC008)

AKHIL J (TVE19EC009)

ATHUL K (TVE19EC021)

FEBIN D SAM (TVE19EC026)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING TRIVANDRUM

KERALA

JUNE 2023

DEPT. OF ELECTRONICS & COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING TRIVANDRUM

2022 - 23

CERTIFICATE

This is to certify that the report entitled **AUTONOMOUS VEHICLE** submitted by **AHMED RISHAD B A** (TVE19EC008), **AKHIL J** (TVE19EC009), **ATHUL K** (TVE19EC021) & **FEBIN D SAM** (TVE19EC026) to the APJ Abdul Kalam Technological University in partial fulfillment of the B.Tech. degree in Electronics and Communication Engineering is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Jinu Jayachandran
(Project Guide)
Assistant Professor
Dept. of ECE
College of Engineering
Trivandrum

Prof. Adersh V. R
(Project Coordinator)
Assistant Professor
Dept. of ECE
College of Engineering
Trivandrum

Dr. Pradeep R
(Project Coordinator)
Professor
Dept. of ECE
College of Engineering
Trivandrum

Dr. Joseph Zacharias
(Head of Department)
Professor and Head
Dept. of ECE
College of Engineering
Trivandrum

DECLARATION

We hereby declare that the project report **AUTONOMOUS VEHICLE**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Prof. Jinu Jayachandran

This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

AHMED RISHAD B A

AKHIL J

ATHUL K

FEBIN D SAM

Trivandrum

16-06-2023

Abstract

Autonomous Vehicle, also known as self-driving car, is a vehicle capable of sensing its environment and operating with least human involvement or without any involvement at all. Often considered the future of mobility, an autonomous vehicle is an active area of research and development. It is a multidisciplinary endeavour that actively exploits the emerging fields of Artificial Intelligence, the Internet of Things, Embedded Systems, and conventional Automobile Technology.

In this project, we intend to build a vehicle capable of automatic lane adjustment, obstacle detection, sign board identification, traffic light response, and remote access control. The primary objective is to sense the environment and extract relevant information. The visuals are captured by a camera and each frame is processed for feature extraction. The lanes are identified and the vehicle is aligned. Machine Learning algorithms are used for object and sign board identification. The data associated with the vehicle such as speed, distance, obstacles detected, etc is made remotely available using IoT.

Acknowledgement

We take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us to complete this work successfully. We express our sincere thanks to Dr. Joseph Zacharias, Head of Department, Electronics and Communication Engineering, College of Engineering Trivandrum for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to the Prof. Adersh V. R, Department of Electronics and Communication Engineering, College of Engineering Trivandrum and Dr. Pradeep R, Department of Electronics and Communication Engineering, College of Engineering Trivandrum for the support and co-operation.

We would like to place on record our sincere gratitude to our project guide Prof. Jinu Jayachandran, Assistant Professor, Electronics and Communication Engineering, College of Engineering Trivandrum for the guidance and mentorship throughout this work.

Finally, we thank our family and friends who contributed to the successful fulfillment of this project work.

AHMED RISHAD B A

AKHIL J

ATHUL K

FEBIN D SAM

Contents

List of Figures

List of Symbols

Ω Unit of Resistance

$\varepsilon^{'}$ Real part of dielectric constant

c Speed of light

λ Wavelength

δ Delta

Chapter 1

Introduction

Autonomous Vehicle is the future of mobility and transportation. It refers to a vehicle capable of sensing its environment and operating on its own without any human involvement. The evolution of vehicles dates back to ancient civilisation, when wheels were first discovered. Horses and chariots bullock carts were used for several millennia as a common mode of transportation. The commercial vehicles came out in 1672 with the invention of steam engines. After several years came the Electric Vehicles (EVs) in 1832.

The automobile industry was revolutionized after the invention of IC Engines and for several years they have been dominating the market. In the initial days, vehicles were fully operated manually. As technology advances, vehicles are getting smarter day by day. Today Vehicles are equipped with Advanced Driver Assistance Systems (ADAS) which makes driving much easier. As Artificial Intelligence (AI) and Embedded Systems are becoming more prominent, scientists today actively engage in the development of vehicles that are capable of driving on their own. The Society of Automotive Engineers (SAE) has classified autonomous vehicles into six levels based on the degree of automation. [1]

Level 0 Automation (No Driving Automation)

Under this level the vehicle is operated manually by the user. Most of the vehicles today come under this category. The driver is primarily responsible for controlling the

vehicle. However there maybe some systems like cruise control,ABS present in the vehicle to help the user.

Level 1 Automation (Driver Assistance)

Under this level there may be automated driving assistance systems like lane control assist,adaptive cruise control etc. There maybe advanced features to enhance safety of the vehicle. However the driver is very much incharge of controlling the vehicle.

Level 2 Automation (Partial Driving Automation)

Under this level the main features are the Advanced Driver Assistance Systems. They include automatic braking,accelaration,steering etc. The vehicle is still incapable of moving on its own and the driver is incharge of the control.

Level 3 Automation (Conditional Driving Automation)

The vehicles in this category are able to drive themselves but only under certain conditions. They are capable of driving on long distance roadways. They can move on their own without the driver having hands on the steering wheel. However the driver needs to be ready to take full control of the vehicle at any point of time.

Level 4 Automation (High Driving Automation)

The vehicle is able to drive on its own under very complex situations. The driver need not have to control the vehicle but can override and take control of the vehicle at any point of time. Level 4 autonomous vehicles can drive under the normal roadways conditions but the world is not yet ready for these vehicles to take over.

Level 5 Automation (Full Driving Automation)

Under this level the vehicle doesn't need a driver. It is capable of taking any decisions. The vehicle can handle any driving conditions without an external human input.

Chapter 2

Literature Review

2.1 Autonomous Vehicle System

Autonomous vehicles integrate planning modules with external environment perceptions . According to the requirements of the planning module, the driving module of automatic vehicles manipulates the actuators, and parameters like speed and trajectory of vehicles are monitored. The control system of autonomous vehicles is responsible for the safe and stable movement of vehicles. This control system includes both lateral and longitudinal controllers. A longitudinal controller regulates velocity and a lateral controller manipulates the steering angle of the wheel. [2]

2.1.1 Longitudinal Controller

The work of a longitudinal control system is to track the desired speed predicted by the system of planning while taking care of the safe movement of the vehicle. Input is required for longitudinal control systems are position and speed. The position required for maintaining a safe distance from other vehicles at the same time keeping a check on the current speed. The desired speed is mentioned by a system of planning. The output from a longitudinal control system manages two of the actuators i.e., brake and throttle. They regulate the vehicle's speed and maintain a safe distance from other vehicles. The system is described by the equation 2.1 below.

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt} \tag{2.1}$$

where K_P, K_I and K_D are representations of proportional, integral and derivatives gain in time domain t. PID controller u(t), acts as an input to the plant model.

The plant system can be linear or non-linear depending on functionalities. It deals with state space and transfer functions. Linear time variant systems can be expressed by transfer function. Transfer function (G) is a relation between input U and output Y.

$$Y(s) = G(s)U(s) \tag{2.2}$$

$$s = +j \tag{2.3}$$

Transfer function, is expressed in Laplace domain, as a function of S, a complex variable. Main reason to use Laplace transform is that, it makes it easier to determine transform function of plant system. Laplace transform provides a better and useful insight in understanding control performance. Applying Laplace transform of PID control generates:

$$U(s) = G_C(s)E(s) = \frac{K_D s^2 + K_P s + K_I}{s} E(s) \tag{2.4}$$

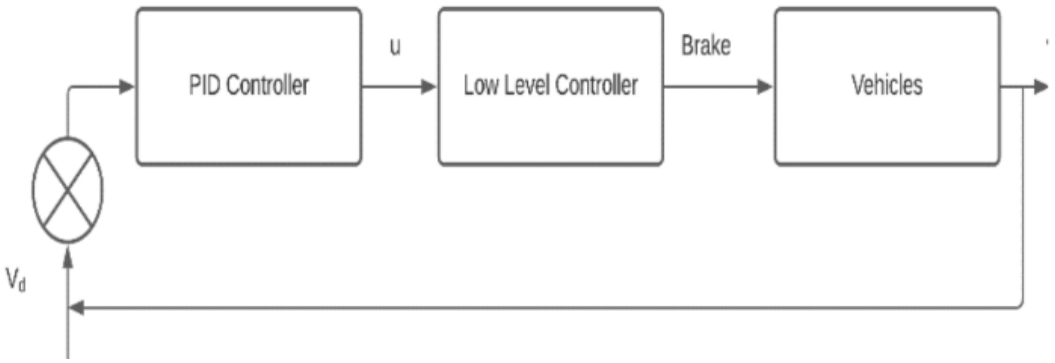


Figure 2.1: Longitudinal Control

2.1.2 Lateral Controller

Planning system generates the desired trajectory by using lateral controller. Orientation errors, cross track errors and their derivatives (calculated by pose and position of vehicles from planning module and perception module), are used by lateral controller as inputs. Output generated by lateral controller controls the steering actuators. For better analysis of vehicle dynamic, lateral characteristics of vehicles are analyzed by implementing it using a bicycle model. For the lateral control system, a geometric

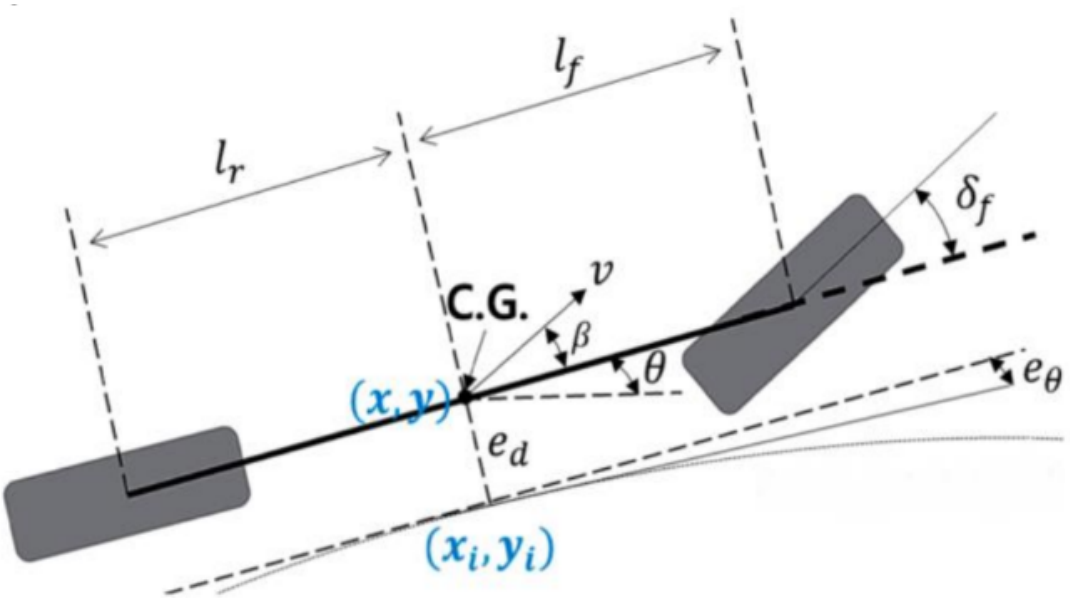


Figure 2.2: Bicycle Model of a Vehicle

controller, Stanley is implemented. It is a simple and effective steering control law used by Stanford University's DARPA grand challenge team. It is represented as:

$$\delta(t) = e_2(t) + tan^{-1}\frac{Ke_1(t)}{v_f(t)}, \delta(t)\epsilon[\delta_{min}, \delta_{max}] \tag{2.5}$$

In the law, $e_1(t)$ represents cross-track error, $e_2(t)$ represents orientation error, and k represents gain parameter. Grain parameter determines convergence rate. Cross-track error is distance from center of front axle to closest point on path. For large positive change in cross- track, it is represented by:

$$\frac{K_e(t)}{V_f(t)} \approx \frac{\pi}{2} \rightarrow \delta(t) \approx \omega(t) + \frac{\pi}{2} \tag{2.6}$$

The error dynamics for model, when it is not at maximum steering angle small cross-track errors, can lead to exponential decay characteristics.

$$\dot{e}(t) = -v_f(t)sin(\omega(t) - \delta(t)) \tag{2.7}$$

$$e(t) \approx -ke(t) \tag{2.8}$$

The Stanley control law neglects dynamics like noisy measurements, tire force effects, mass etc. For adjusting these dynamics, certain adjustments are made in the control system. During low-speed operation, inverse speed causes numerical instability. To overcome it, softening constant K_s is added to the controller

$$\delta(t) = e_2(t) + tan^{-1}\frac{K_{e1}(t)}{K_s + v_f(t)} \tag{2.9}$$

2.2 Image Processing

The tasks of processing and analyzing images and videos are encountered in the development of onboard data processing and control systems and, in particular, systems for automatic object detection and tracking . Also, these tasks are encountered in the field of transport analytics due to the urgent problem of the growing number of vehicles on the roads. Image and video analysis algorithms allow us to obtain information which may include the number of objects moving in interest directions; object flow density; parameters of objects, as well as determine the occurrence of emergency situations. [3]

It must be taken into account that the video analytics system must provide the real-time processing of large amounts of data presented in the form of image sequence. Often communication channels do not have the necessary bandwidth which leads to the appearance of distortions and noise in images, delays in receiving a new frame

2.2.1 Color Model

Color model which describes as a mathematical approximation of the inherently nonquantifiable nature of human visual perception.

RGB stands for red, green, blue. The RGB color model is additive: red, green, and blue light are added together in varying proportions to produce an extensive range of colors.

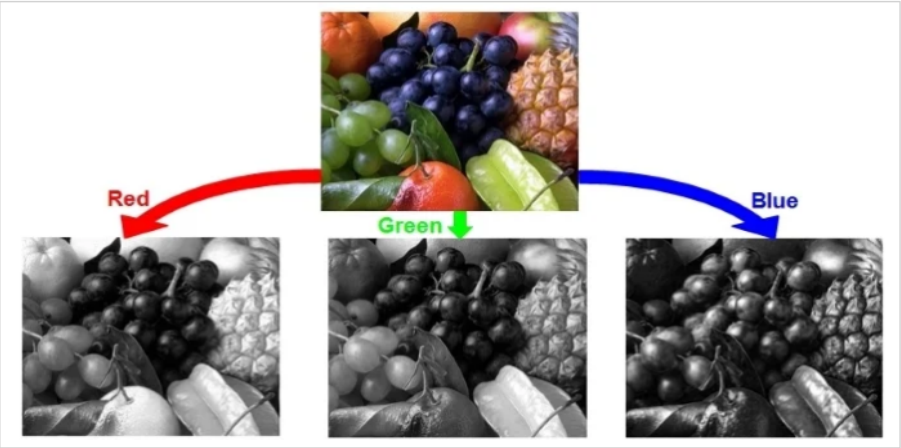


Figure 2.3: RGB colour model [7]

The gray level of an image is simply designed to have colors that are all expressed in gray. Indeed, ”gray” is a color in which all the components: red, green and blue have the same intensity in the RGB space (Red, Green and Blue)

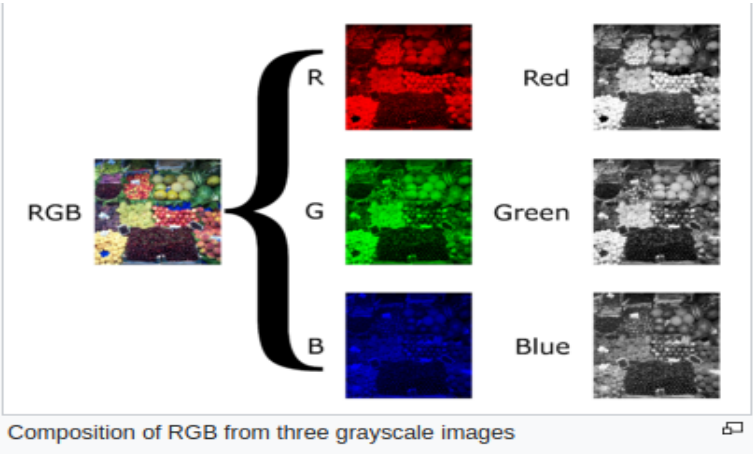


Figure 2.4: Grey Model [8]

In computing, although the grayscale can be computed through rational numbers, image pixels are usually quantized to store them as unsigned integers, to reduce the required storage and computation. Some early grayscale monitors can only display up to sixteen different shades, which would be stored in binary form using 4 bits. [citation needed]But today grayscale images intended for visual display are commonly stored with 8 bits per sampled pixel. This pixel depth allows 256 different intensities (ie, shades of gray) to be recorded, and also simplifies computation as each pixel sample can be accessed individually as one full byte

2.3 Lane Detection

Lane detection usually requires the use of relevant algorithms to extract the pixel features of the lane line, and then the appropriate pixel fitting algorithm is used to complete the lane detection[2]

Figure 2.5: Lane Detection [9]

2.3.1 Bird-view Transformation

A novel approach called Common bird-view Transformation used to tackle the domain shift problem for lane detection,. We notice that the distinct geometric properties of lane lines in various new test scenarios are caused by their camera viewpoint settings. Since lane lines are always painted on road surfaces and with the planar assumption, these special geometric structures of lane lines provide a method to explicitly transform the lane spatial feature into arbitrary camera view angles. By

aligning spatial characteristics of lane lines, the cross-domain shift is expected to be alleviated. Inspired of this, we introduce a virtual common aerial view point, and transform lane lines in all test scenarios into this common viewpoint. Through this simple operation, all lane lines will share similar geometry properties of intervals, positions, slopes, etc. [4]

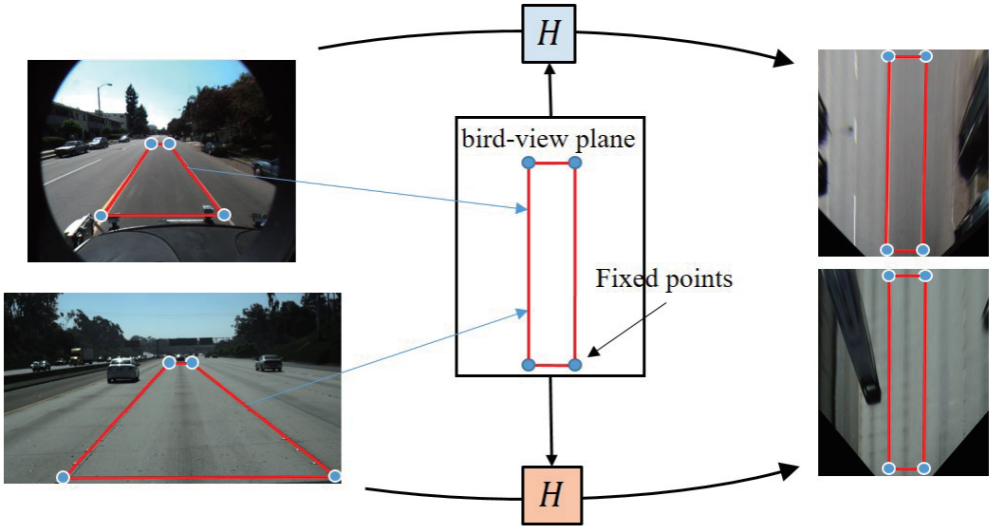


Figure 2.6: Bird-view Transformation [4]

2.3.2 Lane Marking

Given that the focus of the detection pipeline is on lane lines, the next step is to create a binary image that emphasizes pixels that correspond to lane markings. Intuitively, this could be accomplished through RGB thresholding, as lane markings are expected to be either yellow or white. However, RGB thresholding is known to perform poorly under wide ranges of brightness for any given color (e.g. markings under shadows versus direct sunlight). For this reason, it's helpful to perform thresholding using a different color space such as gray scale that is more invariant to differing levels of brightness.

The next step is to segment pixels that correspond to the left and right lanes. First, x-coordinates corresponding to the bottom of the image for each lane are estimated and then a sliding-windows search is performed. In order to find these two x-coordinates, a column-wise count of white pixels is taken for the bottom half of the image. The two maximums of this count are taken to represent the x-coordinates of each lane.

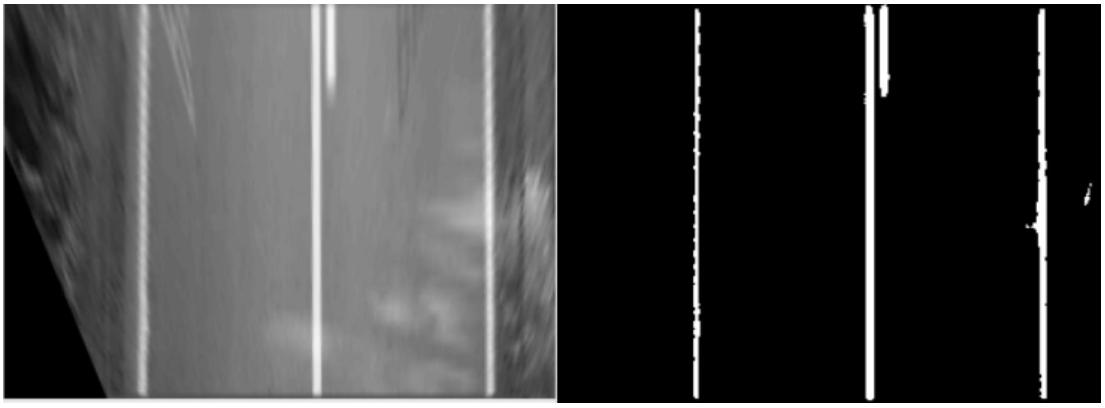


Figure 2.7: Lane Marking

2.4 Object Detection

Nowadays, video identification is established across many domains of corporations. Its usage ranges from video surveillance, sports broadcasting, electrical cars, robot navigation and many more. Either, a model can be trained for object identification or trained model could be used which would identify it the data. Although, both are techniques used in learning. They need labelled information to teach the object identification model. Or we could also use frame difference approach. Whenever an object is seen in motion, it means that the object is at a distinct position at each consecutive frame. [5]

2.4.1 Machine learning

There are several machine learning algorithms, among which we can cite the Artificial Neural Networks (ANN) which are a fairly attractive technique for computer vision problems, particularly those of recognition. There is a big number neural network models which may vary according to the number of layers, neurons, learning algorithms, and so on. The ANN(Artificial Neural Network) is a good algorithm that could help recognize patterns in an image, it can with a training set, containing 2000 images, classify an image with 96 percentage of accuracy rate.

Figure shows the structure of the neural network. One advantage of using neural network is that once the network is trained, it only needs to load trained parameters

afterwards, thus prediction can be very fast.

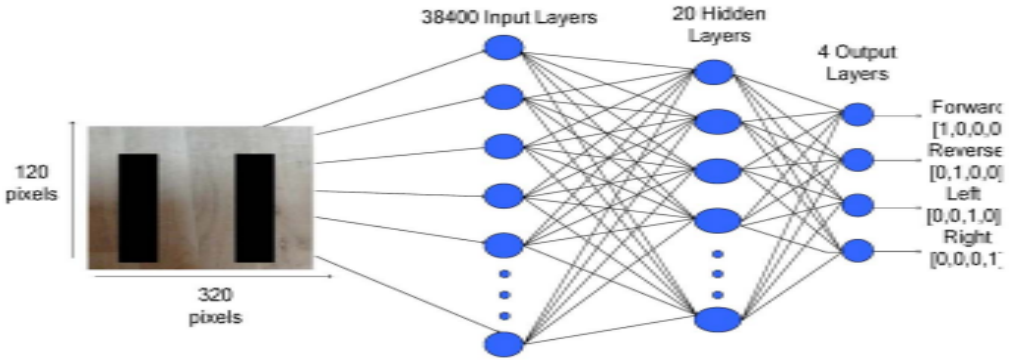


Figure 2.8: Structure of the neural network [11]

To obtain a good generalization performance, one could use the logistic regression cost function which reflects the difference between the obtained output values and the reference values .

2.4.2 Haar Cascade

Haar Cascades were first introduced in 2001, and it was one of the most popular object detection algorithms in OpenCV. Haar Cascade is an algorithm that can detect objects in images, irrespective of their scale in image and location. This algorithm is not so complex and can run in real-time. We can train a Haar Cascade detector to detect various objects like cars, bikes, buildings, fruits, etc. Haar Cascade uses the cascading window, and it tries to compute features in every window and classify whether it could be an object. Sample Haar features traverse in window-sized across the picture to compute and match features. Haar Cascade works as a classifier. It classifies positive data points → that are part of our detected object and negative data points → that don't contain our object.

- Haar Cascades are fast and can work well in real-time.
- Haar Cascade is not as accurate as modern object detection techniques are.
- Haar Cascade has a downside. It predicts many false positives.
- Simple to implement, less computing power required.

Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle. [6]

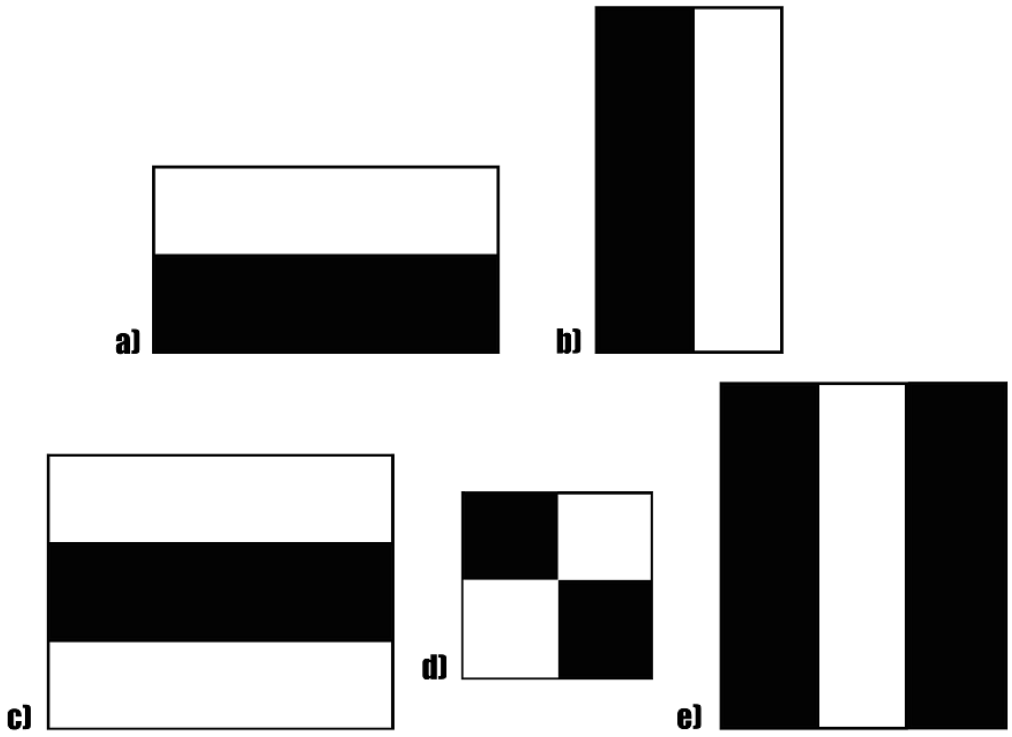


Figure 2.9: Window image [12]

Chapter 3

Methodology

Autonomous Vehicle system is a continuous feedback loop which collects data from physical environment using various sensors for navigation. The feedback from the environment is used to adjust the speed,direction and other physical attributes of the vehicle. The autonomous vehicle runs in the physical environment consisting of roads, lanes, signboards, and obstacles. The primary data used in this project are the visual images of the environment. Using a high-resolution camera mounted on the vehicle, a front view of the road is captured. Image processing is done on the obtained image to get road lanes and upcoming obstacles. Navigational changes are produced based on the above data to move the vehicle in a desired direction. Based on navigational changes, mechanical force is given to wheels and the required motion is achieved without human intervention. The general block diagram of the project is given below.

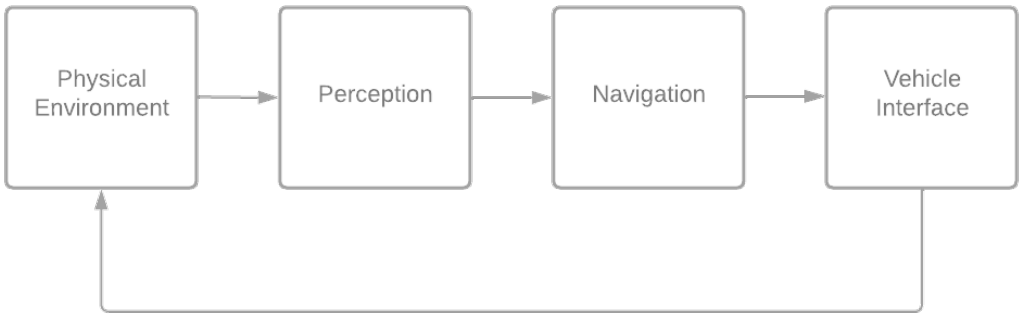


Figure 3.1: General block diagram

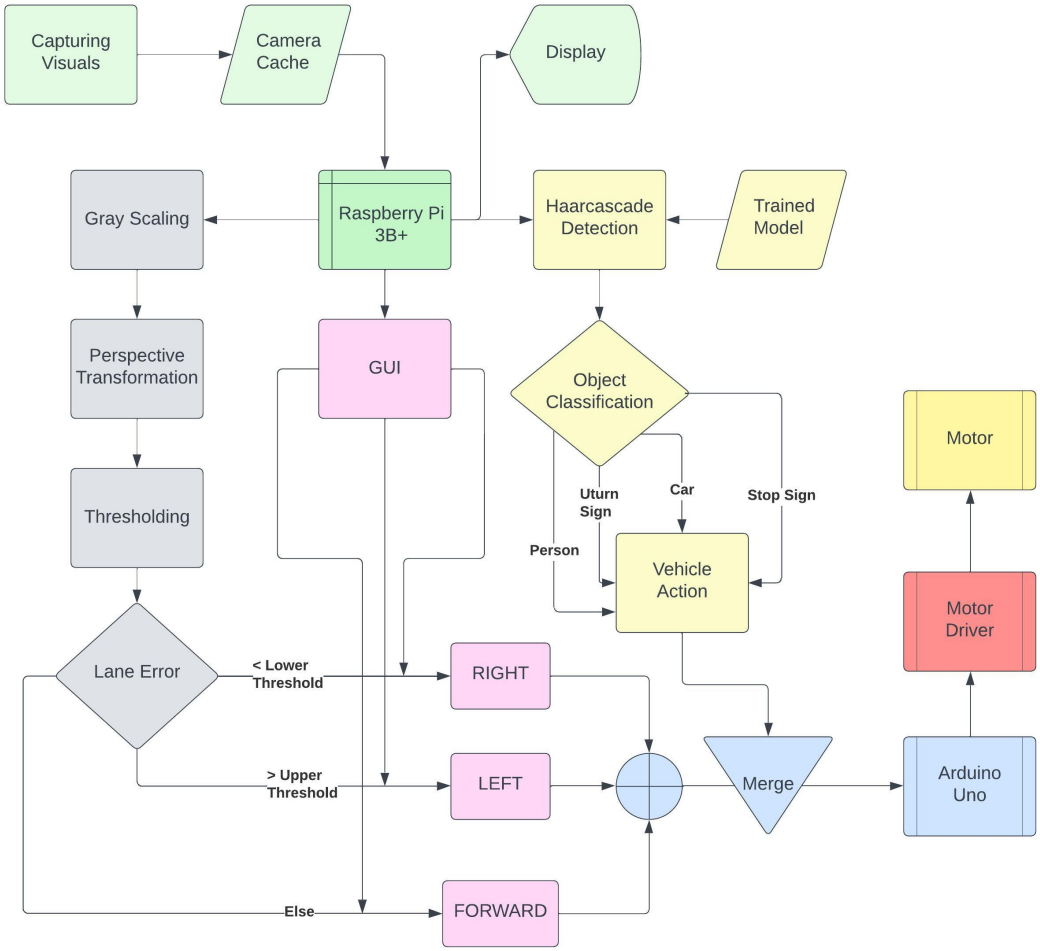


Figure 3.2: Full block diagram

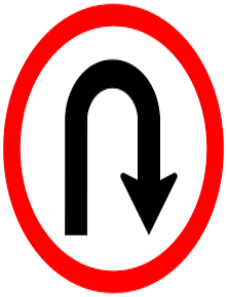
3.1 Physical Environment

Physical Environment refers to the environment in which the vehicle traverses. The algorithm for the system depends heavily upon the physical conditions in which it operates. In this project, we refer to a single-lane road and its surroundings as the physical environment. The road is evenly surfaced and there are no pits or holes that can hinder vehicle movement. The lighting should be ambient,(not too bright or not too dark) so that the features can be extracted easily. The operating temperature is taken to be the average outdoor day temperature.

In an environment there are several features which are helpful to navigate the vehicle in the desired direction. A feature is any piece of information which tells us about the

(a) STOP sign [13]

(b) Zebra sign [14]



(c) U-turn sign [15]

Figure 3.4: Sign Boards

content of an image. An environment may contain a lot features which may not be useful. So we define a Region of Interest(RoI). Region of interest helps us to process data in a desired location rather than the image as a whole. This drastically reduces the processing time. The various features found usually in a normal roadway are lanes,traffic signals,traffic signs,obstacles,vehicles,pedestrians,buildings,animals etc. Among these, lanes are the most crucial feature. The vehicle can be equipped with an automatic lane adjustment system for precise movement.

Figure 3.3: Road lanes [16]

The sign boards indicate traffic rules such 'STOP','GO SLOW','SCHOOL AHEAD' etc. While detecting the signs, it is important to note how far away they are located. A 'GO SLOW' sign may not be taken into consideration if it is located far away. Similarly traffic signals include RED for "STOP" and GREEN for "GO". The lights of a traffic signal must be distinguished from an ordinary rear brake light.

Objects on the road at real time include vehicles and pedestrians situated at various levels. It is important to differentiate each of them so that the vehicle can respond accordingly.

3.2 Perception

Perception is mainly constituted of two parts, image capturing and image processing. An overview of the process is as shown below:

3.2.1 Image Capturing

For analysing the front view,we need to capture high resolution images. With the help of Raspberry Pi 3 and and RaspiCam image capturing is realised

Raspberry Pi 3

Raspberry Pi is a series of small single-board computers (SBCs) developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries. It is widely used in many areas, such as for weather monitoring, because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of the HDMI and USB standards.

In this we are using Raspberry Pi 3 Model B+. The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market.

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4 GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	■ 2.4 GHz and 5 GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE ■ Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps) ■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	■ 1 × full size HDMI ■ MIPI DSI display port ■ MIPI CSI camera port ■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	■ 5 V/2.5 A DC via micro USB connector ■ 5 V DC via GPIO header ■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50 °C

Figure 3.5: Raspberry Pi 3 Model B+ Specifications [17]

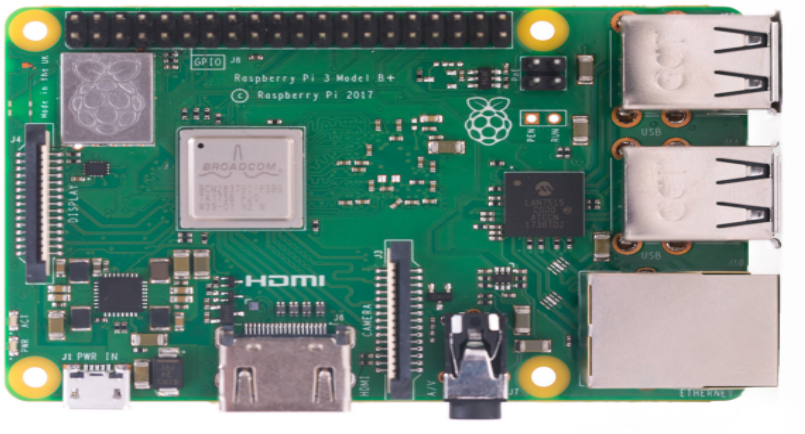


Figure 3.6: Raspberry Pi 3 Model B+ [18]

Raspbian OS

Raspbian is a free operating system based on. Debian optimized for the Raspberry Pi hardware. It comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

We can connect Raspberry using remote desktop in windows. It is done by following below steps

- Connecting Raspberry to your wifi
- Find the IP address of the raspberry pi
- Using a remote desktop connection, log in to the Raspbian

Raspberry pi camera

This raspberry pi camera has a 5-megapixel native resolution sensor capable of 2592 x 1944 pixel static images. This supports 1080p30, 720p60 and 640x480p60/90 video. The camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system.

This 5mp raspberry pi camera module which is a portable light weight camera that is capable of 1080p video and still images and connects dirctly to our raspberry pi.Capturing is done using picamera library of python, with the command "camera.capture()"

Figure 3.7: Raspberry Pi Camera [19]

3.2.2 Image Processing

The captured image is processed with help of OpenCV python library.

OpenCV

A Python package called OpenCV makes it possible to carry out image processing and computer vision tasks. It offers a variety of features, such as tracking, face recognition, and object detection. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

Figure 3.8: OpenCV [20]

These algorithms can be applied to a variety of tasks, including the detection and recognition of faces, the identification of objects, the classification of human actions in videos etc.

3.2.3 Lane detection

Lane detection can be implemented by following methods.

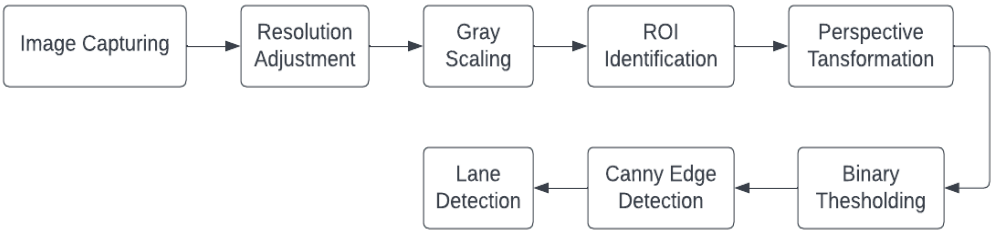


Figure 3.9: Lane Detection Algorithm

Region of Interest

We don't need a full portion of the image for lane detection. Latency due to processing also needs to be considered. A region of interest is plotted using "cv.line()" function which contains a view from the vehicle which is at some distance apart.

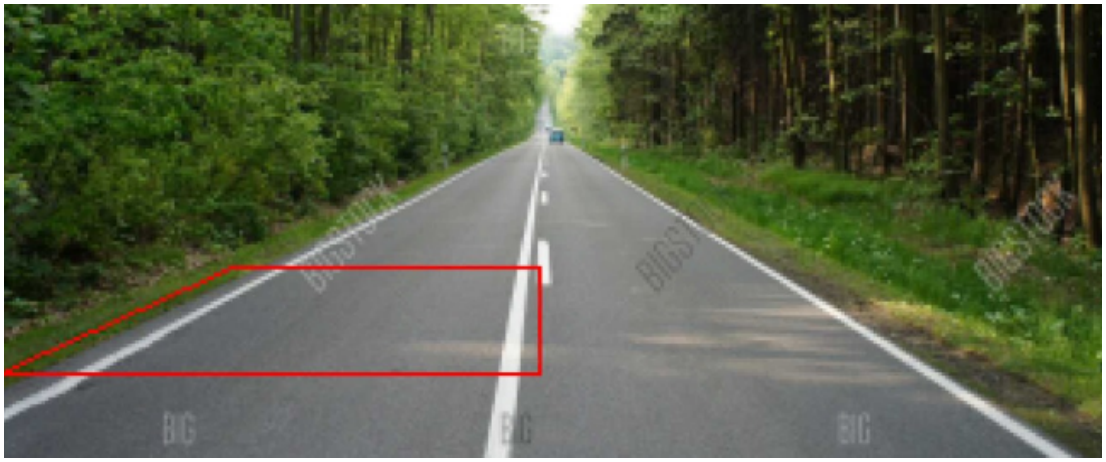


Figure 3.10: Region of Interest [21]

```
points1= [(0,180),(65,150),(150,150),(150,180)]
pointsdes1 = [(60,240),(60,0),(300,0),(300,240)]

points = np.float32(points1)
pointsdes = np.float32( pointsdes1)

histogramlane=np.zeros(320)
ROIlane=np. array([])

cv. line(imag,points1[0],points1[1],(0,0,255),1)
cv. line(imag,points1[1],points1[2],(0,0,255),1)
cv. line(imag,points1[2],points1[3],(0,0,255),1)
cv. line(imag,points1[3],points1[0],(0,0,255),1)
```

Perspective Transformation

Region of interest is in a parallelogram shape. For getting bird eye view or simply called top view ,"cv.getPerspectiveTransform()" is used.

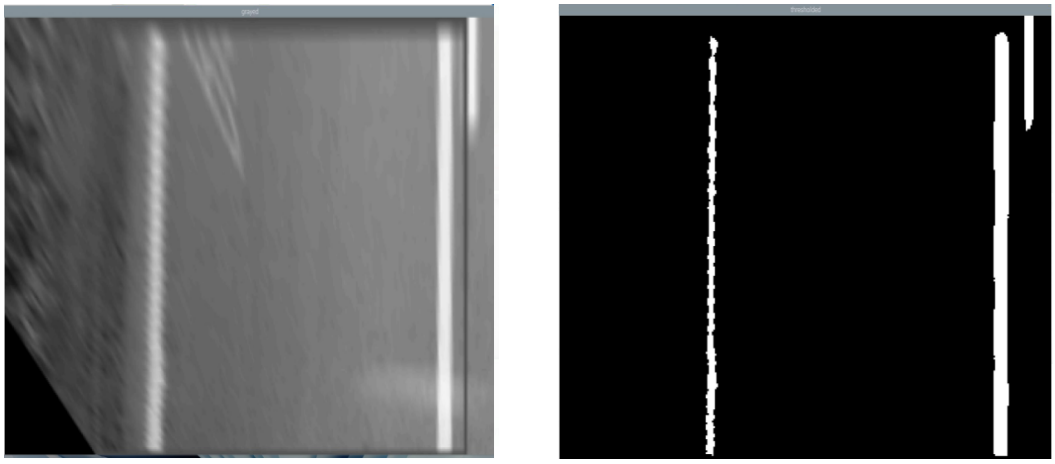
Figure 3.11: Transformed image

```
matrix = cv.getPerspectiveTransform(points,pointsdes)
result = cv.warpPerspective(imag, matrix, (320,240))

cv. namedWindow("perspective", cv.WINDOW_KEEPRATIO)
cv.resizeWindow("perspective", 640, 480)
cv.imshow("perspective", result)
```

Thresholding

white color of the lane should be identified clearly. So that persecuted image is converted into a grey scale.

Gray scaled image is converted into a black and white image using thresholding. Pixels having a value greater than 180 is taken as white pixel, and less than 180 are taken as black pixel. This is done using "cv.threshold()" function.



(a) Gray Scaled image (b) Binary Thresholded Image

Figure 3.12: Thresholded image

```
result = cv. cvtColor(result, cv. COLOR_BGR2GRAY)
cv. namedWindow("grayed", cv.WINDOW_KEEPRATIO)
cv. resizeWindow("grayed", 640, 480)
cv. imshow("grayed", result)

ret,thresholded = cv.threshold(result,180,255,cv. THRESH_BINARY)
cv. namedWindow("thresholded", cv.WINDOW_KEEPRATIO)
cv.resizeWindow("thresholded", 640, 480)
cv.imshow("thresholded", thresholded)
```

Lane detection

For finding the two lane lines, the summation of normalized pixel values is done column-wise. The columns which have got maximum values taken as lane lines.

center of the lane lines plotted using x-coordinates of two lane lines. Also, the center of the window is also plotted

```
thresholded = cv. cvtColor(thresholded, cv. COLOR_GRAY2BGR)
histogramlane. resize(320)
histogramlane[:] = 0
for i in range(320):
    ROIlane=thresholded[140:240,i]
```

```
    ROIlane=ROIlane/255
    histogramlane[i]=np.sum(ROIlane)
print(histogramlane)
leftpos=np.argmax(histogramlane[0:130])
print("leftpo=",leftpos)
rightpos=np.argmax(histogramlane[320:170:-1])
print("rightpos=",rightpos)
cv. line(thresholded,(leftpos,0),(leftpos,240),(0,255,0),2)
cv.line(thresholded,(320-rightpos,0),(320-rightpos,240),(0,255,0),2)
lanecenter=(320-rightpos+leftpos)//2
print("lanecenter",lanecenter)
cv.line(thresholded,(lanecenter,0),(lanecenter,240),(0,0,255),2)
framecenter=160
cv.line(thresholded,(framecenter,0),(framecenter,240),(255,0,0),2)
cv.namedWindow("thresholded1", cv. WINDOW_KEEPRATIO)
cv.resizeWindow("thresholded1", 640, 480)
cv.imshow("thresholded1", thresholded)
result=lanecenter-framecenter
```

3.2.4 Object detection

Object detection is identifying traffic signs,obstacles, vehicles, etc. Large amounts of sample images are collected and trained with help of the TensorFlow library of python. It needs to be tested with an accuracy of greater than 98 percent.

The image from Raspicam is fed to the deep learning model. The deep learning model should be able to identify traffic signs, obstacles, and other vehicles accurately and in less time.

3.2.5 Haarcascading Object Detection

The object detection is primarily implemented using the Haarcascade feature in Opencv. In this process the algorithm is given a set of ”positive” images consisting of the desired objects and a set of ”negative” images. Subsequently a model is trained

using line and edge features until the error threshold is very low. The trained model is obtained as an xml file. The python code for implementation of ”STOP” detection is given below.

```
from PIL import Image
from picamera. array import PiRGBArray
from picamera import PiCamera
import cv2
import numpy as np
import time

CameraResolution = (640,480)
CameraFrameRate = 60
DisplayResolution = (1280,720)
Capture_Image = None
print("Initializing Camera. . . !!")
camera = PiCamera(resolution = CameraResolution)
camera.contrast = 0
camera.framerate = CameraFrameRate
rawCapture = PiRGBArray(camera,size = CameraResolution)
       # allow the camera to warmup
time. sleep(1)
print("Done. . . !!")

STOP = cv2.CascadeClassifier('data/stop. xml')
for frame in camera.capture_continuous(rawCapture, format="bgr",
   use_video_port=True):
          #grab the raw NumPy array representing the image, then
              initialize the timestamp # and occupied/unoccupied text
          Capture_Image = frame. array
          # Source image
          gray = cv2. cvtColor(Capture_Image, cv2.COLOR_BGR2GRAY)
          cv2.namedWindow("Footage", cv2.WINDOW_KEEPRATIO)
          cv2.resizeWindow("Footage",640,480)
```

```
cv2.imshow("Footage",Capture_Image)

# image, reject levels level weights.

stop = STOP.detectMultiScale(gray,
    scaleFactor=1.05,minSize=(24, 24),
    flags=cv2.CASCADE_SCALE_IMAGE)
if (len(stop)):
    cv2.rectangle(Capture_Image, (stop[0][0], stop[0][1]),
        (stop[0][0] + stop[0][2], stop[0][1] + stop[0][3]),
        (255, 255, 0), 2)
    Capture_Image =
        cv2.putText(Capture_Image,"stop",(0,50),cv2.
        FONT_HERSHEY_SIMPLEX,1,(255,0,255),2, cv2.LINE_AA)
    cv2.imshow("Footage",Capture_Image)
cv2.waitKey(1)
rawCapture.truncate(0)
```

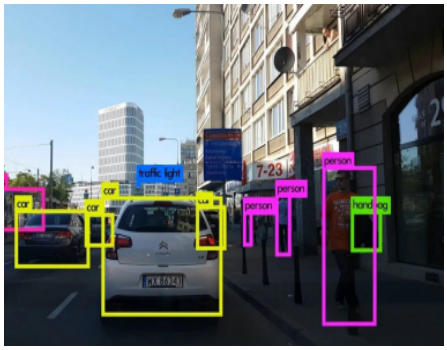


Figure 3.13: Object detection [22]

3.3 Navigation

By analyzing the image for lane detection and object detection, obtained results should be used for controlling vehicular motion.

- Lane is deviating to right - wheel should be rotated to the right.

- Lane is deviating to left -wheel should be rotated to left.

- Stop or go slow sign

- Speed should be decreased.

PWM is used for controlling speed. H-bridge characteristics of the motor driver are used for rotating wheels in a clockwise and anti-clockwise direction

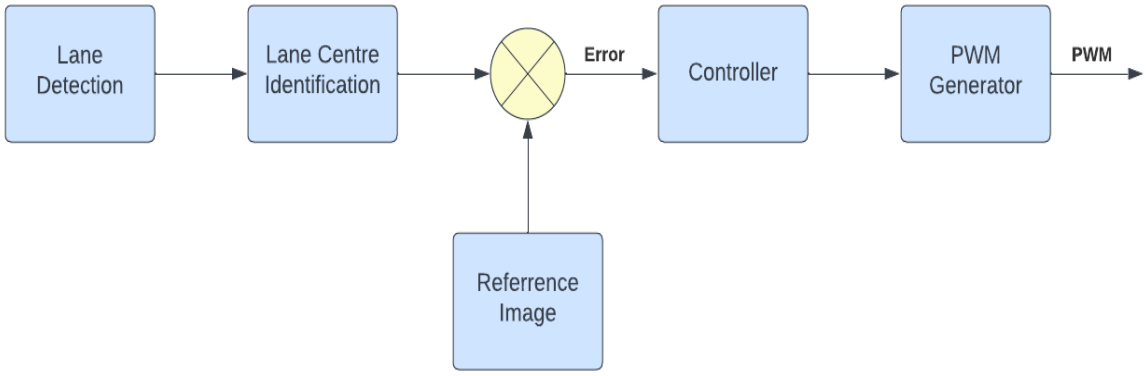


Figure 3.14: Navigation Algorithm

3.3.1 Pulse Width Modulation

By dividing an electrical signal into discrete pieces, pulse-width modulation (PWM) or pulse-duration modulation (PDM) is a technique for lowering the average power produced by an electrical signal. By rapidly flipping the switch between the supply and the load on and off, the average amount of voltage (and current) provided to the load is managed. The total power provided to the load increases while the switch is on for a longer period of time compared to when it is off.

pulse width modulation enabled for GPIO pins of Raspberry Pi following method

A low duty cycle equates to low power because the electricity is off for most of the time; the word duty cycle reflects the ratio of ”on” time to the regular interval or ”period” of time. The duty cycle is measured in percentages, with 100 percent representing the total on. A digital signal has a duty cycle of 50 percent and looks like a ”square” wave when it is on for 50 percent of the time and off for the other 50 percent.

Duty Cycle

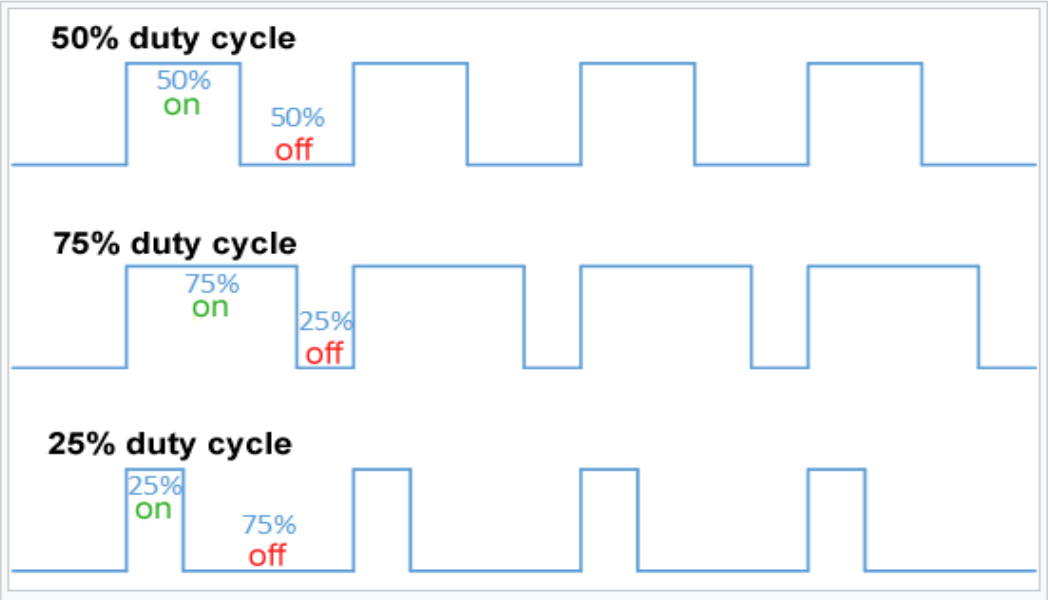


Figure 3.15: Duty Cycle [23]

When a digital signal spends more time in the on the state than the off state, it has a duty cycle of ¿50 percent. When a digital signal spends more time in the off state than the on state, it has a duty cycle of ¡50 percent. Here is a pictorial that illustrates these three scenarios:

3.3.2 H-bridge

H bridge is used to supply power to a two terminal device. By proper arrangement of the switches, the polarity of the power to the device can be changed.

The direction of rotation of a DC motor can be altered by switching the polarity of the power supply. Apart from changing the rotation direction, the H-bridge can provide additional operation mode, ”brake” and ”free run until frictional stop”. The H-bridge configuration is typically used to switch the polarity or direction of the motor, but it may also be used to ’brake,’ which is when the motor suddenly stops because its terminals have been shorted.

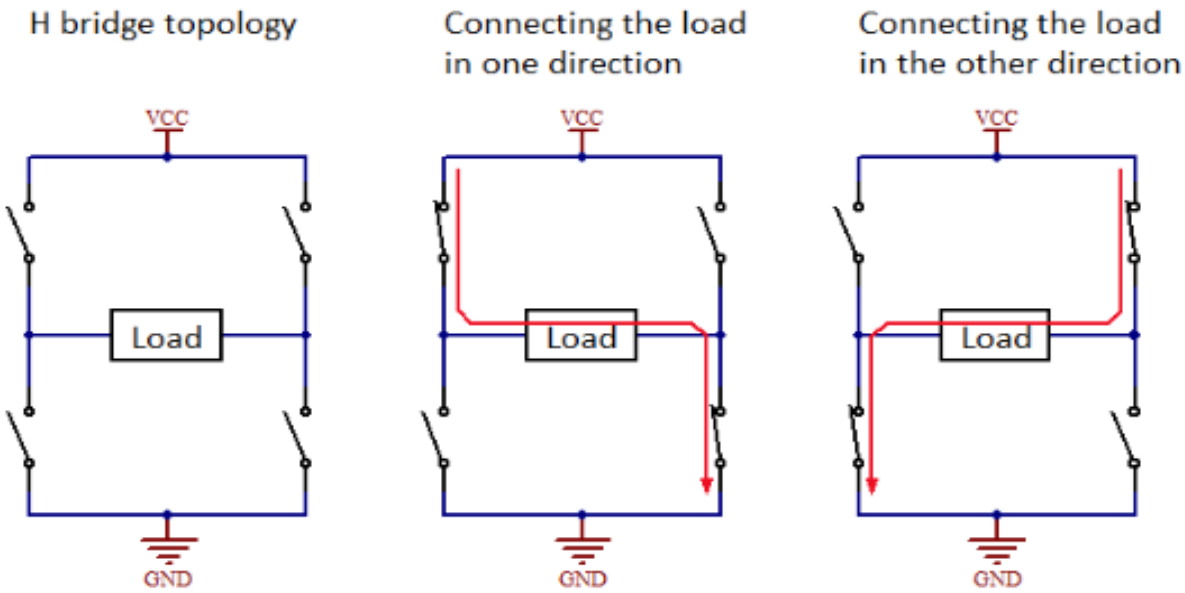


Figure 3.16: The two basic states of an H bridge [24]

The following table summarizes operation, with S1-S4 corresponding to the diagram above. In the table below, ”1” is used to represent ”on” state of the switch, ”0” to represent the ”off” state. Using H bridge and PWM characterstics , the Left and Right functions for motors written

S1	S2	S3	S4	Result
1	0	0	**1**	Motor moves right
0	**1**	**1**	0	Motor moves left
0	0	0	0	Motor coasts
1	0	0	0	
0	**1**	0	0	
0	0	**1**	0	
0	0	0	**1**	
0	**1**	0	**1**	Motor brakes
1	0	**1**	0	

Figure 3.17: H bridge State Table [25]

3.3.3 Graphical User Interface

The Graphical User Interface (GUI) provides a medium for manual control of the vehicle. The basic design is created using Tkinter library in Python.

```
import tkinter as tk

class Motor_driver:

    def gui_init(self):

        gui = tk.Tk()
        top_frame = tk.Frame(gui)
        top_frame.pack(side = tk.TOP)
        middle_frame = tk.Frame(gui)
        middle_frame.pack()
        bottom_frame = tk. Frame(gui)
        bottom_frame. pack(side = tk.BOTTOM)
        f = tk.Button(top_frame,text = "start",width = 10,command =
            self.forward)
        f.pack()
        l = tk.Button(middle_frame,text = "left",width = 10,command =
            self.left)
        l.pack(side = tk.LEFT)
        s = tk. Button(middle_frame,text = "stop",width = 10,command =
            self. stop)
        s. pack(side = tk.LEFT)
        r = tk.Button(middle_frame,text = "right",width = 10,command =
            self.right)
        r.pack(side = tk. LEFT)
        b = tk.Button(bottom_frame,text = "reverse",width = 10,command
            = self.reverse)
        b.pack()
        gui.mainloop()
```

3.4 Vehicle Interface

4 single shaft DC motors are mounted on the chassis, and 2 L298N motor drivers are used for controlling 4 motors. The signals from Raspberry pie are taken from GPIO pins and fed to 2 motor drivers. The H bridge circuit of the motor driver enables clockwise and anti clock wise rotation. Pulse width modulation is used for controlling the speed characteristics.

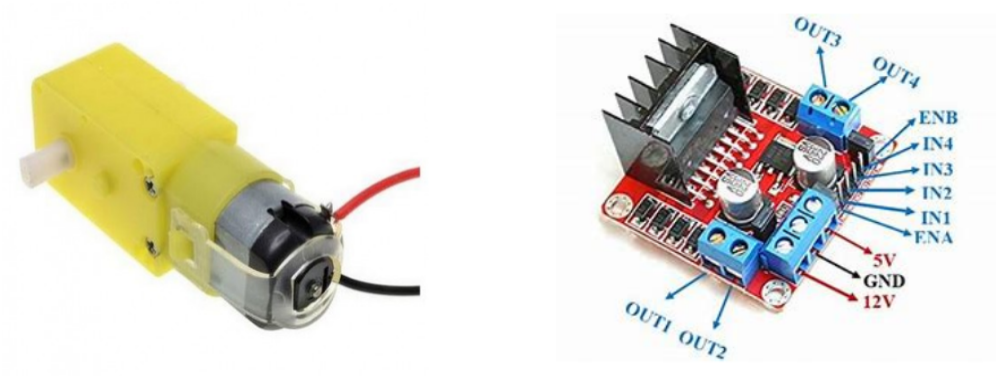
3.4.1 Single shaft DC motor

Single shaft DC motor(5V)

A DC motor is an electrical device that transforms electrical energy into mechanical energy. The electrical energy used as an input source in a DC motor is converted into mechanical rotation. When the DC motor's field coil is powered, a magnetic field forms in the air gap. The direction of the magnetic field that is produced is in the radii of the armature. The North pole side of the field coil is where the magnetic field enters the armature, and the South pole side is where the magnetic field "exits" the armature.

Working principle of DC motor

When kept in a magnetic field, a current-carrying conductor gains torque and develops a tendency to move. In short, when electric fields and magnetic fields interact, a mechanical force arises. This is the principle on which the DC motors work.



(a) DC motor [26] (b) L298N motor driver [27]

Figure 3.18: Motor and Driver

3.4.2 L298N Motor Driver

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.78M05 Voltage regulator will be enabled only when the jumper is placed. ENA and 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.

3.5 Circuit diagram of Hardware interfacing

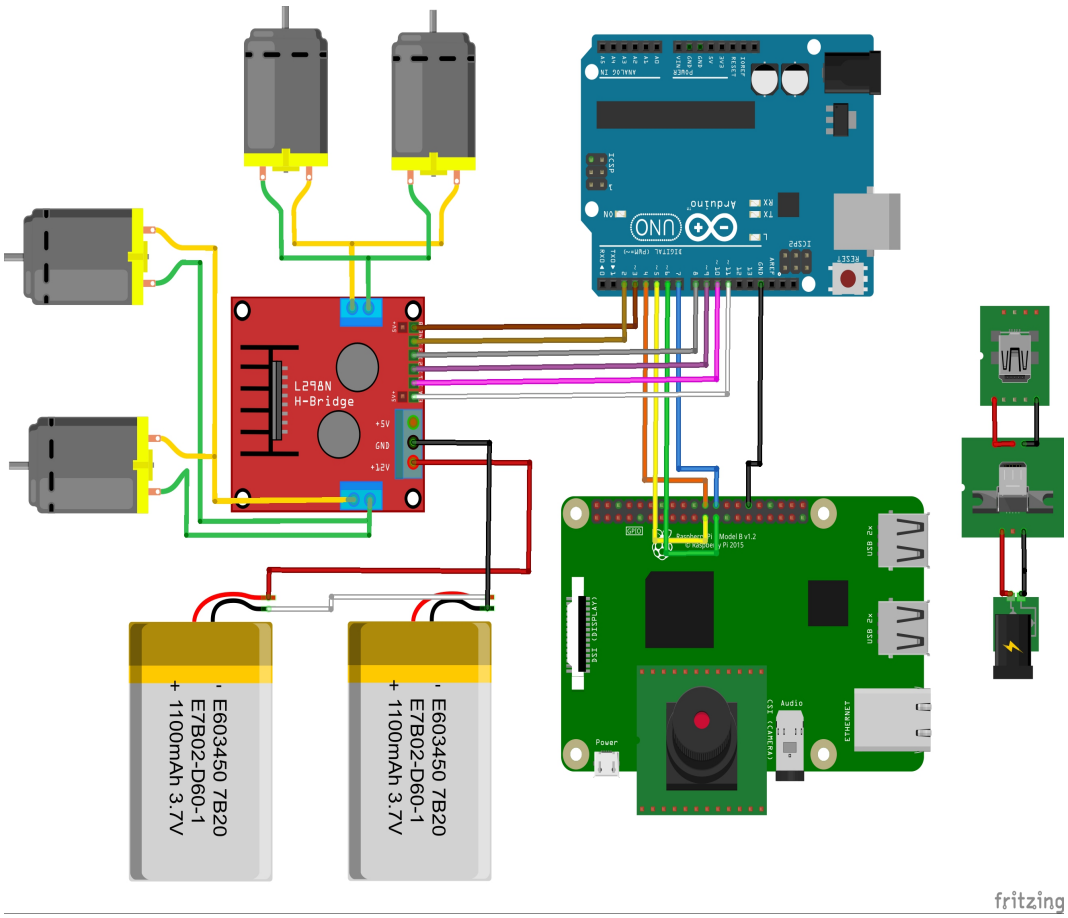


Figure 3.19: Hardware Interfacing

3.6 Arduino Code

```
const int EnableL = 5;
const int HighL = 6;   // LEFT SIDE MOTOR
const int LowL =7;

const int EnableR = 10;
const int HighR = 9;   //RIGHT SIDE MOTOR
const int LowR =8;

const int D0 = 12;    //Raspberry pin 21 LSB
const int D1 = 13;    //Raspberry pin 22
const int D2 = 3;     //Raspberry pin 23
const int D3 = 4;    //Raspberry pin 24 MSB
int a,b,c,d,data;

void setup() {
Serial.begin(9600);
pinMode(EnableL, OUTPUT);
pinMode(HighL, OUTPUT);
pinMode(LowL, OUTPUT);

pinMode(EnableR, OUTPUT);
pinMode(HighR, OUTPUT);
pinMode(LowR, OUTPUT);

pinMode(D0, INPUT_PULLUP);
pinMode(D1, INPUT_PULLUP);
pinMode(D2, INPUT_PULLUP);
pinMode(D3, INPUT_PULLUP);
}
void Data()
{
   a = digitalRead(D0);
```

```
   b = digitalRead(D1);
   c = digitalRead(D2);
   d = digitalRead(D3);
   data = 8*d+4*c+2*b+a;
}
void Forward()
{
  digitalWrite(HighL, HIGH);
  digitalWrite(LowL, LOW);
  analogWrite(EnableL,100);

  digitalWrite(HighR, HIGH);
  digitalWrite(LowR, LOW);
  analogWrite(EnableR,100);
}
void Backward()
{
  digitalWrite(HighL, HIGH);
  digitalWrite(LowL, LOW);
  analogWrite(EnableL,100);

  digitalWrite(HighR, HIGH);
  digitalWrite(LowR, LOW);
  analogWrite(EnableR,100);
}
void Stop()
{
  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL,0);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,0);
```

```
}
void Left1()
{
  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL,100);

  digitalWrite(HighR, HIGH);
  digitalWrite(LowR, LOW);
  analogWrite(EnableR,100);
}
void Left2()
{
  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL,125);

  digitalWrite(HighR, HIGH);
  digitalWrite(LowR, LOW);
  analogWrite(EnableR,125);
}
void Left3()
{
  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL,125);

  digitalWrite(HighR, HIGH);
  digitalWrite(LowR, LOW);
  analogWrite(EnableR,175);
}
void Right1()
{
  digitalWrite(HighL, HIGH);
```

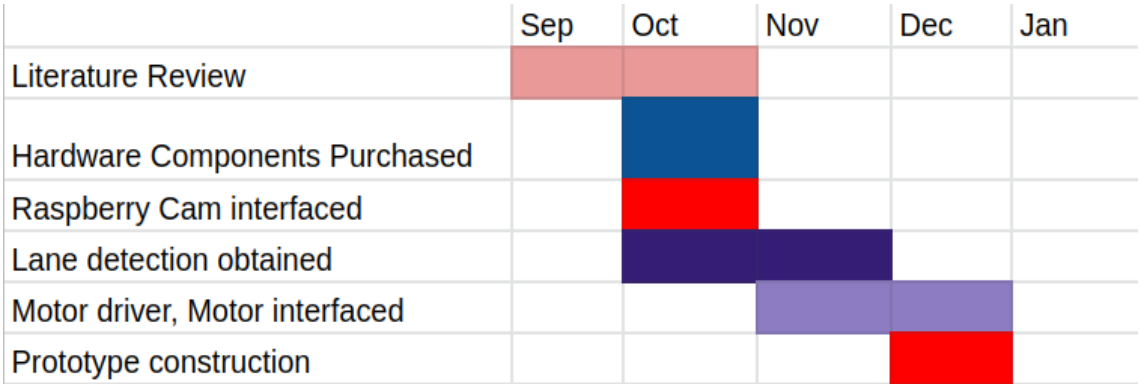
```
  digitalWrite(LowL, LOW);
  analogWrite(EnableL,100);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,100);
}
void Right2()
{
  digitalWrite(HighL, HIGH);
  digitalWrite(LowL, LOW);
  analogWrite(EnableL,125);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,125);
}
void Right3()
{
  digitalWrite(HighL, HIGH);
  digitalWrite(LowL, LOW);
  analogWrite(EnableL,175);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,175);
}
void loop()
{
  Data();
  Serial. println(data);
  if(data==0)
     Forward();
```

```
  else if(data==1)
     Right1();

  else if(data==2)
     Right2();

  else if(data==3)
     Right3();

  else if(data==4)
     Left1();

  else if(data==5)
     Left2();

  else if(data==6)
     Left3();

  else if (data>6)
     Stop();
}
```

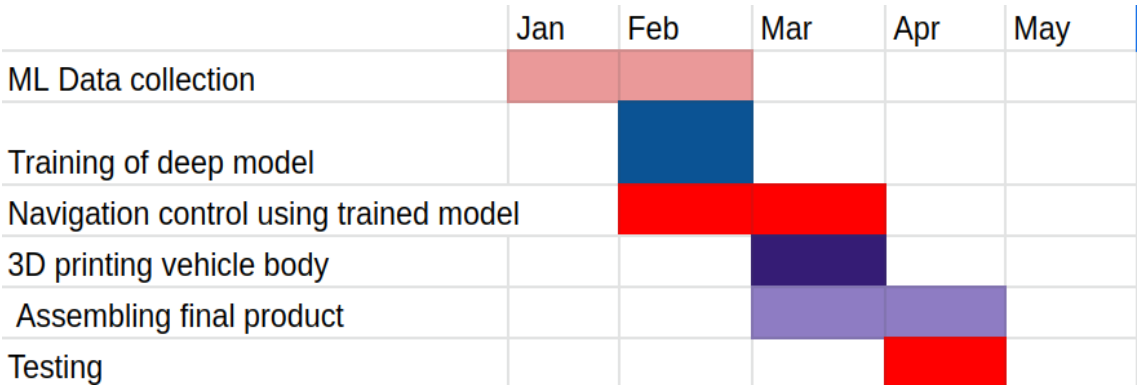
Chapter 4

Work Plan

4.1 Phase 1



4.2 Phase 2



Chapter 5

Final Results

5.1 Autonomous Vehicle

This is the prototype of an Autonomous vehicle that has got features such as

(i) Automatic lane detection

(ii) Automatic lane correction

(iii) Object detection

(iv) Sign board detection

(v) traffic light identification.

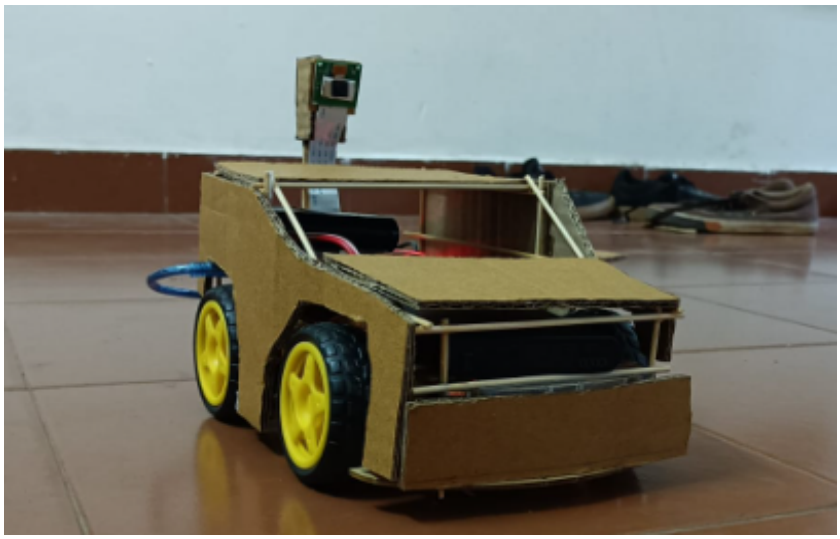


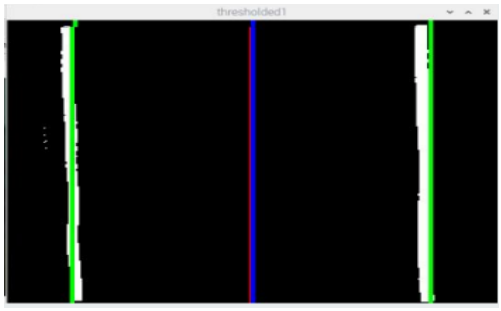
Figure 5.1: Atonomous Vehicle

5.2 Automatic Lane Detection

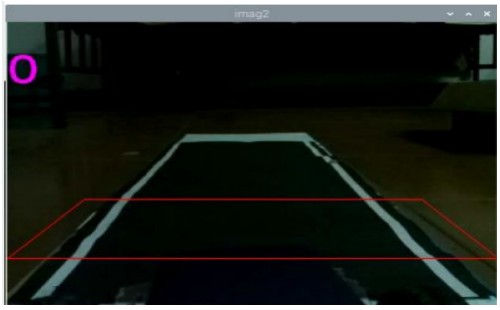
Lane detection is done on real-time captured images.

5.2.1 Zero lane error

The scenario where lane error is zero or the vehicle is aligned to the direction of the road.



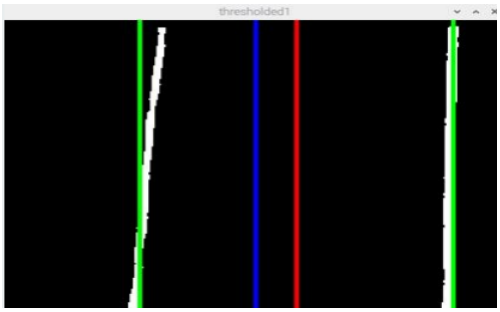
(a) Thresholded bird view



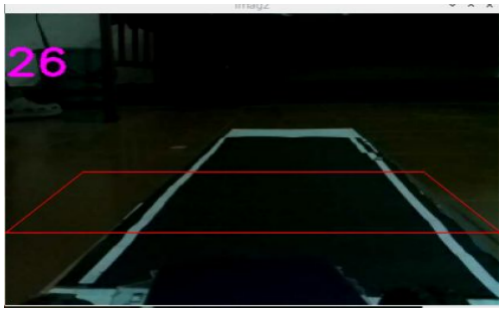
(b) Real time camera view

5.2.2 Positive lane error

The scenario where lane error is positive or the vehicle deviates left to the direction of the road. So, the car needs to be turned right.



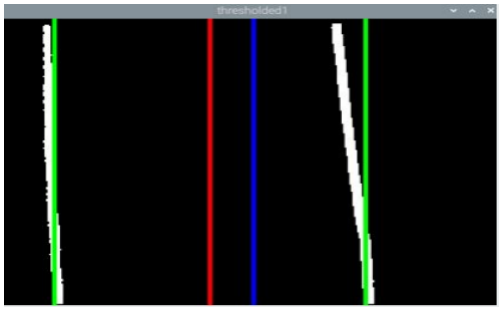
(a) Thresholded bird view



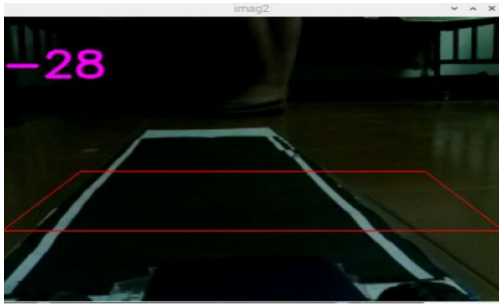
(b) Real time camera view

5.2.3 Negative lane error

The scenario where lane error is positive or the vehicle deviates right to the direction of the road. So, the car needs to be turned left.



(a) Thresholded bird view



(b) Real time camera view

5.3 Automatic Lane Correction

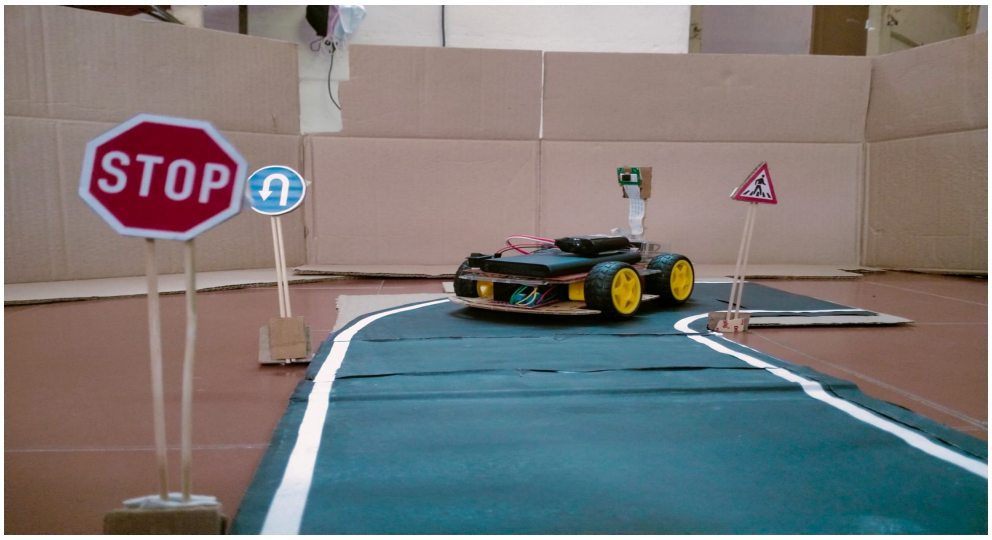


Figure 5.5: vehicle

In the above figure, the vehicle is in a particular scenario. The log window of the vehicle is also given below.Initially, car is at rest and the lane error is -16.After the ”Forward” command is given, the vehicle determines the it needs to turn left to correct the lane error.

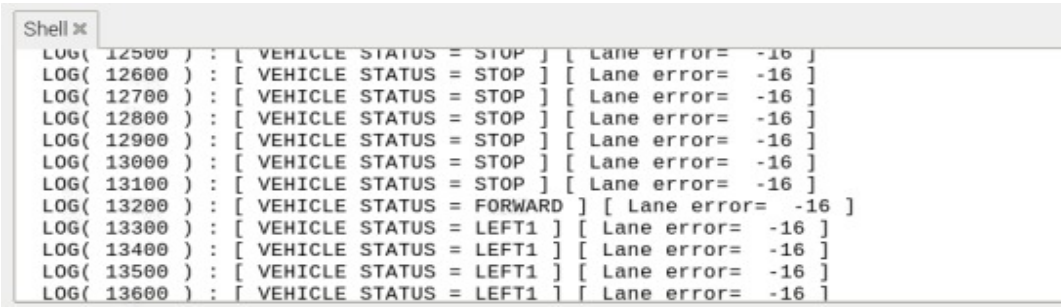


Figure 5.6: logwindow

5.4 Sign Board Detection

5.4.1 Stop sign detection

The vehicle detects the Stop sign board which falls in a range of width and height. So, only detects stop sign boards at a particular distance. Her height and width are taken to be 50 pixels. Vehicle commands to stop after stop-sign detected.

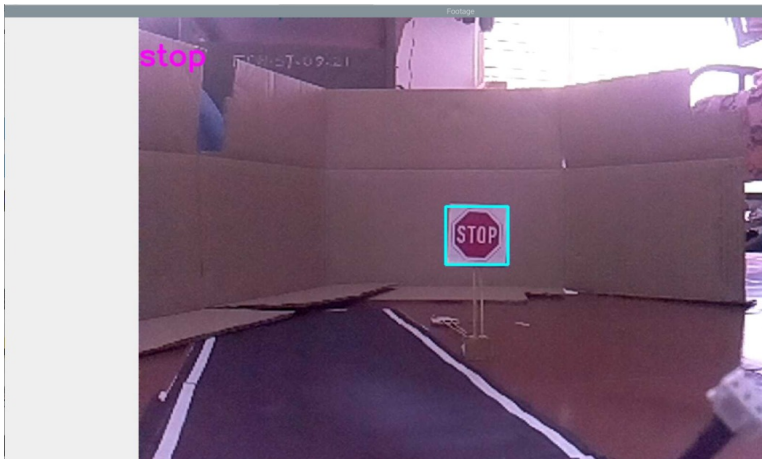


Figure 5.7: Stop sign Detection

5.4.2 Uturn sign detection

The vehicle detects the Uturn sign board which falls in a range of width and height. So, only detects Uturn sign boards at a particular distance. Her height and width are taken to be 50 pixels. Vehicle commands to uturn after Uturn-sign detected.

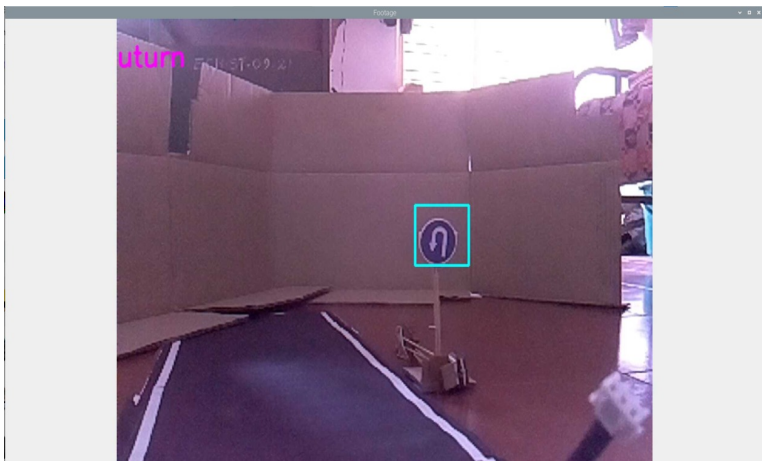


Figure 5.8: Uturn Detection

5.4.3 Zebra sign detection

The vehicle detects the Zebra sign board which falls in a range of width and height. So, only detects Zebra crossing sign boards at a particular distance. Her height and width are taken to be 50 pixels. Vehicle commands to stop fora little after Zebra-sign detected.

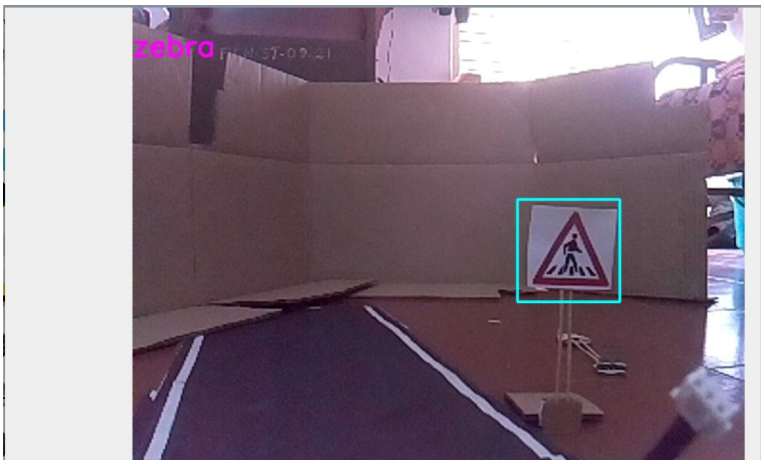


Figure 5.9: Zebra Detection

5.5 Object Detection

If a person walks on the road is unidentified then there is a high chance of accidents. To prevent this human identification is included to stop the vehicle.

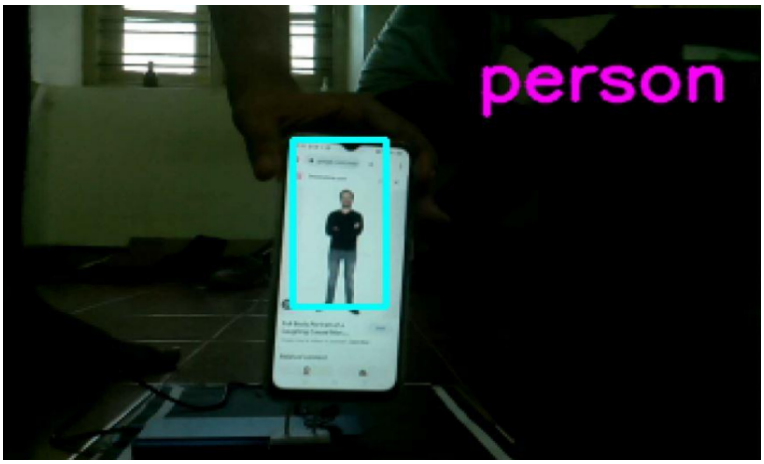


Figure 5.10: Zebra Detection

5.6 Traffic Light Detection

Another feature of our Autonomous vehicle is traffic light detection. When the green light is detected vehicle starts and when the red light is detected vehicle commands to stop.

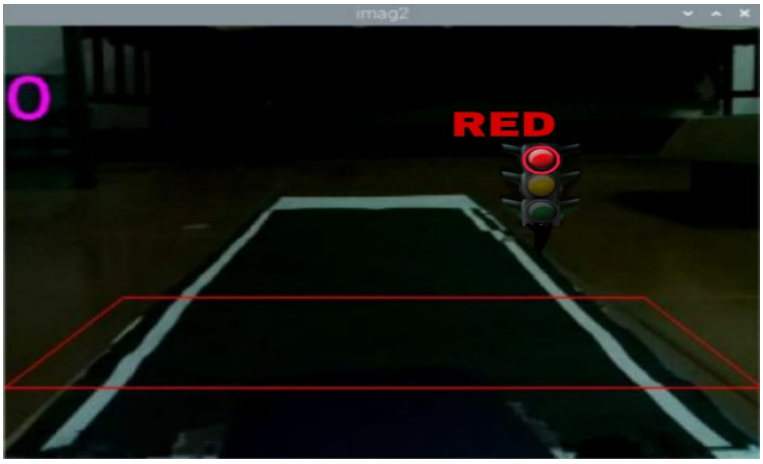


Figure 5.11: Red Sign Detection

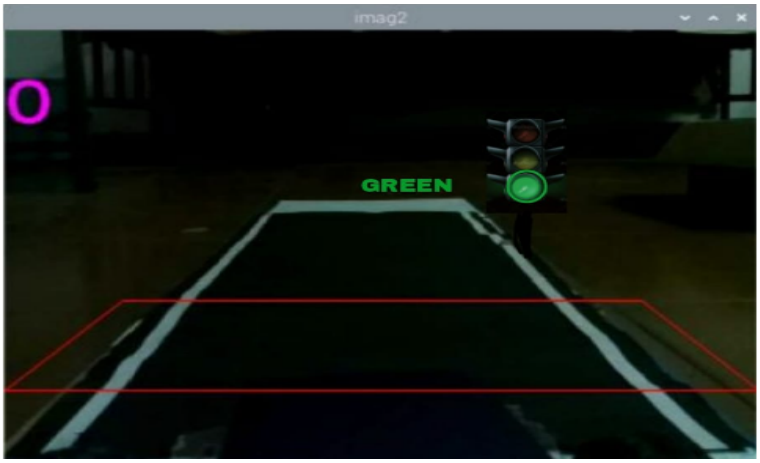


Figure 5.12: Green Sign Detection

5.7 Remote Controlling

Python tkinder library is used to create virtual joystick,Wifi connectivity is used to remotely communicating .

Figure 5.13: Remote controle

5.8 Latency reduction

1. Lane detection ,sign detection and motor control is performed on different CPU threads using parallel processing.

2. Latency can be minimised using distant ROI.

3. Code has been optimised using OOPS.

4. Arduino is used as slave device for controlling the motor. There by reducing the workload on Raspberry Pi.

5. To increase the speed of communication between Pi and Arduino, digital pins are used instead of UART serial communication.

Chapter 6

Conclusion

In this project,we have demonstrated a self driving car capable of automatic lane adjustment and realtime object detection. The driving conditions for this project are very specific and the current model fails to perform in a real life scenario. The visuals taken are entirely from the front view and the rear view of the vehicle is omitted for reducing computational complexity. The future scope of this project aims at improvising vehicle design,and introducing IoT for remote access control. The efficiency of the object detection can be improved by choosing better ML models and larger data sets. The current model is incapable of working under dark conditions. This can be resolved by introducing Infrared Night Vision cameras. The binary thresholding algorithm fails under high intensity lighting. This can be eliminated by an inbuilt illumination which can provide ambient lighting in all conditions. This project doesn't employ a fully autonomous vehicle as the driving range is limited to a few metres. A standard GPS tracking system produces an error of a few metres, hence GPS based navigation is impractical in this setup. The project primarily focuses on developing a vehicle which uses only visuals to sense the environment. The environment sensing can further be improved by using other proximity sensors. However humans almost entirely depend on vision for driving. This can cause human error and it is the basis of almost all road accidents. There arises the significance of autonomous vehicles. Employing Intra communication between the vehicles further reduces risk of accidents and provide a safe travel environment for everyone.

References

[1] Society of Automotive Engineers *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*,SAE J3016-202104

[2] Tarun Tiwari, Satyam Agarwal, Aakarsh Etar. , *Controller Design for Autonomous Vehicle*,2021 International Conference on Advances in Electrical, Computing,Communication and Sustainable Technologies

[3] Boris A. Alpatov, Pavel V. Babayan, Maksim D. Ershov. , *Embedded Image Processing and Video Analysis in Intelligent Camera-based Vision System*,2020 9th MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO), 8-11 JUNE 2020, BUDVA, MONTENEGRO

[4] Chunlei Yu[1], Tuopu Wen[1], Long Chen[1,2], Kun Jiang[1]. , *Common bird-view Transformation for Robust Lane Detection*,Proceedings of 9th IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems July 29- August 2, 2019, Suzhou, China.

[5] Ayushi Sharma, Jyotsna Pathak, Muskan Prakash, J N Singh. , *Object Detection using OpenCV and Python*,2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)

[6] Viola, P. , Jones, M. (n. d.). , *Rapid object detection using a boosted cascade of simple features.* , Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.

[7] (Image Credits) https://www.baeldung.com/cs/convert-rgb-to-grayscale

[8] (Image Credits) https://www.baeldung.com/cs/convert-rgb-to-grayscale

[9] (Image Credits) https://www.analyticsvidhya.com/blog/2020/05/tutorial-real-time-lane-detection-opencv/

[10] (Image Credits) Ayushi Sharma,Jyotsna Pathak,Muskan Prakash,J N Singh., *Object Detection using OpenCV and Python*,2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)

[11] (Image Credits) Hajer Omrane., *Neural controller of autonomous driving mobile robot by an embedded camera*

[12] (Image Credits) https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/

[13] (Image Credits) https://www.youtube.com/watch?v=yvfI4p6Wyvk

[14] (Image Credits) https://bgr.com/lifestyle/tiktok-video-popular-stop-sign-viral/

[15] (Image Credits) https://www.istockphoto.com/vector/pedestrian-crossing-road-sign-vector-illustration-gm1346521181-424266469

[16] (Image Credits) https://www.vecteezy.com/free-vector/u-turn-sign

[17] (Image Credits) https://www.researchgate.net/figure/Raspberry-Pi-3-specifications-fig3-328865076

[18] (Image Credits) https://www.silverlineelectronics.in/pages/what-is-a-raspberry-pi

[19] (Image Credits) https://www.amazon.in/Raspberry-Camera-Cable-Extension-Lengthening/dp/B082P3K1P4

[20] (Image Credits) https://pyimagesearch.com/free-opencv-computer-vision-deep-learning-crash-course/

[21] (Image Credits) https://www.bigstockphoto.com

[22] (Image Credits) https://www.telusinternational.com/insights/ai-data/article/an-introduction-to-5-types-of-image-annotation

[23] (Image Credits) https://electronics360.globalspec.com/article/14808/how-to-control-a-dc-motor-with-an-arduino

[24] (Image Credits) https://embetronicx.com/tutorials/tech-devices/l293d-motor-driver-working/attachment/h-bridge-working/

[25] (Image Credits) https://www.powerelectronictips.com/driving-brushed-and-brushless-dc-motors/

[26] (Image Credits) https://odseven.com/products/odseven-dc-gearbox-motor-tt-motor-200rpm-3-to-6vdc-wholesale

[27] (Image Credits) https://www.hackster.io/lakshyajhalani56/l298n-motor-driver-arduino-motors-motor-driver-l298n-0f2cfd