CONTROL OF AUTONOMOUS VEHICLE THROUGH IMAGE PROCESSING

A mini -project report submitted in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Mechanical Engineering

by

K. NAGARAJU **N150981**

S. SHYAM KUMAR **N150611**

CH. SURESH **N150348**

G. UDAY **N150007**

Under the guidance of

Dr. J. Srinivasa Rao

DEPARTMENT OF MECHANICAL ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES NUZIVIDU

NUZIVIDU – 521202, KRISHNA Dist.,

ANDHRA PRADESH

November 2019

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES NUZIVIDU
Nuzividu – 521202, Krishna Dist., Andhra Pradesh

DEPARTMENT OF MECHANICAL ENGINEERING

APPROVAL OF THE VIVA-VOCE BOARD

Date: 19/11/2019

This is to certify that the mini project entitled "**CONTROL OF AUTONOMOUS VEHICLE THROUGH IMAGE PROCESSING"** submitted by **CHANDANAPALLI SURESH (ID NO:N150348), GEDELA UDAY KUMAR (ID NO:N150007), SURADA SHYAM KUMAR (ID NO:N150611)** and **KARUTURI NAGARAJU (ID NO:N150981)** to Rajiv Gandhi University of Knowledge Technologies Nuzividu, in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Mechanical Engineering** has been accepted by the External Examiners and that these students have successfully defended the mini-project in the Viva-Voce examination held today.

External Examiner(s)

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES NUZIVIDU
Nuzividu – 521202, Krishna Dist., Andhra Pradesh

DEPARTMENT OF MECHANICAL ENGINEERING

CERTIFICATE

This is to certify that the mini-project entitled "**CONTROL OF AUTONOMOUS VEHICLE THROUGH IMAGE PROCESSING**" is a bona-fide record of work done by **CHANDANAPALLI SURESH (ID NO: N150348), GEDELA UDAY KUMAR (ID NO: N150007), SURADA SHYAM KUMAR (ID NO: N150611) and KARUTURI NAGARAJU (ID NO: N150981)** under my supervision and I consider it worthy to fulfil the partial requirements for the award of the degree of Bachelor of Technology in Mechanical Engineering.

Project Guide
Dr. J. Srinivasa Rao
Assistant Professor
Department of Mechanical Engineering

Head of the Department
Dr. S. A. Basha
Assistant Professor
Dept. of Mechanical Engineering

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES NUZIVIDU
Nuzividu – 521202, Krishna Dist., Andhra Pradesh

DEPARTMENT OF MECHANICAL ENGINEERING

DECLARATION

We **CHANDANAPALLI SURESH (ID NO: N150348), GEDELA UDAY KUMAR (ID NO: N150007), SURADA SHYAM KUMAR (ID NO: N150611) and KARUTURI NAGARAJU (ID NO: N150981)** hereby declare that the project report entitled "**CONTROL OF AUTONOMOUS VEHICLE THROUGH IMAGE PROCESSING**", done by us and under the supervision of **Dr. J. SRINIVASA RAO**, submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Mechanical Engineering**.

We assert that the statements made and conclusions drawn are an outcome of the mini-project work. We further declare to the best of our knowledge and belief that the report does not contain full/part of any work which has been already submitted for the thesis evaluation in this or any other university.

Sl. No.	Name	ID No.	Signature
1	**CH.SURESH**	**N150348**	
2	**G.UDAY KUMAR**	**N150007**	
3	**S.SHYAM KUMAR**	**N150611**	
4	**K.NAGARAJU**	**N150981**	

ACKNOWLEDGEMENTS

We were grateful to our guide **Dr. J. Srinivasa Rao**, Assistant professor of Mechanical Engineering Department, RGUKT Nuzividu. His attitude. Way of approach and spontaneity aided us to tackle many issues we faced while gearing up for the mini-project.

Finally, we acclaim our friends and family in giving us financial and mental support for availing us to engage in our project from odds. We appreciate each other for everyone's contribution towards the success of the mini-project.

ABSTRACT

Since evolution transportation is a primary concern for humans. From walking to sitting in a rocket and flying off earth, huge milestones achieved in transportation but the major concern is nothing is suitable in comforting a wide range of consumers. Most of the transportation is happening through cars. So now people need a safe journey while having utmost comforts.

So, in order to ease their lives and to reduce number of detrimental incidents to the people, an autonomous system which obeys the pre-planned instructions needs to be developed. In this Autonomous system all the information related to traffic indications, safety precautions are preloaded so there will be minimum or no detrimental incidents.

In the development of autonomous system primarily 3 phases exist. Those are Lane detection, object detection and distance measurement. Work-plan is divided into 3 parts covering these 3 distinguished fields. After finishing these three major tasks simulation can be initiated and prototype can be produced after getting the desired results in the simulation.

TABLE OF CONTENTS

List of Figures

List of Abbreviations

LiDAR	Light Detection and Ranging
CNN	Computer Neural Networks
RADAR	Radio Detection and Ranging
SAE	Society of Automotive Engineers
AVT	Autonomous Vehicle Technology
RGB	Red Green Blue
IDE	Integrated Development Environment
OPENCV	OPEN Computer Vision
GPS	Global Positioning System

Chapter 1 **Introduction**

An autonomous vehicle is a robotic vehicle that is programmed with a set of detailed instructions in order to sense the changes in the driving environment and act according to the need in order to make a safe and comfortable travel without human intervention.

1.1 Reasons for creating an autonomous system

Since, the early stage of human life, transportation is a major concern for the human species. Along with time there were several advancements in the field of transportation. No matter how much advancement occurred in this field, still there are many incidents which made people lose their lives. The primary causes of these unexpected accidents are distracted and drowsy driving, over-speeding, violation of traffic rules, drunken driving, etc. To overcome these constraints, we aim to design an efficient and reliable autonomous car.

According to a report, nearly 40,000 accidents occur annually in the well-developed countries like the US, UK etc, Road accidents lead to lots of unpredictable consequences such as abnormal deaths, perpetual injuries, loss of earnings, etc.,

1.2 Autonomous system capabilities

In our project, the autonomous vehicle will be able to sense the changes in the environment and works on the concept of Convolution Neural Networks [CNN], Image/video processing.

In this project, the vehicle will be able to detect the lanes and obstacles, calculate the distance from the obstacle to the vehicle, and blow the horn in case something comes in midway. It can be able to change lanes according to the driving environment. All the calculations and analysis will be handled by Artificial Intelligence while solving the problems occurring in the Transportation and enhancing from the day-to-day analysis.

Especially this autonomous system is designed, keeping the Indian Roads in a major concern and handling the obstacles on the Indian Roads.

1.3 Organisation of the report

In chapter 1, the autonomous system requirements, capabilities it holds, the environment it needs to work on, the major concerns for having this autonomous system and it's abilities to help the mankind in easing their lives along with organisation of the report has been mentioned.

In chapter 2, all the literature work that we've done on this Autonomous system which is an inclusion of studying and summarising journal papers has been articulated.

In chapter 3, the work plan and objectives of the autonomous system will be articulated. The sections of the work-plan are detailed.

In chapter 4, describing all the information related to the tools required for the simulation of the experiment has completed.

In chapter 5, the simulation without mentioning code which detects lanes and identifies the vehicles path will be discussed.

In chapter 6, the detailed simulation through code by using openCV as a platform will be articulated.

Chapter 7 deals with the analysis of the Autonomous system and its comparison to the available systems and finding out similarities and dissimilarities among them.

Chapter 8 deals with overall conclusion for the experiment, advantages, limitations and its applications are noted.

In this chapter, we've discussed the autonomous system requirements, capabilities it holds, the environment it needs to work on, the major concerns for having this autonomous system and its abilities to help the mankind in easing their lives.

Chapter 2 **Literature Review**

In previous chapter, we've defined the Autonomous system and the necessities for developing an autonomous system and its capabilities. In this chapter, all the literature work that we've done on this Autonomous system which is an inclusion of studying and summarising journal papers has been articulated.

2.1 **Literature citation**

As mentioned earlier, there are several levels exist in the autonomous system, the levels apply to the driving automation features that are engaged in any given instance of on-road operation of an equipped vehicle. As such, although a given vehicle may be equipped with a driving automation system that is capable of delivering multiple driving automation features that perform at different levels. The levels of driving automation are defined by reference to the specific role played by each of the primary actors in performance of the DDT and/or DDT fallback. (SAE International., 2016)

2.1.1 **Levels of automation**

The autonomous vehicle with increasing levels of automation has the increasing ability to manage safety-critical functions such as steering, accelerator controls, and breaking once the driver decides to cede control. Driver becomes an intermittent operator in the vehicle.so the word driver becomes disappeared. The National Highway Traffic Safety Administration describes these technologies using five levels of automation. These levels describe a continuum of vehicle control that ranges from vehicles that do not have any of their control systems automated. Along with this level 2 which is combined function automation and limited self –driving automation (level 3). (Blanco et al., 2015)

2.1.2 **Previous Developments**

Let's talk about the autonomous systems that were developed by the earlier researchers that previously worked on the autonomous systems. They stated that self-driving cars still have a long way to go but driving support technologies are becoming widespread, even in regular cars. The systems that were previously developed helped the drivers to maintain a steady speed, to keep a safe distance from the car in front and to keep the vehicle in the centre of the lane by using adaptive cruise control with lane centering (Gold et al., 2013).

2.1.3 **Terms adopted in autonomous system**

In this autonomous system, one can consider many terms in terms of the levels of the automation involved in the driving and assisting system. The terms that were adopted are self-driving, driverless, automated, autonomous, robocars and also semi-autonomous. Self-driving is the umbrella term that needs to achieved overall. (Roy, 2019)

2.1.4 **Branding for the prototype**

While mass producing an autonomous system for wider use, a manufacturer needs a branding which influences the end user's expectations of the drivers. If there's difference exists between the expectations and actual product developed then it's hard for the users to develop trust among the manufacturers. also, over-trust and neglecting use of the system may lead to adverse effects. so, developing a system which stays within the expectations of the user and becomes beneficial for the users should be the overall aim. So, the naming of the systems should make the driver whether it is actually controlling the system or assisting the driver to minimize the efforts in controlling the vehicle. (Abraham et al., 2017)

2.1.5 **Real-life scenes**

Since its activation on Tesla vehicles in October 2015 and yet, even with so many miles travelled, we know very little about when, where, and how the human supervisors elect to utilize the automation capabilities of the system. A few anecdotal accounts published on social media, often for the purpose of entertainment, and small-scale studies have begun to detail perceptions of and actual system use and behaviour. (Dikmen et al., 2016)

2.1.6 **Testing and inspection**

Autonomous vehicle needs to be driven for thousands of miles or maybe for more than thousands of miles in order to achieve the trust of the testing vendor. There may be fatalities, accidents if the vehicle is released with improper testing. Therefore, at least for fatalities and injuries, test driving alone cannot provide sufficient evidence for demonstrating safety of the autonomous vehicle. (Kalra et al., 2016)

This work takes an objective, data-driven approach toward evaluating functional vigilance in real-world AI-assisted driving by analysing naturalistic driving data in Tesla vehicles that were instrumented for data collection as part of the MIT Autonomous Vehicle Technology Study (MIT-AVT). (Fridman et al., 2017)

2.1.7 **Vigilance and field-testing results**

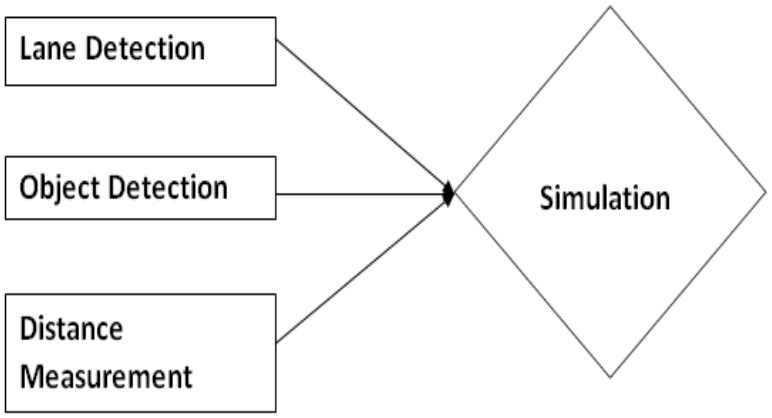
The bulk of the work on vigilance decrement over the past 70 years has been conducted under controlled conditions such that individual variables affecting vigilance could be rigorously studied. These experiments were conducted both in the laboratory and in the field. Applications span a wide range of domains from general visual search tasks to medical diagnosis tasks to agriculture. The majority of results across studies observe that humans make errors on tasks that require prolonged attention. These observations carry over to the task of monitoring and supervising automation, finding that increases in level of automation correlate with an increase in vigilance decrement. (Mackworth et al., 1948)

In this chapter, the literature excerpts from different sources have been mentioned. It is an amalgamation of several factors which influence the autonomous system including safety measures, levels of automation system, testing of the system and the overall advancements as of now. The next chapter in this Article will discuss about the practical objectives and work-plan that is essential for this autonomous system.

Chapter 3 **Objectives and Work Plan**

In previous chapter, the entire information which has been acquired from different sources is strictly evaluated and all the significant issues related to the autonomous system have been mentioned and, in this chapter, the work plan and objectives of the autonomous system will be articulated.

3.1 **Work plan**



3.1.1 **Lane**

In the autono icle needs to identify the switch lanes whenever it i g an image of the road as ar ıs dividers on either sides o y OpenCV in order to identify the lanes and the centre line. The program then converts the RGB image into black and white image through softening. After this process, the program will analyse the image through matplotlib (a python library) through the given coordinates. The region of interest is highlighted through this coordinates and lane detection gets completed.

3.1.2 **Object Detection**

In object detection, the autonomous system needs to identify the objects on the road, traffic signals, diversions etc., (woks needs to be carried out).

3.1.3 **Distance measurement**

This section deals with the measuring of the distance from the vehicle to the surrounding objects. This is possible by using the 2 types of sensors

Chapter 4 **Experimental and Simulation Tools Required**

In the previous chapter we dealt with work-plan of the total experiment. In this chapter, the description of all the information related to the tools required for the simulation of the experiment

4.1 **Experimental Tools**

Raspberry Pi

The foundation seed for development for Raspberry Pi journey started in 2006. Researchers named Eben Upton, rob mullins, jack lang and Alan Mycroft at University of Cambridge's computer laboratory felt low considering the decline in the skill level of level A and the students applying for computer science. The main motive behind the foundation and development of the raspberry pi is to provide affordable and tiny computers to the students in the time when computers are expensive and programming knowledge and support for programming is very less. Eben Upton and his team worked from the year 2006 to 2008 and they produced the final released version and named it as Raspberry Pi. In the year 2008, the processors used in mobile devices became cheaper and powerful and it has the full potential to support and run multimedia processing and programming. Eben and his team saw the high potential of the project and they planned a joint venture with Pete Lomas, MD of norcott technologies (Hardware design and Manufacture Company) and David Braben to make Raspberry Pi foundation.

The raspberry Pi was officially created in year 2012 and the developing stone was laid by Raspberry pi foundation and within 3 years, the model B entered mass level production. Within 2 years of official launch of Pi, 2 million pieces are sold till date.

Upon the launch of Raspberry Pi, because of its portability and affordability it started a new movement of portable and low powerful computers. Taking raspberry pi into consideration various similar boards like intel Galileo, Dwengo, Beaglebone, ORCID etc, have come up with their boards providing similar or little bit more configuration compared to Pi.

Raspberry Pi is a small single board computer developed by Raspberry Pi Foundation, United Kingdom. The board is very portable having extreme computing power and it is capable of supporting amazing projects. It costs from 5 to 35$ based upon model.

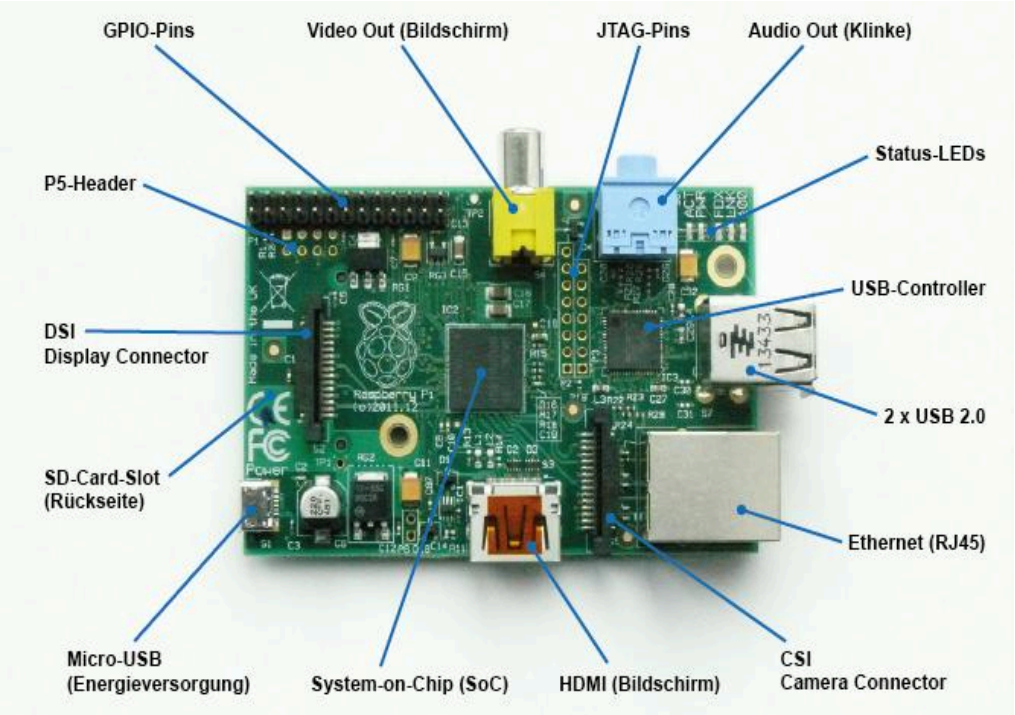


Figure 2. Real Time Raspberry Pi Structure

The raspberry pi board contains of Broadcom based ARM processor, Graphics chip, RAM, GPIO and several ports for supporting external devices. The raspberry pi is operated similar to PC.it requires additional hardware like Keyboard, Mouse, Display Unit, Power Supply, SD Card with OS Installed (Acting like Hard Disk) for operation. Raspberry Pi also facilitates USB ports, Ethernet for Internet. Similar to other computers, raspberry pi also runs based on operating system. The raspberry pi system is supported by linux distribution of operating systems.

Micro-USB power supply

Raspberry Pi requires 5 volts (V) +/- 5% per USB 2.0 standard. Talking of various

models of Raspberry Pi: Model B: 700mA at 3.5 watts (W); Model A: 500mA at 2.5 watts. The power port on Printed Circuit Board (PCB) of Raspberry Pi is Micro-USB type B interface, so a Pi compatible power supply uses standard USB A connector on one side and Micro-USB B connector on other side.

SD card slot

Secure Digital Card slot (SD Card) slot is a solid-state removable storage device which is required to run operating systems on Raspberry Pi as Raspberry Pi doesn't have any onboard memory and data storage functionality. Raspberry Pi supports both SDHC (Secure Digital High Capacity) and SDXC (Secure Digital eXtended Capacity). The best suited card for

proper running of all sorts of operating systems without any disturbances is Class 10 with speed @ 10MB/sec.

USB and Ethernet ports

Raspberry Pi Model B comprise of 2 USB 2.0 ports whereas Model B+ comprise of 4 USB 2.0 ports. USB ports enable the connectivity of external peripherals like Keyboard, Mouse, USB-Hub, Wi-Fi dongle etc. In order to enable Internet connection online and to update the software's or to install latest packages from online repositories, Raspberry Pi supports Ethernet Connection. Raspberry Pi (Every Model) comprise of RJ45 Ethernet Jack which supports CAT5/6 cables.

HDMI (High-definition multi-media interface)

HDMI Port enables Raspberry Pi to be connected to HDTV via HDMI cable. Raspberry Pi supports maximum resolution of 1920x1200. With the help of HDMI Full HD MPEG-4 (Moving Pictures Expert Group 4) can be streamed via HDMI.

Video out (RCA cable)

In addition to HDMI Connectivity which facilitates HD connection, Raspberry Pi also has provision to be connected to standard monitor or TV using RCA video cable. RCA cable is less expensive as compared to HDMI but along with RCA cable, the user has to buy 3.5mm stereo cable for audio facilitation.

GPIO (General Purpose Input Output)

GPIO facilitates connecting all sorts of peripheral devices to Raspberry Pi. Raspberry Pi has onboard GPIO with 40 pins, 26 of which are used as digital inputs or outputs. More importantly, 9 of the 14 new GPIO pins are dedicated inputs/outputs, it also facilitates the onboard UART, I2C, SPI Bus and still large amount of free GPIO pins are there for add-on attachments.

CSI Camera Connector

Raspberry Pi has a camera serial interface type 2(CSi-2). CSI_2 facilitates connection of small camera to Broadcom processor. The primary function of this interface is to standardize the attachment of camera modules to the processors. The CSI-2 version 1.01 supports upto 4 data lanes and each lane carries 1Gbps bandwidth. These specifications enable faster exchange of data.

DSI Display Connector

Raspberry Pi connector S2 is a display serial interface (DSI) for connecting LCD panel using a 15-pin ribbon cable. The Mobile Industry Processor Interface (MIPI) inside the Broadcom BCM2835 IC feeds graphics data directly to the display panel through this connector.

System on Chip (SoC):

Raspberry Pi (System on Chip) SoC is ARM Based by Broadcom Technologies. The ARM (Advanced RISC machine) processor runs from 700 MHz to 1 Ghz. The SoC also facilitates videocore 4 GPU, and is capable for fast 3D core, openGL and supports Blueray and H.264 video playback.

Pros of Raspberry Pi

1. The primary and major advantage of raspberry pi is its portability and affordability. It is small yet powerful and compatible with various projects. It can be used from small to medium level tasks like running as web server, database server and media servers.
2. Raspberry pi can be used to write and compile several programming languages. Raspberry pi mainly uses python as programming language which is less complex and easy to compile and execute
3. Raspberry pi supports open-source operating system making it easy for the developers around the world to develop various platforms based on the linux sources.
4. Raspberry supports addon hardware like Camera, USB devices etc.

Limitations

1. It cannot act as full-pledged computer because the ethernet port and processing CPU is not so fast to handle multitasking computing cycles.
2. It is not compatible with fully functional windows OS.
3. Battery is not present, so continuous power supply is mandatory in order to work.

4.1.2 Utilization of Resources

In order to make the prototype, SD card, GPIO pins, USB Ports, CSI camera connector, ethernet port and micro-USB ports have been used.

SD-Card

It is used to run the operating system of the raspberry pi system. In the system, there is 2 GIGABYTE SD-Card is provided. It runs the operating system as well as stores the compiled programming instructions or simply the files which are created within the system manually.

General Purpose Input Output Pins (GPIO)

These GPIO pins are used to send signals to the Arduino board which is connected to the raspberry pi system with jumper wires and linked to the servo motors for controlling the power needed to the wheels. The entire path depends upon the camera input of the path which

is taken by Camera serial interface. The data taken by the camera sensor will be analysed by the program written in CPP in geany within the raspberry Pi environment. Based upon the data received and analysed by the raspberry pi environment, it decides the amount of power needs to be send to the motors in order to be in the absolute path without any deviation.

USB ports

It has 2 USB 2.0 ports and 2 USB 3.0 Ports. These USB ports can be used to connect peripheral devices like Keyboard, mouse, storage devices etc.

Micro-USB port

It is a type C port. It is used to power the raspberry device through a power bank that delivers power output upto 18W which is a little bit higher than the required power input of 15W to the raspberry pi device.

Camera module

The Raspberry Pi Camera Module v2 is a high quality 8 mega-pixel camera based around the Sony IMX219 image sensor. It is a custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It is connected to the raspberry pi CSI-2 port in order to record the path and send it to the raspberry pi environment to analyse the path and get the exact values according to the test path.

L298N module

The L298N module is a motor controller. The Arduino board which is connected to the raspberry pi, sends signals to the L298N module instructing the amount of power needs to be delivered to each motor in order to commute. This module has input and output pins to connect jumper wires from the controller to the motor.

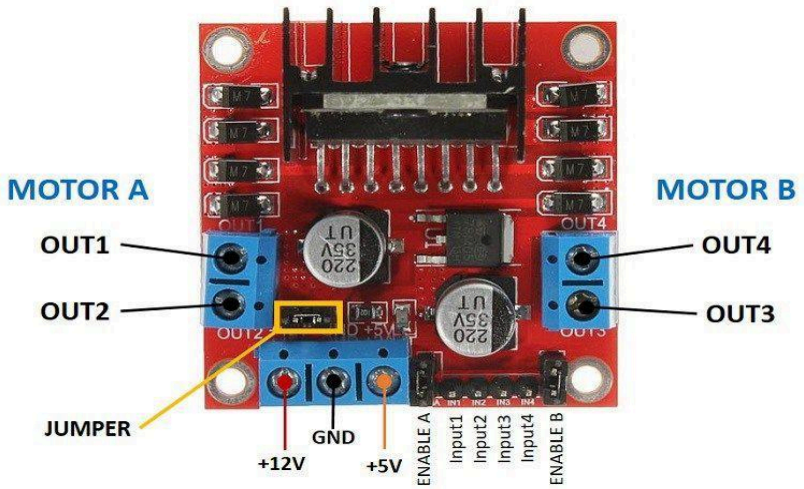


Figure 3. L298N module

4.1.3 **Motors**

Motors can be defined as a machine which is powered by electricity or internal combustion, that supplies motive power for a vehicle or for another device with moving parts. There are 3 types of motors available namely *DC, Stepper, and Servo motors.*

DC Motors

DC motors are electromagnetic devices that use the interaction of magnetic fields and conductors to convert electrical energy to mechanical energy for rotation. There are many types of DC motors out in the market. The brushed and brushless motors are the most common DC motors.

Brushed DC motors

Brushed DC motors are the motors that are used in many household appliances, toys, computer cooling fans etc. These motors are easy to construct and control. These are cost effective, low power driven and mostly preferrable while designing bots in order to conduct experiments.

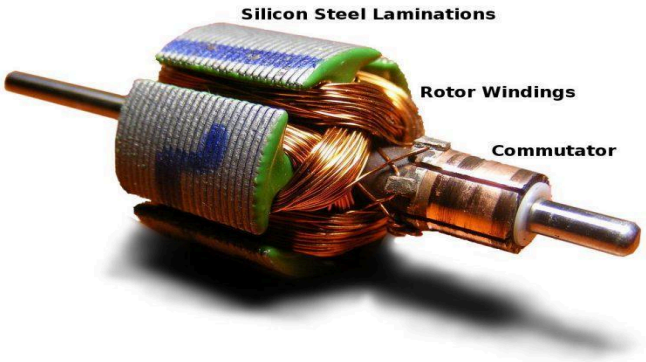


Figure 4. Brushed DC motor Interior

In the brushed motors, the current is provided via two stationary metallic brushes that make contact with the different segments on the ring. As the commutator rotates, the brushes make contact with the next segment and therefore continue the rotation of the motor. As a result, this generates friction and therefore heat and sparks gets generated.

The movement of DC motors can be explained as the DC motors consist of coils connected to segments of a ring, or commutator. The coils are surrounded by a pair of magnets, or a stator, that envelopes the coils in an electric field. When current is passed through a wire in a magnetic field, the wire experiences a force, and so the coils in the motor experience a force that pushes the coil and begins the rotation.

Advantages

1. It is simple to control. In order to control the DC brushed motor, apply a voltage to start driving them. If the voltage is lowered, the motors will be slowed down. And if the voltage is reversed then they will move in reverse direction.
2. High torque is achieved at low speeds.
3. These are 75-80 % efficient in terms of power.
4. These motors cost very less compared to other motors available in the market.

Limitations

1. Aside from the audible noise from the rubbing parts, electromagnetic noise is also generated as a result of the strong sparks that occur at areas where the brushes pass over the gaps in the commutator. This can potentially cause interference in other parts of the system.

It needs constant maintenance as Brushes could get easily worn out as a result of continuous moving contact. Speed could be limited due to brush heating.

4.1.4 Traffic light Equipment

Led lights

A pair of led lights were connected to the printed circuit board. Those are mini red and green led lights.

HW battery

A single hw battery is used to power the pcb.

resistor

A resistor is used to reduce the illumination produced by the bulbs on the pcb.

Along with this few wires and plaster is used to complete the traffic light equipment.

4.2 Simulation tools

4.2.1 Arduino

Arduino is an open -source microcontroller which can be easily programmed, erased and reprogrammed at any instant of time. Based on simple microcontroller boards, it is an open-source computing platform that is used for constructing and programming electronic devices. It is also capable of acting as a mini computer just like other microcontrollers by taking inputs and controlling the outputs for a variety of electronics devices. Arduino uses a hardware known as the Arduino development board and software for developing the code known as the Arduino IDE (Integrated Development Environment). This development board can also be used to burn (upload) a new code to the board by simply using a USB cable to upload. The Arduino IDE provides a simplified integrated platform which can run on regular personal computers and allows users to write programs for Arduino using C or C++.

Types of Arduino Boards

Arduino boards are available with many different types of built-in modules in it. Boards such as Arduino BT come with a built-in Bluetooth module, for wireless communication. These built-in modules can also be available separately which can then be interfaced (mounted) to it. These modules are known as Shields.

Commonly used Arduino Shields

Arduino Ethernet Shield

It that allows an Arduino board to connect to the internet using the Ethernet library and to read and write an SD card using the SD library.

Arduino Wireless shield

It allows your Arduino board to communicate wirelessly using Zigbee (Zigbee is a standard that addresses the need of very low-cost implementation of low power devices with low data rate for short range wireless communications).

Elements of Arduino board

It can be classified into two categories.

1. Hardware
2. Software

Hardware

The Arduino Development Board consists of many components that together makes it work. Here are some of those main component blocks that help in its functioning.

Micro-Controller

This is the heart of the development board, which works as a mini computer and can receive as well as send information or command to the peripheral devices connected to it. The microcontroller used differs from board to board; it also has its own various specifications.

External Power supply

This power supply is used to power the Arduino development board. with a regulated voltage ranging from 9 – 12 volts.

USB Plug

This plug is a very important port in this board. It is used to upload (burn) a program to the microcontroller using a USB cable. It also has a regulated power of 5V which also powers the Arduino board in cases when the External Power Supply is absent.

Reset Button
This button is present on the board and can be used to resets the Arduino microcontroller.

Analog Pins
There are some analog input pins ranging from A0 – A7 (typical). These pins are used for the analog input / output. The no. of analog pins also varies from board to board.

Digital I/O pins
There are some digital input pins also ranging from 2 to 16 (typical). These pins are used for the digital input / output. The no. of these digital pins also varies from board to board.

Power and GND pins
There are pins on the development board that provide 3.3, 5 volts and ground through them.

Software

The program code written for Arduino is known as a sketch. The software used for developing such sketches for an Arduino is commonly known as the Arduino IDE. This IDE contains the following parts in it.

Console Tool Bar
This toolbar contains various buttons like Verify, Upload, New, Open, Save and Serial Monitor. On the bottom right-hand corner of the window there displays the Development Board and the Serial Port in use.

Usage of Arduino in the Project
Arduino is used as a micro controller in order to instruct the L298N module to distribute the desired power input to the respected motors.

4.2.2 Circuit of the System

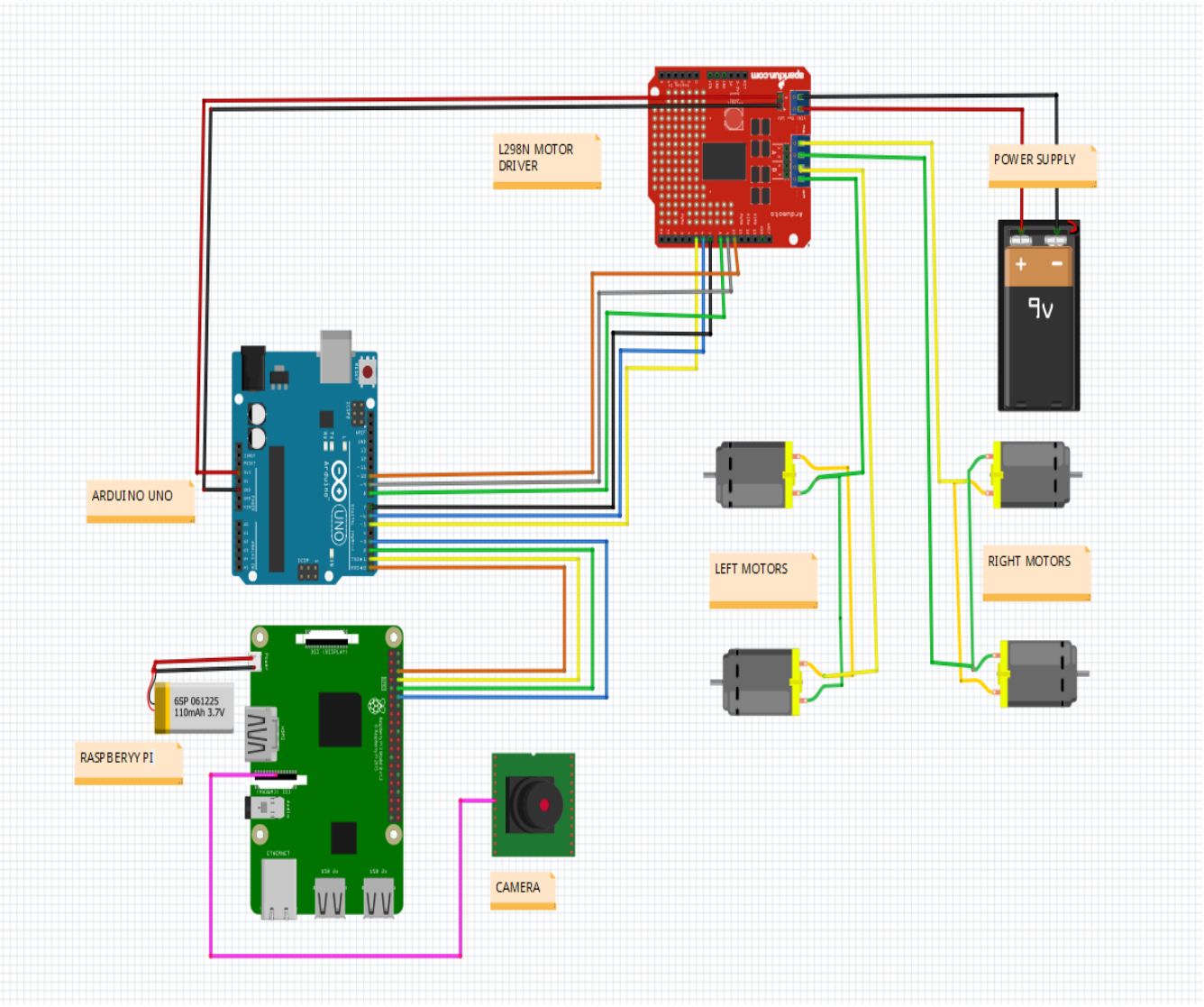


Figure 5. Circuit Depicting the system connections

There are 4 motors connected to the robo chassis. The motors are connected in parallel to the L298N module which controls the power input to the motors. 6 jumper wires are connected from L298N bridge to Arduino Ports (10,9,8,7,6,5). 4 jumper wires are connected from Arduino ports 0,1,2,3 to Raspberry pi physical pins (21,22,23,24). Camera module is connected to raspberry pi with ribbon pins.

Chapter 5 **Simulation of the Lane Detection & Object Detection**

In the previous chapter, the required tools for experimenting and simulation are articulated. The tools are divided into sections. In current chapter, the simulation with code which detects lanes and identifies the vehicles path will be discussed.

5.1 **Simulation**

This picture depicts the standard lane image which contains dividers on either sides of the lane and also the centre line. The image needs to be read by OpenCV in order to identify the lanes and the centre line.

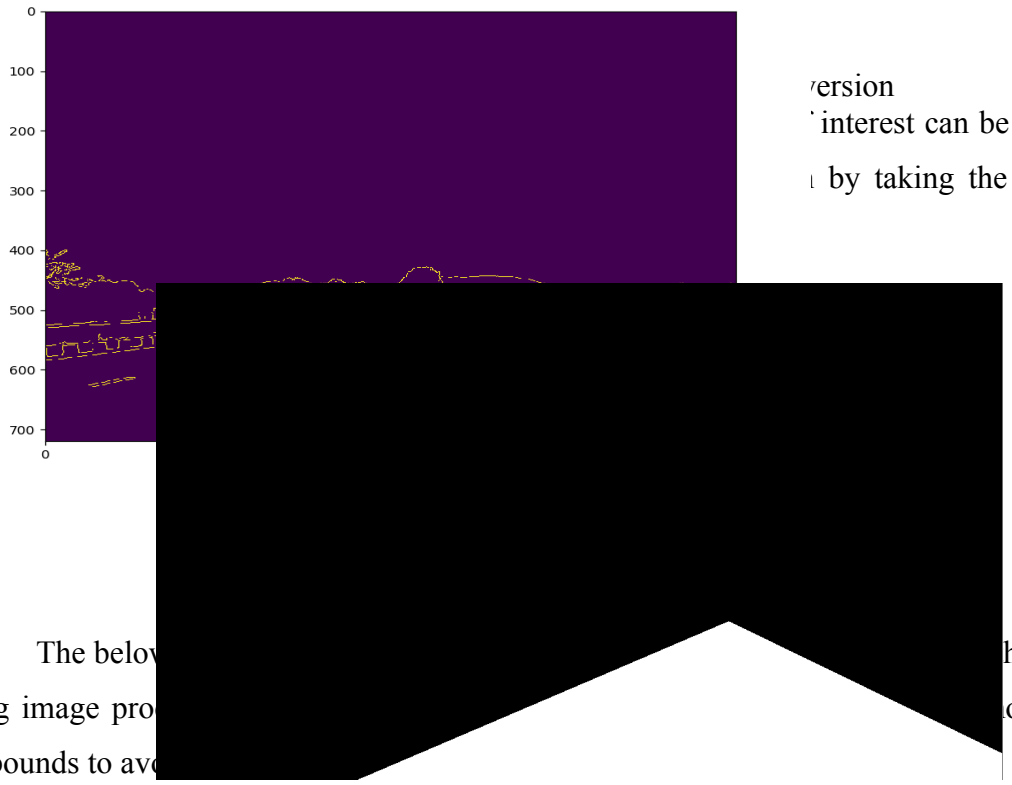
mage
ne RGB image into Black
ar e binary number of white
pi

ge in order
to ore time to
get processed by program to
calculate and to

nto binary numbers. The
po e pixels with binary value
25 xels with binary value 0.
Th he lane and boundaries of
th

g up
is a library available in
py ibrary which is comprised

of all the tools which are required for graphical representation of the lane and also it signifies the coordinate systems.



ersion
interest can be briefed as
th by taking the X and Y
cc

The belov he code by
using image pro ove within
the bounds to av

Figure 12. Final Result

In this chapter, the detection of lanes is briefed as a primary step towards reaching the desired results. Results yielded positively. In the next chapter, the detailed simulation through code by using OpenCV as a platform will be mentioned.

5.2 Continuation

In the previous session, all the simulation is done in python using OpenCV modules. Lane detection is completed using the data samples taken on open roads. Here, for easing the prototype making and to be compatible with Arduino IDE, the programming code is written in CPP language.

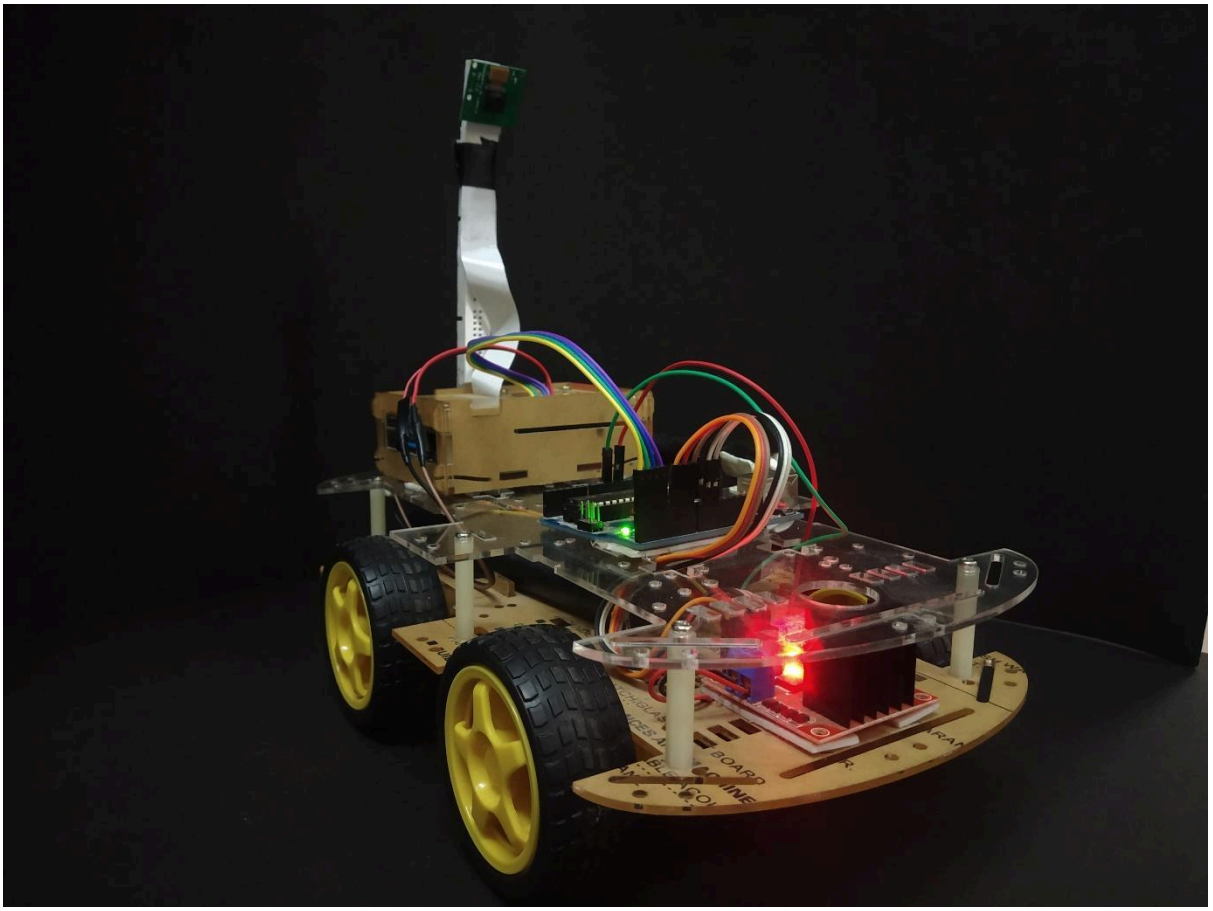


Figure 13. Autonomous Vehicle Prototype

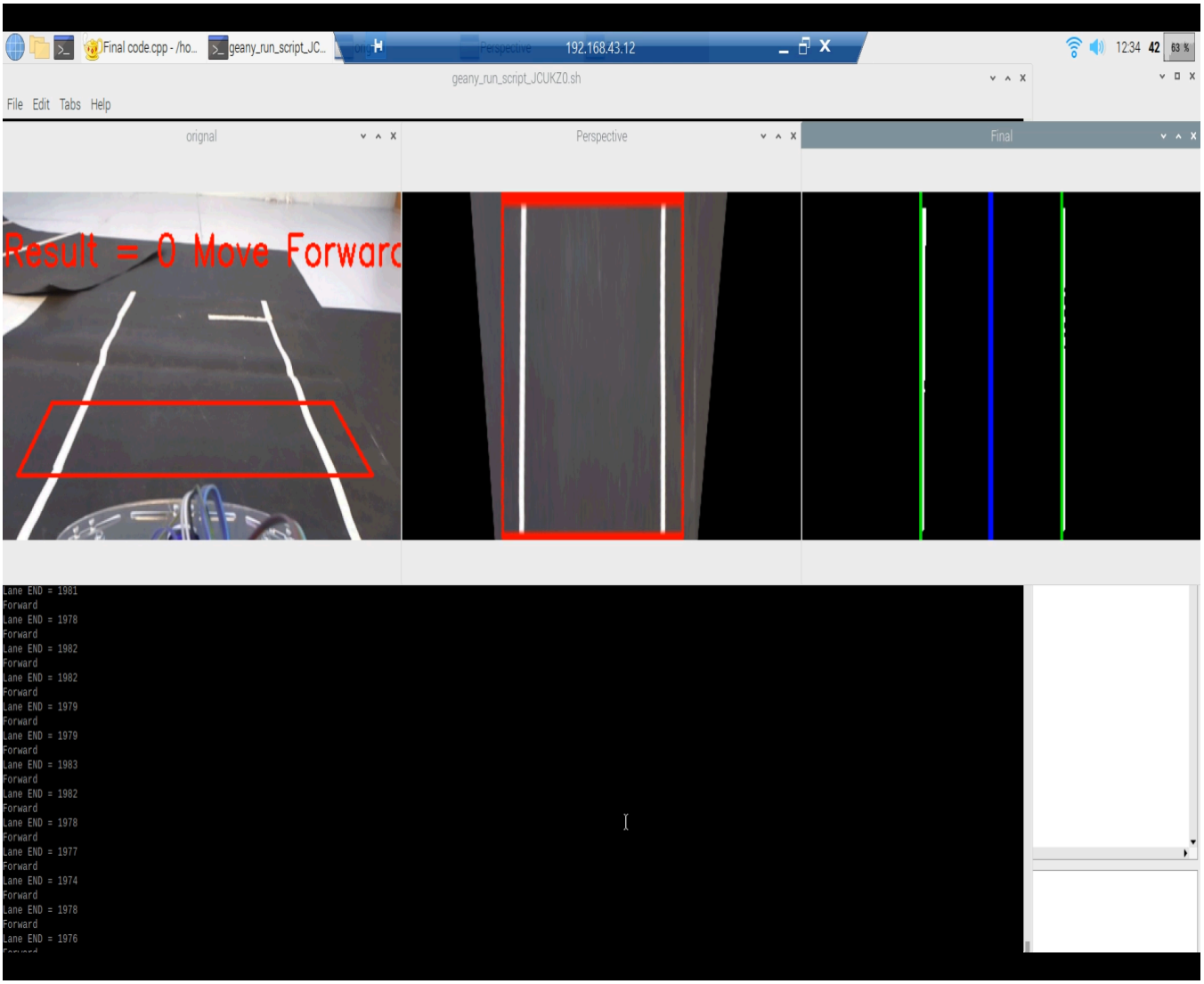


Figure 14. The simulation Windows of the Prototype depicting Region of Interest, Perspective (bird eye) and Grey Scale

The program is written in such a way that the data points value that is obtained from the camera is analysed by the raspberry pi environment. If the data points match with the values given in the program, then the vehicle moves forward.

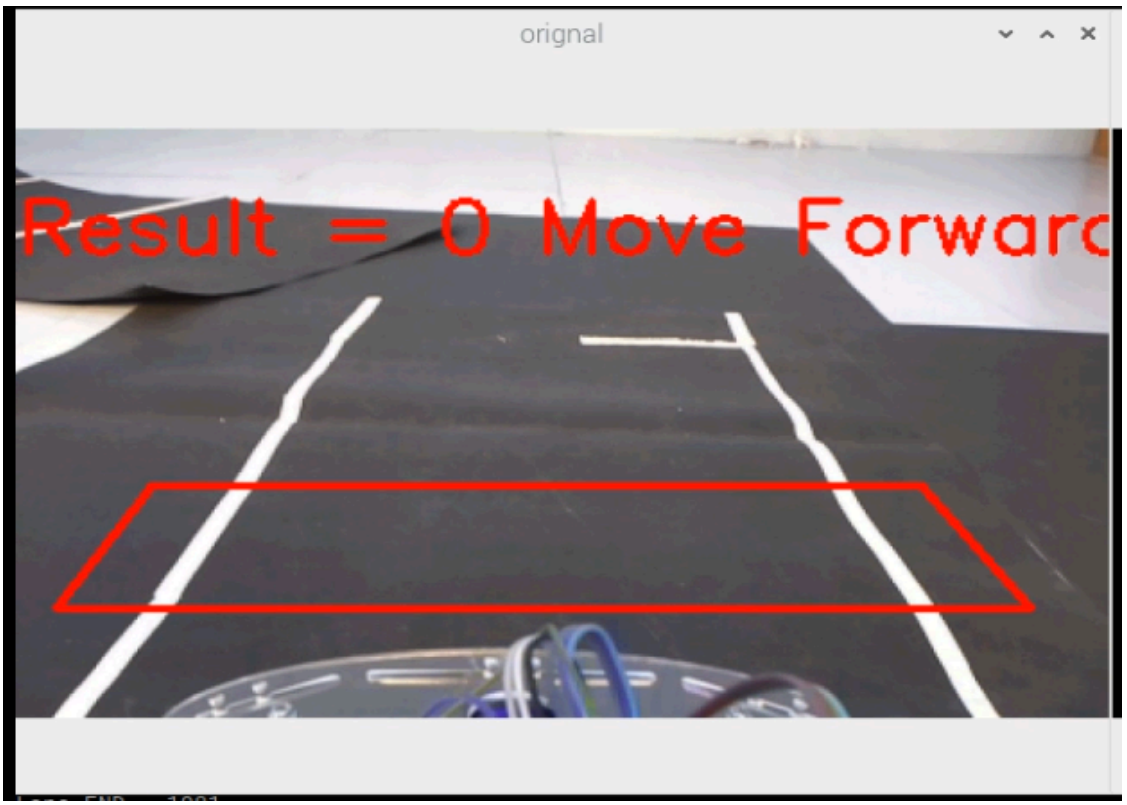


Figure 15. Simulation Window for forward motion

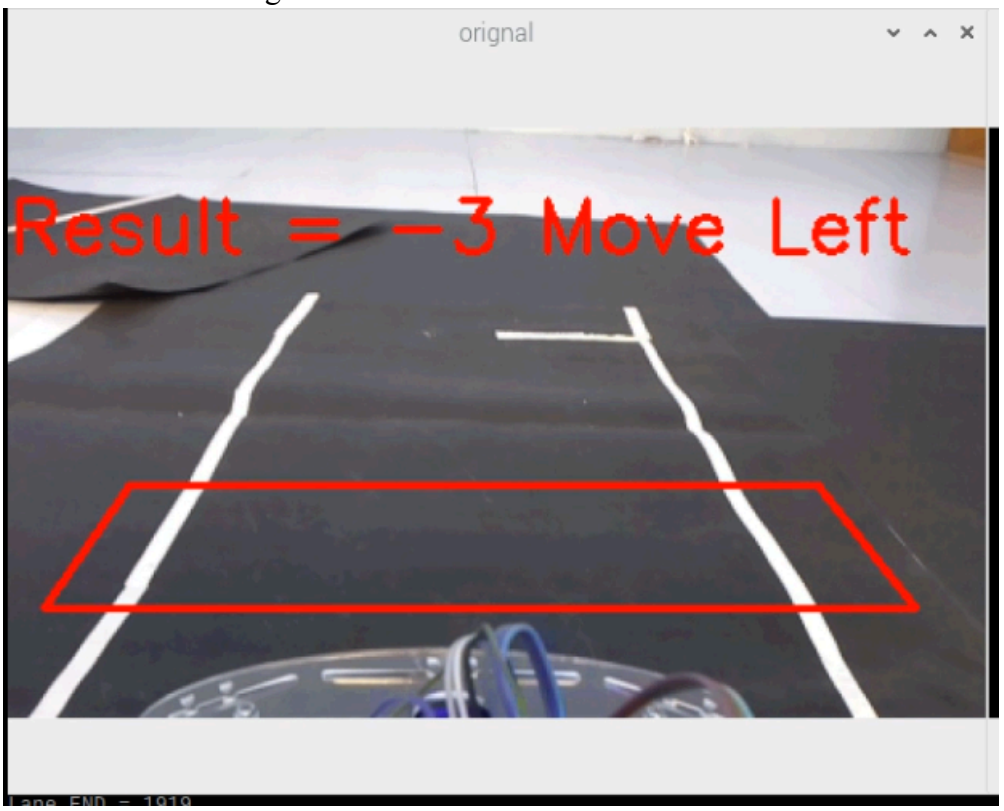


Figure 16. Simulation window for left motion

5.3 Object Detection

The prototype successfully detects the lane in the previous sessions. Now, the system needs to identify the various objects that it crosses while travelling like humans, animals, stop signs[19], traffic signals and a lot more real-world object. In order to make this possible, the system is fed with data samples of various objects in image formats. The system analyses the given data and identifies the object later whenever it came across. Here neural networks play a major role in making this possible. Some key terms that are frequently used were mentioned below.

1. positive samples

Positive samples basically are that different images of the object that the system should detect. The data is fed in different angles & orientations.

2. negative samples

These are the images which does not any image of the object that we want to detect. So basically, for the negative sample we will capture the image of the track and surrounding environment in which the system needs to be tested.

3. Training

The whole process of training the neural network using step 1&2 within the software that develops the data that is used for the motion of the vehicle while identifying the lane as well as surrounding objects.

4. Detection

the entire data [the images of the positive samples along with the code] is uploaded into the autonomous system mainframe. This will make the system work as designed.

5.3.2 Neural networks training

The neural network training is implemented in the system[21] in order to detect the objects as well as to calculate distance between the vehicle and objects. The training needs certain data like positive samples, negative samples and cascade software in order to crop signs of various indications from various orientations [23].

Initially, a stop sign is being used to train the network. Stop sign image is downloaded in various angles. Nearly, 300 images of the positive sign and 700 negative samples were downloaded and uploaded into the cascade software. Those images are then cropped using the tool in order to improve the accuracy of the tool. It makes the system to detect the stop sign from any orientation [22]. Code is written in C++ in order to take the images and display in the graphical user interface. The cascade software is trained using the downloaded images

of the stop sign. This generates an XML file which is later used to run the testing of detection of stop sign on real time images.

Figure 17. Detection of stop sign from a positive sample

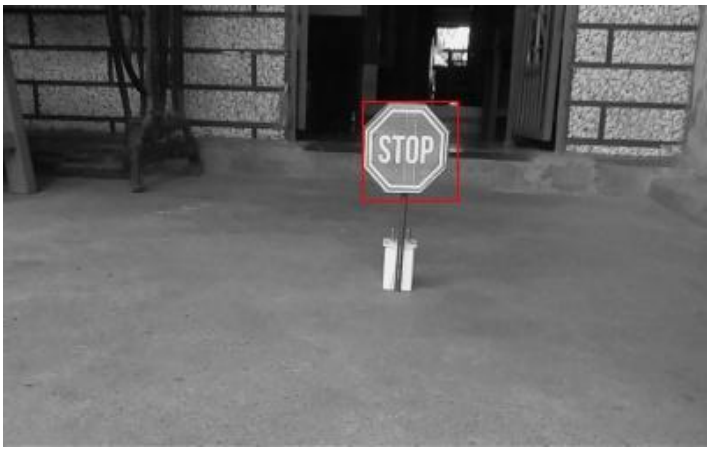


Figure 18. Stop sign sample 2 from an altered distance and angle

Figure 19. Detection sample

The running environment is also fed into the system in similar way as positive samples. The samples are taken and uploaded into the cascade software in the negative values. The cascade software is then trained and creates an XML file which is dumped into the code in order to detect the Realtime environment while the system is engaged.

in this way the objects are identified by using neural networks. One can extend the identification of objects to any extent of objects without limit, the positive sample needs to be taken trained by the cascade software from various orientations. And the generated file needs to be integrated into the main source code so as the raspberry pi system controls the Arduino board upon identifying the object and generating a decision for the vehicle's movement[21].

5.4 Traffic light Detection training

In order to train the prototype to detect traffic signals and initiate appropriate response to the traffic signals, a model of traffic indications was developed. for producing the model, 1 HW battery, 1 PCB, 4 led lights, 2 resistors and wires were used to connect the whole circuit board.

5.4.1 Capturing positive samples

For an autonomous system to work dynamically in Real time scenarios, it needs to fed with lot of data samples and different circumstances of the environment. More number of positive samples makes the training much more effective and accurate. Therefore, a large number of positive samples were fed into the cascade software through which Xml file was generated and used to incorporate into the neural network of the autonomous system. The positive samples were collected while the red light in the traffic model was ON and OFF.

5.4.2 Capturing negative samples

Negative samples were also collected by turning the green light on and switching off the red light. These images were taken by the default camera attached to the prototype. These images were saved in the project directory listing in the folder named n (negative). These negative images help the system to differentiate the object of interest and stationary objects as discussed in the previous chapter while training the stop sign detection.

Figure 20. Traffic indications model with red and green led lights

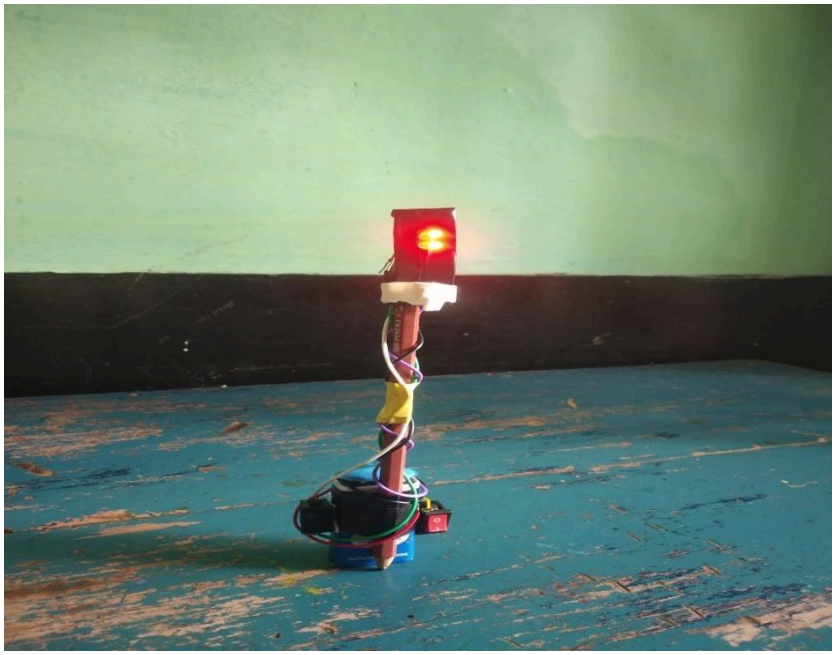


Figure 21. Traffic model with red light switched on for positive sample collection.

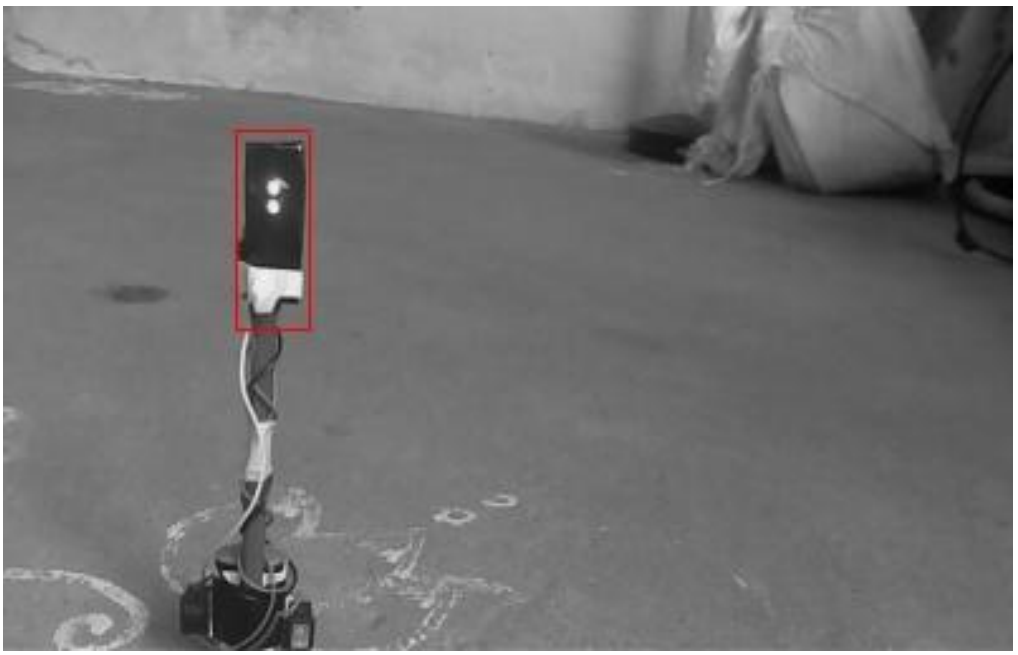


Figure 22. Figure depicting the traffic control equipment in the cascade software while training.

Chapter 6 **Simulation by using Programmed code**

In the previous chapter, the detection of lanes is briefed as a primary step towards reaching the desired results. Results yielded positively. In this chapter, the detailed simulation through code by using openCV as a platform will be discussed.

6.1 **Code supporting the entire Autonomous system**

```
#include <opencv2/opencv.hpp>
#include <raspicam_cv.h>
#include <iostream>
#include <chrono>
#include <ctime>
#include <wiringPi.h>

using namespace std;
using namespace cv;
using namespace raspicam;

// Image Processing variables
Mat frame, Matrix, framePers, frameGray, frameThresh, frameEdge, frameFinal, frameFinalDuplicate,
frameFinalDuplicate1;
Mat ROILane, ROILaneEnd;
int LeftLanePos, RightLanePos, frameCenter, laneCenter, Result, laneEnd;

RaspiCam_Cv Camera;

stringstream ss;


vector<int> histrogramLane;
vector<int> histrogramLaneEnd;

Point2f Source[] = {Point2f(40,135),Point2f(360,135),Point2f(0,185), Point2f(400,185)};
Point2f Destination[] = {Point2f(100,0),Point2f(280,0),Point2f(100,240), Point2f(280,240)};


//Machine Learning variables
CascadeClassifier Stop_Cascade, Object_Cascade, Traffic_Cascade;
Mat frame_Stop, RoI_Stop, gray_Stop, frame_Object, RoI_Object, gray_Object, frame_Traffic,
RoI_Traffic, gray_Traffic;
vector<Rect> Stop, Object, Traffic;
int dist_Stop, dist_Object, dist_Traffic;

 void Setup ( int argc,char **argv, RaspiCam_Cv &Camera )
 {
   Camera.set ( CAP_PROP_FRAME_WIDTH,  ( "-w",argc,argv,400 ) );
   Camera.set ( CAP_PROP_FRAME_HEIGHT,  ( "-h",argc,argv,240 ) );
   Camera.set ( CAP_PROP_BRIGHTNESS, ( "-br",argc,argv,50 ) );
   Camera.set ( CAP_PROP_CONTRAST ,( "-co",argc,argv,50 ) );
```

```
   Camera.set ( CAP_PROP_SATURATION,  ( "-sa",argc,argv,50 ) );
   Camera.set ( CAP_PROP_GAIN,  ( "-g",argc,argv ,50 ) );
   Camera.set ( CAP_PROP_FPS,  ( "-fps",argc,argv,0));

}

void Capture()
{
   Camera.grab();
   Camera.retrieve( frame);
   cvtColor(frame, frame_Stop, COLOR_BGR2RGB);
   cvtColor(frame, frame_Object, COLOR_BGR2RGB);
   cvtColor(frame, frame_Traffic, COLOR_BGR2RGB);
   cvtColor(frame, frame, COLOR_BGR2RGB);

}

void Perspective()
{
        line(frame,Source[0], Source[1], Scalar(0,0,255), 2);
        line(frame,Source[1], Source[3], Scalar(0,0,255), 2);
        line(frame,Source[3], Source[2], Scalar(0,0,255), 2);
        line(frame,Source[2], Source[0], Scalar(0,0,255), 2);


        Matrix = getPerspectiveTransform(Source, Destination);
        warpPerspective(frame, framePers, Matrix, Size(400,240));
}

void Threshold()
{
        cvtColor(framePers, frameGray, COLOR_RGB2GRAY);
        inRange(frameGray, 230, 255, frameThresh);
        Canny(frameGray,frameEdge, 900, 900, 3, false);
        add(frameThresh, frameEdge, frameFinal);
        cvtColor(frameFinal, frameFinal, COLOR_GRAY2RGB);
        cvtColor(frameFinal, frameFinalDuplicate, COLOR_RGB2BGR);   //used in histrogram function
only
        cvtColor(frameFinal, frameFinalDuplicate1, COLOR_RGB2BGR);   //used in histrogram
function only

}

void Histrogram()
{
   histrogramLane.resize(400);
   histrogramLane.clear();

   for(int i=0; i<400; i++)      //frame.size().width = 400
   {
        ROILane = frameFinalDuplicate(Rect(i,140,1,100));
```

```
        divide(255, ROILane, ROILane);
        histogramLane.push_back((int)(sum(ROILane)[0]));
    }

        histogramLaneEnd.resize(400);
      histogramLaneEnd.clear();
        for (int i = 0; i < 400; i++)
        {
                ROILaneEnd = frameFinalDuplicate1(Rect(i, 0, 1, 240));
                divide(255, ROILaneEnd, ROILaneEnd);
                histogramLaneEnd.push_back((int)(sum(ROILaneEnd)[0]));


        }
          laneEnd = sum(histogramLaneEnd)[0];
          cout<<"Lane END = "<<laneEnd<<endl;
}

void LaneFinder()
{
   vector<int>:: iterator LeftPtr;
   LeftPtr = max_element(histogramLane.begin(), histogramLane.begin() + 150);
   LeftLanePos = distance(histogramLane.begin(), LeftPtr);

   vector<int>:: iterator RightPtr;
   RightPtr = max_element(histogramLane.begin() +250, histogramLane.end());
   RightLanePos = distance(histogramLane.begin(), RightPtr);

   line(frameFinal, Point2f(LeftLanePos, 0), Point2f(LeftLanePos, 240), Scalar(0, 255,0), 2);
   line(frameFinal, Point2f(RightLanePos, 0), Point2f(RightLanePos, 240), Scalar(0,255,0), 2);
}

void LaneCenter()
{
   laneCenter = (RightLanePos-LeftLanePos)/2 +LeftLanePos;
   frameCenter = 188;

   line(frameFinal, Point2f(laneCenter,0), Point2f(laneCenter,240), Scalar(0,255,0), 3);
   line(frameFinal, Point2f(frameCenter,0), Point2f(frameCenter,240), Scalar(255,0,0), 3);

   Result = laneCenter-frameCenter;
}


void Stop_detection()
{
   if(!Stop_Cascade.load("//home//pi//Desktop//MACHINE LEARNING//Stop_cascade.xml"))
   {
        printf("Unable to open stop cascade file");
   }
```

```
  RoI_Stop = frame_Stop(Rect(200,0,200,140));
  cvtColor(RoI_Stop, gray_Stop, COLOR_RGB2GRAY);
  equalizeHist(gray_Stop, gray_Stop);
  Stop_Cascade.detectMultiScale(gray_Stop, Stop);

  for(int i=0; i<Stop.size(); i++)
  {
        Point P1(Stop[i].x, Stop[i].y);
        Point P2(Stop[i].x + Stop[i].width, Stop[i].y + Stop[i].height);

        rectangle(RoI_Stop, P1, P2, Scalar(0, 0, 255), 2);
        putText(RoI_Stop, "Stop Sign", P1, FONT_HERSHEY_PLAIN, 1,  Scalar(0, 0, 255, 255), 2);
        dist_Stop = (-1.07)*(P2.x-P1.x) + 102.597;

    ss.str(" ");
    ss.clear();
    ss<<"D = "<<dist_Stop<<"cm";
    putText(RoI_Stop, ss.str(), Point2f(1,130), 0,1, Scalar(0,0,255), 2);

  }

}

void Traffic_detection()
{
  if(!Traffic_Cascade.load("//home//pi//Desktop//MACHINE LEARNING//Trafficc_cascade.xml"))
  {
        printf("Unable to open traffic cascade file");
  }

  RoI_Traffic = frame_Traffic(Rect(200,0,200,140));
  cvtColor(RoI_Traffic, gray_Traffic, COLOR_RGB2GRAY);
  equalizeHist(gray_Traffic, gray_Traffic);
  Traffic_Cascade.detectMultiScale(gray_Traffic, Traffic);

  for(int i=0; i<Traffic.size(); i++)
  {
        Point P1(Traffic[i].x, Traffic[i].y);
        Point P2(Traffic[i].x + Traffic[i].width, Traffic[i].y + Traffic[i].height);

        rectangle(RoI_Traffic, P1, P2, Scalar(0, 0, 255), 2);
        putText(RoI_Traffic, "Traffic Light", P1, FONT_HERSHEY_PLAIN, 1,  Scalar(0, 0, 255, 255), 2);
        dist_Traffic = (-1.07)*(P2.x-P1.x) + 102.597;

    ss.str(" ");
    ss.clear();
    ss<<"D = "<<P2.x-P1.x<<"cm";
    putText(RoI_Traffic, ss.str(), Point2f(1,130), 0,1, Scalar(0,0,255), 2);

  }
```

```
}



void Object_detection()
{
   if(!Object_Cascade.load("//home//pi//Desktop//MACHINE LEARNING//Object_cascade.xml"))
   {
          printf("Unable to open Object cascade file");
   }

   RoI_Object = frame_Object(Rect(100,50,200,190));
   cvtColor(RoI_Object, gray_Object, COLOR_RGB2GRAY);
   equalizeHist(gray_Object, gray_Object);
   Object_Cascade.detectMultiScale(gray_Object, Object);

   for(int i=0; i<Object.size(); i++)
   {
          Point P1(Object[i].x, Object[i].y);
          Point P2(Object[i].x + Object[i].width, Object[i].y + Object[i].height);

          rectangle(RoI_Object, P1, P2, Scalar(0, 0, 255), 2);
          putText(RoI_Object, "Object", P1, FONT_HERSHEY_PLAIN, 1,  Scalar(0, 0, 255, 255), 2);
          dist_Object = (-0.48)*(P2.x-P1.x) + 56.6;

     ss.str(" ");
     ss.clear();
     ss<<"D = "<<dist_Object<<"cm";
     putText(RoI_Object, ss.str(), Point2f(1,130), 0,1, Scalar(0,0,255), 2);

   }

}


int main(int argc,char **argv)
{

   wiringPiSetup();
   pinMode(21, OUTPUT);
   pinMode(22, OUTPUT);
   pinMode(23, OUTPUT);
   pinMode(24, OUTPUT);

   Setup(argc, argv, Camera);
   cout<<"Connecting to camera"<<endl;
   if (!Camera.open())
   {

          cout<<"Failed to Connect"<<endl;
```

```
}

        cout<<"Camera Id = "<<Camera.getId()<<endl;


while(1)
{

auto start = std::chrono::system_clock::now();

Capture();
Perspective();
Threshold();
Histrogram();
LaneFinder();
LaneCenter();
Stop_detection();
Object_detection();
Traffic_detection();

if (dist_Stop > 5 && dist_Stop < 20)
{
        digitalWrite(21, 0);
        digitalWrite(22, 0);    //decimal = 8
        digitalWrite(23, 0);
        digitalWrite(24, 1);
        cout<<"Stop Sign"<<endl;
        dist_Stop = 0;

        goto Stop_Sign;
}

   if (dist_Object > 5 && dist_Object < 30)
{
        digitalWrite(21, 1);
        digitalWrite(22, 0);    //decimal = 9
        digitalWrite(23, 0);
        digitalWrite(24, 1);
        cout<<"Object"<<endl;
        dist_Object = 0;

        goto Object;
}



if (laneEnd > 4500)
{
        digitalWrite(21, 1);
        digitalWrite(22, 1);    //decimal = 7
        digitalWrite(23, 1);
```

```
        digitalWrite(24, 0);
        cout<<"Lane End"<<endl;
}


if (Result == 0)
{
        digitalWrite(21, 0);
        digitalWrite(22, 0);    //decimal = 0
        digitalWrite(23, 0);
        digitalWrite(24, 0);
        cout<<"Forward"<<endl;
}


else if (Result >0 && Result <10)
{
        digitalWrite(21, 1);
        digitalWrite(22, 0);    //decimal = 1
        digitalWrite(23, 0);
        digitalWrite(24, 0);
        cout<<"Right1"<<endl;
}

   else if (Result >=10 && Result <20)
{
        digitalWrite(21, 0);
        digitalWrite(22, 1);    //decimal = 2
        digitalWrite(23, 0);
        digitalWrite(24, 0);
        cout<<"Right2"<<endl;
}

   else if (Result >20)
{
        digitalWrite(21, 1);
        digitalWrite(22, 1);    //decimal = 3
        digitalWrite(23, 0);
        digitalWrite(24, 0);
        cout<<"Right3"<<endl;
}

   else if (Result <0 && Result >-10)
{
        digitalWrite(21, 0);
        digitalWrite(22, 0);    //decimal = 4
        digitalWrite(23, 1);
        digitalWrite(24, 0);
        cout<<"Left1"<<endl;
}
```

```
    else if (Result <=-10 && Result >-20)
{
        digitalWrite(21, 1);
        digitalWrite(22, 0);    //decimal = 5
        digitalWrite(23, 1);
        digitalWrite(24, 0);
        cout<<"Left2"<<endl;
}

    else if (Result <-20)
{
        digitalWrite(21, 0);
        digitalWrite(22, 1);    //decimal = 6
        digitalWrite(23, 1);
        digitalWrite(24, 0);
        cout<<"Left3"<<endl;
}

Stop_Sign:
Object:


if (laneEnd > 4500)
{
   ss.str(" ");
   ss.clear();
   ss<<" Lane End";
   putText(frame, ss.str(), Point2f(1,50), 0,1, Scalar(255,0,0), 2);

 }

 else if (Result == 0)
{
   ss.str(" ");
   ss.clear();
   ss<<"Result = "<<Result<<" (Move Forward)";
   putText(frame, ss.str(), Point2f(1,50), 0,1, Scalar(0,0,255), 2);

 }

 else if (Result > 0)
{
   ss.str(" ");
   ss.clear();
   ss<<"Result = "<<Result<<" (Move Right)";
   putText(frame, ss.str(), Point2f(1,50), 0,1, Scalar(0,0,255), 2);

 }

 else if (Result < 0)
{
```

```
  ss.str(" ");
  ss.clear();
  ss<<"Result = "<<Result<<" (Move Left)";
  putText(frame, ss.str(), Point2f(1,50), 0,1, Scalar(0,0,255), 2);

 }


namedWindow("orignal", WINDOW_KEEPRATIO);
moveWindow("orignal", 0, 100);
resizeWindow("orignal", 640, 480);
imshow("orignal", frame);

namedWindow("Perspective", WINDOW_KEEPRATIO);
moveWindow("Perspective", 640, 100);
resizeWindow("Perspective", 640, 480);
imshow("Perspective", framePers);

namedWindow("Final", WINDOW_KEEPRATIO);
moveWindow("Final", 1280, 100);
resizeWindow("Final", 640, 480);
imshow("Final", frameFinal);

namedWindow("Stop Sign", WINDOW_KEEPRATIO);
moveWindow("Stop Sign", 1280, 580);
resizeWindow("Stop Sign", 640, 480);
imshow("Stop Sign", RoI_Stop);

namedWindow("Object", WINDOW_KEEPRATIO);
moveWindow("Object", 640, 580);
resizeWindow("Object", 640, 480);
imshow("Object", RoI_Object);

namedWindow("Traffic", WINDOW_KEEPRATIO);
moveWindow("Traffic", 0, 580);
resizeWindow("Traffic", 640, 480);
imshow("Traffic", RoI_Traffic);


waitKey(1);
auto end = std::chrono::system_clock::now();
std::chrono::duration<double> elapsed_seconds = end-start;

float t = elapsed_seconds.count();
int FPS = 1/t;
//cout<<"FPS = "<<FPS<<endl;

}


return 0;
```

```
}
```

6.2 Code supporting the Vehicle motion

```
int i =0;
unsigned long int j =0;

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

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

const int D0 = 0;      //Raspberry pin 21    LSB
const int D1 = 1;      //Raspberry pin 22
const int D2 = 2;      //Raspberry pin 23
const int D3 = 3;      //Raspberry pin 24    MSB

int a,b,c,d,data;


void setup() {

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, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL,255);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,255);

}


void Backward()
{
  digitalWrite(HighL, HIGH);
  digitalWrite(LowL, LOW);
  analogWrite(EnableL,255);

  digitalWrite(HighR, HIGH);
  digitalWrite(LowR, LOW);
  analogWrite(EnableR,255);

}

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,160);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,255);

}
```

```
void Left2()
{
  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL,90);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,255);

}


void Left3()
{
  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL,50);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,255);

}

void Right1()
{
  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL,255);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,160);  //200

}
void Right2()
{
  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL,255);

  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableR,90);   //160

}

void Right3()
{
  digitalWrite(HighL, LOW);
```

```
 digitalWrite(LowL, HIGH);
 analogWrite(EnableL,255);

 digitalWrite(HighR, LOW);
 digitalWrite(LowR, HIGH);
 analogWrite(EnableR,50);   //100

}

void UTurn()
{
 analogWrite(EnableL, 0);
 analogWrite(EnableR, 0);
 delay(400);

 analogWrite(EnableL, 250);
 analogWrite(EnableR, 250);    //forward
 delay(1000);

 analogWrite(EnableL, 0);
 analogWrite(EnableR, 0);
 delay(400);

 digitalWrite(HighL, HIGH);
 digitalWrite(LowL, LOW);
 digitalWrite(HighR, LOW);   //   left
 digitalWrite(LowR, HIGH);
 analogWrite(EnableL, 255);
 analogWrite(EnableR, 255);
 delay(700);

 analogWrite(EnableL, 0);
 analogWrite(EnableR, 0);
 delay(400);

 digitalWrite(HighL, LOW);
 digitalWrite(LowL, HIGH);
 digitalWrite(HighR, LOW);   // forward
 digitalWrite(LowR, HIGH);
 analogWrite(EnableL, 255);
 analogWrite(EnableR, 255);
 delay(900);

 analogWrite(EnableL, 0);
 analogWrite(EnableR, 0);
 delay(400);

 digitalWrite(HighL, HIGH);
 digitalWrite(LowL, LOW);
 digitalWrite(HighR, LOW);    //left
 digitalWrite(LowR, HIGH);
```

```
  analogWrite(EnableL, 255);
  analogWrite(EnableR, 255);
  delay(700);


  analogWrite(EnableL, 0);
  analogWrite(EnableR, 0);
  delay(1000);



  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);
  digitalWrite(HighR, LOW);
  digitalWrite(LowL, HIGH);
  analogWrite(EnableL, 150);
  analogWrite(EnableR, 150);
  delay(300);
}


void Object()
{

  analogWrite(EnableL, 0);
  analogWrite(EnableR, 0);          //stop
  delay(1000);

  digitalWrite(HighL, HIGH);
  digitalWrite(LowL, LOW);
  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);       //left
  analogWrite(EnableL, 250);
  analogWrite(EnableR, 250);
  delay(500);

  analogWrite(EnableL, 0);
  analogWrite(EnableR, 0);          //stop
  delay(200);

  digitalWrite(HighL, LOW);
  digitalWrite(LowL, HIGH);         //forward
  digitalWrite(HighR, LOW);
  digitalWrite(LowR, HIGH);
  analogWrite(EnableL, 255);
  analogWrite(EnableR, 255);
  delay(1000);

  analogWrite(EnableL, 0);         //stop
  analogWrite(EnableR, 0);
  delay(200);
```

```
 digitalWrite(HighL, LOW);
 digitalWrite(LowL, HIGH);
 digitalWrite(HighR, HIGH);       //right
 digitalWrite(LowR, LOW);
 analogWrite(EnableL, 255);
 analogWrite(EnableR, 255);
 delay(500);

 analogWrite(EnableL, 0);           //stop
 analogWrite(EnableR, 0);
 delay(1000);


 digitalWrite(HighL, LOW);
 digitalWrite(LowL, HIGH);
 digitalWrite(HighR, LOW);      // forward
 digitalWrite(LowR, HIGH);
 analogWrite(EnableL, 150);
 analogWrite(EnableR, 150);
 delay(500);

  i  = i+1;
}

void Lane_Change()
{

 analogWrite(EnableL, 0);
 analogWrite(EnableR, 0);         //stop
 delay(1000);

 digitalWrite(HighL, LOW);
 digitalWrite(LowL, HIGH);
 digitalWrite(HighR, HIGH);
 digitalWrite(LowR, LOW);       //Right
 analogWrite(EnableL, 250);
 analogWrite(EnableR, 250);
 delay(500);

 analogWrite(EnableL, 0);
 analogWrite(EnableR, 0);         //stop
 delay(200);

 digitalWrite(HighL, LOW);
 digitalWrite(LowL, HIGH);        //forward
 digitalWrite(HighR, LOW);
 digitalWrite(LowR, HIGH);
 analogWrite(EnableL, 255);
 analogWrite(EnableR, 255);
 delay(800);
```

```
 analogWrite(EnableL, 0);         //stop
 analogWrite(EnableR, 0);
 delay(200);

 digitalWrite(HighL, HIGH);
 digitalWrite(LowL, LOW);
 digitalWrite(HighR, LOW);       //LEFT
 digitalWrite(LowR, HIGH);
 analogWrite(EnableL, 255);
 analogWrite(EnableR, 255);
 delay(500);

 analogWrite(EnableL, 0);            //stop
 analogWrite(EnableR, 0);
 delay(1000);


 digitalWrite(HighL, LOW);
 digitalWrite(LowL, HIGH);
 digitalWrite(HighR, LOW);      // forward
 digitalWrite(LowR, HIGH);
 analogWrite(EnableL, 150);
 analogWrite(EnableR, 150);
 delay(500);


}


void loop()
{
   if (j > 25000)
   {
     Lane_Change();
     i = 0;
     j = 0;
   }

 Data();
 if(data==0)
  {
    Forward();
    if (i>0)
    {
     j = j+1;
    }
  }

 else if(data==1)
  {
```

```
  Right1();
      if (i>0)
  {
   j = j+1;
  }
 }

else if(data==2)
 {
  Right2();
      if (i>0)
  {
   j = j+1;
  }
 }

else if(data==3)
 {
  Right3();
      if (i>0)
  {
   j = j+1;
  }
 }

else if(data==4)
 {
  Left1();
      if (i>0)
  {
   j = j+1;
  }
 }

else if(data==5)
 {
  Left2();
      if (i>0)
  {
   j = j+1;
  }
 }

else if(data==6)
 {
  Left3();
      if (i>0)
  {
   j = j+1;
  }
 }
```

```
else if(data==7)
 {
   UTurn();
 }

 else if (data==8)
 {
    analogWrite(EnableL, 0);
    analogWrite(EnableR, 0);
    delay(4000);

    analogWrite(EnableL, 150);
    analogWrite(EnableR, 150);
    delay(1000);
 }

    else if(data==9)
 {
   Object();
 }

       else if(data==10)
 {
    analogWrite(EnableL, 0);
    analogWrite(EnableR, 0);
    delay(2000);
 }


   else if(data>10)
 {
   Stop();
 }




}
```

In this chapter, the detailed simulation through code by using CPP along with geany and Arduino software's as a platform for running the code and the code has been optimized to its optimal level. In the next chapter, results and conclusion of the overall experimentation will be mentioned.

Chapter 7 **Results and Discussion**

In the previous chapter, the detailed simulation through code by using CPP as a platform for running the code and the code has been optimized to its optimal level. In this chapter, the analysis of the system and its comparison to the available systems.

7.1 **Results**

The simulation on lane detection has been completed. The real time path images were taken and they were processed through the code by using image processing. The code successfully identified the lane and bounds have been marked on the image. Later the code is being applied to video which contains many frames. The same happened with the video input too. The code successfully identified the lane on video and path was marked on the input. The code has been optimized and results yielded positively towards the desires output.

In this session, the prototype was designed using robo chassis board. The board has Arduino, raspberry pi and L298 motor driver along with DC Motors. The software's has been well utilised and the code was well compiled and the Prototype ran successfully with some limitations. The prototype is able to detect path as well as objects that it has been trained to identify. It is also responding dynamically whenever it encountered the objects that it has been trained with the cascade software.

Chapter 8 **Conclusion and Future Scope**

In previous chapter, the analysis, comparison and conclusion for this autonomous system have been articulated. The analysis was deeply scrutinized. In this chapter, overall conclusion for the experiment, advantages, limitations and its applications are noted.

8.1 **Conclusions**

The control of autonomous vehicle through image processing is mostly dependent on openCV for programming the code. Primarily the work-plan is divided into 3 parts. The first section is lane detection. Lane detection in the autonomous system has completed by taking several real time path images and the code which processes the images was applied on it. The code efficiently ran and processed the image and given required data for the autonomous system. The prototype was developed based upon the literate review. While operating the prototype there has been some complications occurred. They were crossed through careful observation of the code and the prototype bridging. Still, there are some limitations to the system that needs to be fixed and developed further in order to classify this as a fully automated vehicle.

lane detection as well as object detection is completed. The entire project work is made into a prototype. The data has been dumped into the prototype. The prototype can detect lane as well as identify the objects. It is able to identify traffic control indications. It can respond to the encountered objects dynamically.

8.2 **Advantages, Limitations and Applications**

8.2.1 **Advantages**

The main advantage of this autonomous system is that it can reduce the fatality rate, accidents etc., in addition to this, long journey has a tremendous possibility. The driver can take the necessary nap while the autonomous system is controlling the vehicle. Reduced traffic congestion, increased lane capacity, lower fuel consumption, reduced travel time and transportation costs, natural fuels saving.

8.2.2 **Limitations**

Autonomous vehicles don’t track the centre line of the street well on ill-maintained roads. They can’t operate on streets where the line markings are worn away. Also, it is difficult to operate these cars on heavy snow fall (overall completion of the experiment will reveal more of the limitations)

8.3 Future Scope

In order to make it highly accurate and responsive, it is planned that a greater number of objects should be introduced to the autonomous system. The project is currently heading towards relative motion of the vehicle with respect to other vehicles.

In this chapter, overall conclusion for the experiment, advantages, limitations and its applications are noted.

References

[1] S. international, “Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles,” SAE International, (J3016), 2016.

[2] A. Gray, Y. Gao, J. K. Hedrick, and F. Borrelli “Robust predictive control for semi-autonomous vehicles with an uncertain driver model,” in 2013 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2013, pp. 208–213.

[3] M. Blanco, J. Atwood, H. M. Vasquez, T. E. Trimble, V. L. Fitchett, J. Radlbeck, G. M. Fitch, S. M. Russell, C. A. Green, B. Cullinane et al “Human factors evaluation of level 2 and level 3 automated driving concepts,” Tech. Rep., 2015.

[4] C. Gold, D. Dambock, K. Bengler, and L. Lorenz, “Partially automated driving as a fallback level of high automation,” in 6. Tagung Fahrerassistenzsysteme, 2013.

[5] “Euro NCAP 2018 automated driving tests,” http://bit.ly/2Wxinhp, accessed : 2019-03-30.

[6] A. Roy, “The language of self-driving cars is dangerous– here’s how to fix it,” http://bit.ly/2WF86Qx, accessed: 2019-03-30.

[7] H. Abraham, B. Seppelt, B. Mehler, and B. Reimer, “What’s in a name: Vehicle technology branding & consumer expectations for automation,” in Proceedings of the 9th international conference on automotive user interfaces and interactive vehicular applications. ACM, 2017, pp. 226–234.

[8] L. Fridman, “Tesla Vehicle Deliveries and Autopilot Mileage Statistics,” Jan. 2019.

[9] M. Dikmen and C. M. Burns, “Autonomous driving in the real world: Experiences with tesla autopilot and summon,” in Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. ACM, 2016, pp. 225–228.

[10] M. R. Endsley, “Autonomous driving systems: A preliminary naturalistic study of the tesla model s,” Journal of Cognitive Engineering and Decision Making, vol. 11, no. 3, pp. 225–238, 2017.

[11] N. Kalra and S. M. Paddock, “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?” Transportation Research Part A: Policy and Practice, vol. 94, pp. 182–193, 2016.

[12] L. Fridman, D. E. Brown, M. Glazer, W. Angell, S. Dodd, B. Jenik, J. Terwilliger, J. Kindelsberger, L. Ding, , B. Seppelt, L. Angell, B. Mehler, and B. Reimer, “MIT autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and interaction with automation,” CoRR, vol. abs/1711.06976, 2017. [Online]. Available: http://arxiv.org/abs/1711.06976

[13] N. H. Mackworth, “The breakdown of vigilance during prolonged visual search,” Quarterly Journal of Experimental Psychology, vol. 1, no. 1, pp. 6–21, 1948.

[14] K. R. Boff, L. Kaufman, and J. P. Thomas, “processes and performance,” Harry G Armstrong Handbook of perception and human performance. volume 2. cognitive Aerospace Medical Research Lab, Tech. Rep., 1994.

[15] D. W. Bates, M. Cohen, L. L. Leape, J. M. Overhage, M. Shabot, and T. Sheridan, “Reducing the frequency of errors in medicine using information technology,” Journal of the American Medical Informatics Association, vol. 8, no. 4, pp. 299–308, 2001.

[16] L. R. Hartley, P. Arnold, H. Kobryn, and C. MacLeod, “Vigilance, visual search and attention in an agricultural task,” Applied ergonomics, vol. 20, no. 1, pp. 9–16, 1989.

[17] M. R. Endsley and E. O. Kiris, “The out-of-the-loop performance problem and level of control in automation,” Human factors, vol. 37, no. 2, pp. 381–394, 1995.
[18] R. Molloy and R. Parasuraman, “Monitoring an automated system for a single failure: Vigilance and task complexity effects,” Human Factors, vol. 38, no. 2, pp. 311–322, 1996.
[19] Shine L, Jiji CV (2020) Automated detection of helmet on motorcyclists from traffic surveillance videos: a comparative analysis using hand-crafted features and CNN. Multimedia Tools Application.
[20] Liu J, Yang Y, Lv S, Wang J, Chen H et al (2019) Attention-based BiGRU-CNN for Chinese question classification. J Ambient Intell Humaniz Compute
[21] Aamir M, Pu Y, Rahman Z, Abro WA, Naeem H, Ullah F, Badr AM (2018) A hybrid proposed framework for object detection and classifcation. J Inf Process Syst
[22] Szegedy C, Liu W, Jia Y, Sermanet, P, Reed S (2015) Going deeper with convolutions. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Boston, Massachusetts, 7–12 June 2015
[23] Ren S, He K, Girshick R et al (2015) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149