GROUP - 1

CAPSTONE PROJECT

REPORT

on

AUTONOMOUS VEHICLE CONTROL THROUGH TRAFFIC SIGNS

Project Team Members: EE, 7th SEMESTER

AKANKSHA SINGH - 101904104

AYUSH VASHISTH - 101904020

GURSIMAR SINGH NAGPAL- 101904114

VINAY KUMAR JAIN - 101904017

YASHVARDHAN PACHAURI- 101904095

Under the Guidance of

Dr. SANJAY K. JAIN (PROFESSOR)

DR VISHAL SRIVASTAVA (ASSOCIATE PROFESSOR)

2022

Thapar Institute of Engineering and Technology

Electrical & Instrumentation Engineering Department

Punjab (India)

DECLARATION

We hereby declare that the project entitled **Autonomous Vehicle Control Through Traffic Sign** is an authentic record of our own work carried out in the Electrical & Instrumentation Engineering Department, Thapar Institute of Engineering and Technology, Patiala under the guidance of **Dr. Sanjay K. Jain,** Professor and **Dr. Vishal Srivastava**, Associate Professor during July-December 2022.

Date: **02/11/2022**

S. No.	Name of the Student	Roll No.	Signature
1	Akanksha Singh	101904104	
2	Gursimar Singh Nagpal	101904114	
3	Vinay Kumar Jain	101904017	
4	Ayush Vashisth	101904020	
5	Yashvardhan Pachauri	101904095	

Faculty Supervisor:

Supervisor(s)

1. Dr. Sanjay K. Jain
   (Professor, EIED)
2. Dr. Vishal Srivastava
   (Associate Professor, EIED)

I thoroughly reviewed the project report, and students amended it in accordance with my recommendations.

Project Report Mentor:

1. Dr. Rakesh Yadav
   (Assistant Professor, EIED)

06/12/2022

ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude to our supervisors **Dr. Sanjay K. Jain and Dr. Vishal Srivastava** and also to report mentor **Dr Rakesh Yadav** of Electrical and Instrumentation Engineering Department (EIED), Thapar Institute of Engineering & Technology for their **,** who guided us towards the successful completion of this project. We would like to thank other the faculty & staff members particularly the evaluation panel for their patience and time in evaluating us and for suggestions to improve this project work. Also, we would like to thank our friends and all other respondents and group members whose responses and co-ordination were of utmost importance for the completion of this project work.

S. No.	**Name of the Student**	**Roll No.**
1	Akanksha Singh	101904104
2	Gursimar Singh Nagpal	101904114
3	Vinay Kumar Jain	101904017
4	Ayush Vashisth	101904020
5	Yashvardhan Pachauri	101904095

ABSTRACT

Autonomous vehicles are the vehicles of the future. The research is underway in many developed countries to replace human-driven vehicles with autonomous vehicles on roads. The features like self-driving nature, environmentally friendly, automatic obstacle detection, less fuel consumption, time-efficient, robustness; etc are fascinating the people to adapt them, But autonomous vehicles are still a distant dream for India. This is because these vehicles will find most Indian roads too chaotic to function. Such chaos can be due to the poor road condition, accumulation of vehicles in a single place beynd its capacity, or maybe due to some environmental cause. Negligence of drivers toward traffic signs, traffic signals and not to mention traffic rules sometimes causes critical accidents and therefore must be given serious attention. This problem is only expected to get worse, with the continued increase in the numbers of vehicles in urban centers around the world . One of the ways to limit this problem is by using Traffic Sign Detection And Recognition (TSDR) System. The TSDR system can be supportive to the Intelligent Transportation System (ITS). It has been developed and implemented with the aims of minimizing the number of accidents caused by driver' negligence or poor judgement. In this project an attempt to implement "TSDR system" on IOT-based hardware which will culminate as a proto-model of the real-life autonomous vehicle system that can support drivers and increase driving safety**.** The system identify and recognizes traffic signs from images captures by cameras or imaging sensors , and shows users the traffic rules that apply to that road. In spite of the technology's warnings, it is possible that the user may violate the traffic sign rules, and violating the traffic sign rules may cause accidents that endanger people's lives.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

TSDR	-	Traffic Sign Detection And Recognition
WSN	-	Wireless Sensor Network
ROI	-	Region of Intersection
ANN	-	Artificial Neural Network
API	-	Application Programming Interface
RGB	-	Red Blue Green
TSDR	-	Traffic Sign Detection & Recognition
IoT	-	Internet of Things
GPIO	-	General Purpose Input/Output
CSI	-	Camera Serial Port
IDE	-	Integrated Development Environment
CNN	-	Convolutional Neural Network
RF	-	Radio Frequency
GTSRB	-	German Traffic Sign Recognition Benchmark

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Driverless cars, robotic cars, and self-driving vehicle are all autonomous vehicles that can fulfill classic cars' main transportation. An autonomous vehicle can sense its surroundings and navigate on its own. The majority of automated vehicles are prototypes or demonstration systems. Globally, the number of people traveling by vehicle is increasing dramatically, which results in issues with traffic congestion and issues with traffic light junctions among the general public. Specifically, traffic light monitoring and control do not provide adequate procedures in an emergency. Traffic jam leads to delay in the ambulance to reach its destination on time which is can be critical. Traffic light control offers early warning of approaching emergency vehicles and guides them through traffic jams to their goal. The system was developed A multi-agent-based autonomous traffic light controller called fuzzy logic that systemizes In order to combat issues like traffic jams, accidents, and speeding, wireless sensors are used. Traffic light control system which is intelligent, introduced reinforcement learning opted by adaptive optimization technique is presented by an intelligent. Based on radio frequency (RF) transmission Radio, frequency emergency vehicle traffic signal control system is designed to operate when signals from crisis vehicles are received. This novel distributed traffic light control design uses network of wireless sensor. Non Wire sensors are placed in lanes of traffic entering and exiting intersections. The latest methods are used to regulate the traffic flow sequence in an traffic control system which is adaptive built on the new Wireless Sensor Network (WSN) traffic infrastructure. These technologies dynamically adapt to traffic conditions at both single and multiple intersections. System took the lead of the crisis of new technologies to construct signal control smarter. They use communication, sensor networks, and more advanced algorithms to configure traffic lights. Also, A system was created to control traffic flow, introducing a new technology "dynamic traffic controller" Intersection game theory which follows traffic control is one group of techniques that is employed for algorithms that can leverage the capabilities of future autonomous/automated vehicles to wash away the usual state-of-the-art for the control system.

Forward-facing cameras can be used to analyze Traffic signs in many modern cars, vehicles, and trucks. In addition, one of the main applications of a traffic-sign recognition system is recognising the speed limit. Typical speed information can also be extracted from additional speed limit signs and displayed on the dashboard of the car to notify the driver of the signal, GPS data is the primary source of speed information. Many high-end vehicles, mainly European models, have this advanced driver-assistance feature.

A traffic sign is a roadside device that uses words or symbols to convey, guide, limit, warn, or direct information. As a result of the advancement of sophisticated automotive technologies, Mercedes-Benz, BMW, and other well-known automobile manufacturers are actively investing in his ADAS (Advanced Driver Assistance Systems) research. Lane departure warning systems and TSR (Traffic Sign Recognition) systems, which warn drivers to check their speed, are both types of ADAS systems that are currently available. Traffic accidents may result if drivers and pedestrians are not aware of this information. Computer technology is desperately needed to automatically recognise traffic signs given the growing demand for car intelligence. To address this issue, research in this field started in the 1980s.

1.2 LITERATURE REVIEW

Research on traffic sign recognition started long back however the first paper was published back in 1987. This paper was researched by Akatsuka and Imai who tried to create a simple traffic sign recognition system. A system that can selfrecognize other road signs and is used to assist drivers by alerting them to certain restrictions and hazards during speeding or road works. Aditionally it can assist the drivers to automatically recognize certain traffic signs. Recognition and categorization are the two general components of the traffic sign recognition process. A. Identification Finding the areas of interest (ROI) where traffic signs are most likely to be discovered is the aim of employing traffic sign recognition. The technique to trim the excess space to get the main sign is called as ROI. Region of Interest is used to identify the traffic sign based on its shape dimensions etc. The signs detected are cropped and the background image is removed as it is not an region of interest. By removing the background image, We presume that a sizable portion of the image area might be disregarded as unnecessary.. Traffic signs are created with predetermined colours and shapes, making them simpler to distinguish and identify. B. Classification The shape, colour, and information provided by the sign are used to classify it in this section..

Features of Histogram of oriented gradients (HOG) : HOG breaks single image into tiny, squared-off, to calculate in individual cell a histogram of oriented gradients, further regularize the outcome with the help of a block-wise pattern, further outputs a narration for every cell. Traffic sign detection has been done using a wide variety of techniques. The majority of these methods rely on the HOG and SIFT characteristics.

Many research works were presented in the literature. Using classifiers like colour information and Bags of Features (BoF) as a features extractor, Ohgushi etal introduced a traffic sign classifier. The proposed method struggles, especially when the sign is heavily lighted or partially obscured, to read traffic signs in actual situations. In several studies, the detection of the traffic sign was studied without the classification procedure being used. A method to identify only circular traffic signs on Chinese highways was put forth by Wu et al. On the other hand, researchers concentrate on spotting and identifying the traffic sign. The suggested approach can only identify round signs; it is unable to identify other sign forms. Wali et al. suggested a three-step strategy for detecting and identifying traffic signs. Data preparation was the initial stage. The second step involved finding the symbol, and the third step involved classifying it. They employ colour segmentation with form matching for the detection process and use SVM as a classifier for the classification step. The proposed approach has an accuracy rate of 95.71%. A smartphone-based approach for reading traffic signs was introduced by Lai et al. In order to execute colour space segmentation and shape identification methods employing template matching by computing similarity, they used colour detection. optical character recognition is another[1].

To select the signing class, (OCR) was applied inside the shape's border. Only red traffic signs were included in the proposed system. Gecer et al. suggest using COSFIRE based on colour blobs to identify traffic signs. The suggested technique was based on a Combination of Shifted Filter Responses with computing the response of various filters in various regions in each colour space channel (i.e. RGB). Shamset al. employed a method based on a BoW descriptor modified utilising a spatial histogram to boost the classification process based on an SVM classifier[2].

Hankizi unveils a driving assistance system that employs camera put in place on a vehicle to detect traffic signs in real time. The feature generation and feature matching involved in the suggested recognition system by identifying in order to provide sufficient information for the driver assistance functions. Preprocessing, color-based segmentation, and form feature-based

detection are all carried out as part of a three step detection process to obtain the results. The video sequences are used to evaluate performance based on detection rate. Jung employed a classification and training technique for an artificial neural network. The traffic sign board which was able to be recognise by driver assistance which is for further information about the roads with the use of the currently available deep convolution traffic sign recognition technology. The performance of the recognition framework for autonomous cars is real-time. A precision parameter that takes into account the light level and other environmental elements is used to evaluate performance. After identifying the traffic sign boards using a rectangle detection technique, Moutarde employed a neural network approach to differentiate between them[3].

Additionally, Hua and Angela used Hough-change to display quadrilateral sign sheets. To identify the sign board they used a near circuit ID and an abundance line elimination method after first using an incline-based Hough change for identifying image's edges. Loy and Barnes developed a figure that can recognise polygon-shaped signs using an extended symmetry identifier (such as square, triangle, and octagon). The accuracy of these form creation processes heavily depends on the edge that they employ, and these edges are defenceless against noisy pixels. Plans call for specific symmetrical shapes for Sign Boards. Several authors have created sign's boards various figures using the evenly shaped shape characteristics[4].

By utilising the symmetry component of the signs, Zelinsky and Barnes were able to recognise signs in the image that had some degree of winding evenly shaped. In a further study, Barnes took help of merit polygon disclosure figures and extended symmetry to reduce the visual noise in the images and enhance recognition rates. In any event, it is seen that these sign board symmetry features are colliding when assessed from the photographs when subjected to various challenge adjustments and observations. Additionally, sign board identification is carried out employing design organising techniques. They are nonetheless connected with more little zones that are managed by executing shading division earlier utilising design planning because they are computationally expensive. Betke and Makris propose a quick arrangement planning approach based on imitation toughening figure. The crucial components of the viewed objects are the ones for which features are learned. There are many calculations in the literature that evaluated the invariant characteristics of the printed characters[5].

For instance, Hu's inferred geometric moment invariants are incorporated into numerous optical character affirmation figures. A line of action for minutes was suggested by Zernike in

light of Zernike polynomials. The amount of character holes as well as the characters' level and vertical projections were employed by Rodriguez and Maudes as features. Character affirmation being considered from the standpoint of case mapping was suggested by Wu et al. The histograms of the various character sections provided them with the features. Additionally, character affirmation computations use the Hausdorff exclusion, which is used to take into account two double pictures. Additionally, calculations have been performed with regard to support vector machines and organising frameworks. Another important method for optical character confirmation in numerous applications is the neural framework. Additionally, a probabilistic neural networks approach is employed to recognize letter sets[6].

1.3 NEED ANALYSIS

The human identification system strong enough for managing various dynamic variables, such as luminance and visibility, whereas computer-based recognition faces considerable difficulties. Road and traffic sign identification looks to be an easy problem to solve because its objective is clearly stated. Road signs are erected in specified sites, and their shapes, colours, and pictograms are also predetermined. However, a number of factors that have an impact on how well the detection system performs must be thoroughly examined in order to completely understand the issue. For the current investigation, digital camera photos of road signs are used to collect data. Motion blur, however, may be present in still photographs taken with a moving camera. Additionally, some photographs may show road signs that have been partially or completely covered by other things like cars or people. The presence of structures or billboards that resemble traffic signs can interfere with the equipment and complicate sign recognition, among other problems. The system have capability for handling traffic and road signs when lighting and weather conditions occurs, including those brought on by the passing of the seasons and a variety of weather patterns conditions like sun uphead, fog incurred, rain , and snow. In one section of this chapter, various potential issues are illustrated. Making use of the system in many nations may make the issue worse. Different pictograms and colours are used in various nations. In order for the system to be adaptable, it must support ongoing learning; otherwise, each nation's training program must be repeated. To address all of these issues and give the system the ability for outcome appropriately if detected a traffic sign, large number of sign samples must be given to road sign recognition.

TRAFFIC SIGN RECOGNITION

The field of traffic sign recognition is focused on the recognition and identification of signs on roadside and traffic chalk in traffic scenes recorded by a camera. Here the procedure makes use of artificial intelligence and computer vision to draw out road signs from outdoor photographs captured in erratic bright conditions. These photographs might contain signs that are obscured by other objects or have other issues like lighting of color, not oriented properly, and differences in size as well shape.

1.4 PROBLEM FORMULATION

The main goal behind the system is to improve automation, which would decrease human error and boost speed, efficiency, and reliability. Due to congestion caused by other vehicles, it is very likely that a driver will miss some traffic signs under the current traffic management systems. The number of vehicles in metropolitan agglomerations around the world is increasing steadily, thus this issue is only expected to get worse. In order to detect and recognize all incoming traffic signs, an automobile can be equipped with a visual-based traffic sign recognition system. If the driver chooses not to obey the traffic signs, the same would be shown to them with alarm-triggering elements.

1.5 OBJECTIVES

1. To develop an automated model for the plotting Traffic Signs on the roadsides using Deep Learning, Data Analysis & Image Processing.
2. To deploy the model on Traffic Sign Detection & Recognition (TSDR) System with Violation Control on a miniature vehicle by using concepts of Communication and IoT.

1.6 EXPECTED DELIVERABLES

1. The proposed proto-model will be able to detect traffic signs and will use the Voice Alert Message to notify the driver regarding the type of sign.
2. Furthermore, the proto-model will be tested keeping the following conditions into consideration while capturing images from Video frame sampling in case the traffic sign is detected. Perspective distortion, shifting lighting, partial occlusions, and shadows are the circumstances.

3. Additionally, it must disclose the existence of any potential issues, such as poor visibility, unpleasant weather, and poor positioning.

1.7 NOVELY

Autonomous driving is one of the most anticipated technologies of the 21st century. The various components of methodology and deliverables, mentioned above are implemented and researched separately in several research papers and projects. Through this project, we want to incorporate all the separately researched attributes into a single system and present it in the form of IoT based Hardware proto-model. It is clear why such a system was established given the advantages it offers in terms of saving lives and lowering expenses. The suggested system has the ability to spot patterns in images that were captured by a camera and analysed by a Deep CNN Network. Human error, including drivers who miss a signals or go in the opposite direction of what a traffic sign indicates (for example, a sign indicating a speed limit of 100 KM and a driver travelling at a higher speed). This project's primary objective is to develop and improve efficiency-related challenges.

1.8 CONCLUSION

The proposed area's literature review was reported in this chapter, along with the research gap that resulted from it. Therefore, based on the research deficit, numerous objectives are framed. This chapter includes, the originality as well as report deliverables for suggested work.

CHAPTER 2

THEORY, STANDARDS AND CONSTRAINTS

2.1 OVERVIEW

The three fundamental steps in traffic sign identification are the extraction of the road sign area of interest (ROI), its refinement and classification into relevant classes, and subsequent processing. The only available options for ROIs are grayscale and RGB channels. Convolutional neural networks are used to categorise them once they have been trained on vast amounts of data. Speed and accuracy of recognition are important for traffic sign recognition and implementation in real-time scenarios. Shapes and colours may be twisted as a result of real-world circumstances. Therefore, this classification method has proven to be simple, powerful, and accepted. Small images may be difficult to see. To detect these small targets, advanced modules are introduced that prevent resolution loss and preserve its properties. A passthrough module is thus recommended. This module combines the data from the preceding layer with data from a different layer. The system requires Bluetooth hardware, motor drivers, software for user input, and mobile application development. Traffic signs (speed limit signs) are classified using a Python library called Keras for sign recognition. The motivation behind the development of such a system is clear, as it has life-saving and cost-saving benefits. The proposed system in the figure 2.1. has the ability to recognise individuals in images captured by a camera and processed by a Deep CNN network. Drivers who miss or fail to see certain traffic signs are primarily to blame for auto accidents (e.g. Because of this, the primary objective of this project is to develop and enhance related issues. To prevent recognition errors, a recognition system must classify traffic signs into distinct classes in a real-time setting. The system should make real-time predictions to provide information in a timely manner take appropriate action since traffic sign recognition should be incorporated in a driving car environment. Also the recognition & classification of traffic sign must be fast enough compared to human driver. High accuracy and very high predictive performance are also needed to be incorporated in the system. Not to mention the model must also be able to detect traffic signs on considerable distances. In the next stage it requires a highly robust and accurate traffic sign classification algorithm that can solve the suggestion from recognition models and can work well even either false positive suggestions. In this

project our goal is to develop a system that can perform better than human drivers under similar conditions.

2.1.1 Raspberry Pi

The Raspberry Pi (Figure 2.1) is a Linux-powered computer with several general-purpose input/output (GPIO) ports that enable you to use physical computing circuits and research the Internet of Things (IoT). Raspberry Pi uses open-source software. Raspberry Pi Foundation, a UK nonprofit organization, produced the Raspberry Pi range intending to foster computer learning and simplify access to computer science education. The Raspberry Pi has been released in various versions and variations since 2012. The Pi Zero is the most affordable model at around $5. Raspberry pi is very cheap and is like single hardware with multiple functions. The Raspberry Pi has always cost around $100 (typically USD 35.

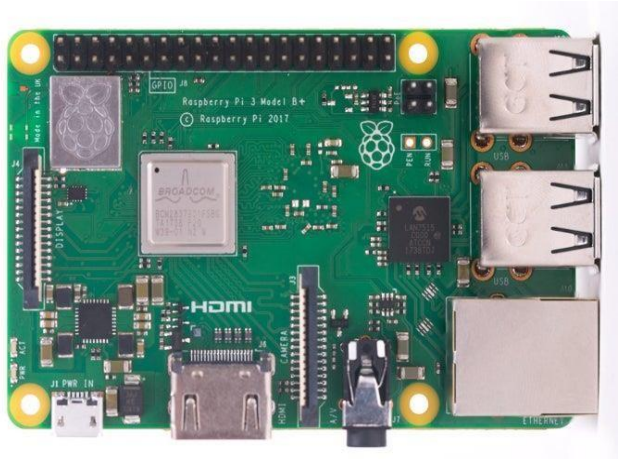

Fig. 2.1 Raspberry Pi

People use Raspberry throughout the world for learning programming, developing hardware-based projects, and automating houses, as well as in commercial settings. The Pi operating system is open-source and uses many open-source programs. There are wide varieties of Raspberry Pi with Linux. There are wide varieties of Raspberry Pi with Linux.

2.1.2 Pi Camera

The Pi Camera Module (Fig. 2.2) encapsulate high resolution images and videos. Direct connection of the Pi camera module is possible RaRKGon account of camera serial port (CSI) interface of the Raspberry Pi boardHere, the Pi Camera v1.3 is used for Pi Camera Features.

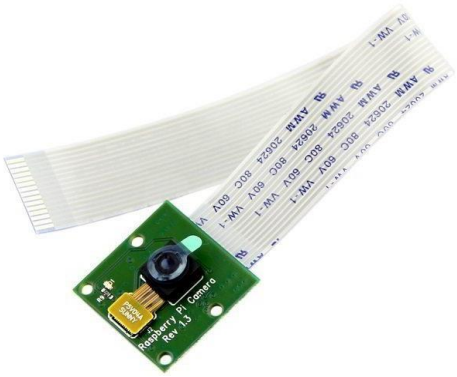

Fig. 2.2 Pi Camera

Its characteristics include: 5 megapixels is the resolution. HD video recording in several frame rates, such as 1080p@30fps, 720p@60fps, and 960p@45fps. Wide still photos with a resolution of 2592x1944 pixels can be captured. The CSI interface is in use.

2.1.3 Arduino Uno

In 2010, Arduino released the open-source microcontroller board known as the Arduino Uno. The Microchip ATMega32P MCU serves as its foundation. Circuitry on the shield and expansion board. The board supports the integrated development environment (IDE) through programming. Additionally, it features 6 analogue I/O pins. a Type B connection USB cord

Fig. 2.3 Arduino Uno

Similar to the Leonardo. Attribution under creative commons. For several hardware revisions, more designs and production files are provided. The original "uno," which means "one" in Italian, stands for the Arduino software. This and the Arduino UDE version 1.0 (Fig 2.3) are the reference releases of Arduino that have subsequently been updated. A pre-programmed bootloader on the board's ATmega328 allows for the loading of new code without the need for an additional hardware programmer.

2.1.4 Bluetooth Module

It is used in a lot of consumer products, like wireless game controllers, wireless headsets, wireless mice, and wireless keyboards, among other things.

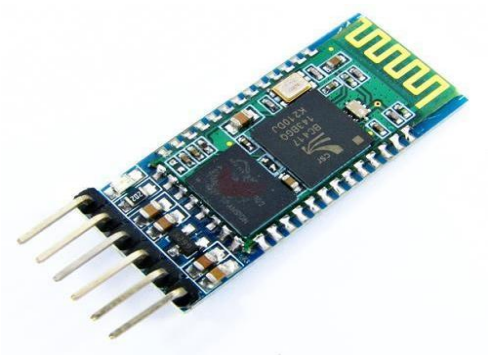

Fig. 2.4 Bluetooth Module

The transmitter and receiver, the atmosphere, the geography, and the conditions of the city all affect the range, which can range up to 100 meters. As shown in fig 2.4

2.1.5 Node MCU

Node MCU (Fig 2.5) is affordable open source IoT platform. Open source prototyping board layouts may be used with Node MCU thanks to its open source firmware. Node and MCU are combined to get the moniker "Node MCU" (microcontroller). The firmware is technically referred to as the "Node MCU" rather than the development kit to which it is related. Both the development board and the firmware are open-source.

Fig. 2.5 Node MCU

The scripting language Lua is used in the firmware. The project, which is based on the ESP8266 Non-OS SDK, is built on top of the firmware. It makes extensive use of open source software and SPIFFS. Users must select the proper module for their project and create the firmware that best meets their demands due to the restricted resources available. 32-bit ESP32 support has also been added.

2.1.6 Bluetooth Speaker

A main speaker and an RF transmitter, which combines the loudspeaker's own RF receiver and transmitter, make up a wireless speaker. The transmitter is connected to any audio device's audio output, including high-end audio equipment, TVs, computers, MP3 players, etc. RCA connectors are typically used for this. The wireless speaker may travel cordlessly because there are no cords involved because the receiver is placed where the listener wishes to hear the sound. The amplifier in receiver/speaker units often amplifies the audio signal before sending it to the loudspeaker. powered by an AC power outlet or a battery. The wireless speaker operates in the 900 MHz signal frequency band, which is commonly used by cordless telephones. RF signals can travel through surfaces like floors and ceilings. The majority of manufacturers state that signals can travel up to 100 meter (150 to 300 feet). In order to avoid potential RF interfaces with nearby wireless equipment like cordless phones or baby monitors, many wireless speakers offer flexible broadcast channels that may be changed with a dial. 2.4 GHz is the frequency range that some models of wireless speakers cover. The 5 GHz frequency band is utilized by the WiFi standard.

2.1.7 Dual H Bridge DC Motor

An "H-bridge" is simply a circuit that controls the direction of rotation by reversing the polarity of the voltage applied to the DC motor. We will use switches to visualize how everything works. Transistors are the typical component of a true H-bridge. Additionally, as previously mentioned, transistors enable PWM control of motor speed. Four switches, all in the open or off state, are shown in the first diagram. A DC motor is the central component of the circuit. The letter "H" with the motor attached to the centre or "bridge" component may be seen clearly in the diagram that has been produced. The term "H-bridge" is therefore utilised. H-Bridge motor driver ICs come in a variety of configurations and perform essentially the same functions.

The L298N is arguably one of the most well-liked. Each integrated circuit in the "L298" family is related to the L298N. The amount of current each family member can handle distinguishes them from one another. At 35 volts DC, the L298N can handle up to 3 amps, which is enough for most hobby motors. Due to its two complete H-bridge circuits (Fig 2.6), the L298N can actually drive two DC motors. Given that the majority of robots have two or four driving wheels, this is perfect for robotics projects.

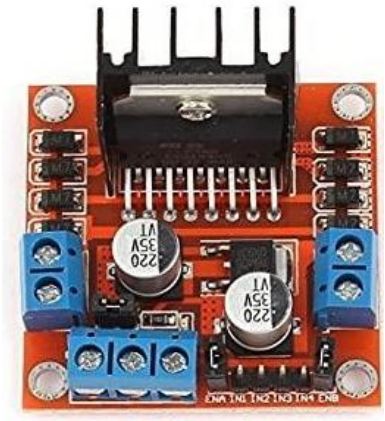

Fig. 2.6 Dual H Bridge DC Motor

2.1.8 Ultrasonic Sensor

An ultrasonic sensor (Fig 2.7) is a device used for the measurement of the distance to a target item with ultrasonic sound waves. As well as for convert sound reflected into an electrical signal. An ultrasonic sensor's two main components are the transmitter and receiver.

Fig. 2.7 Ultrasonic Sensor

The transmitter of an ultrasonic sensor uses piezoelectric crystals to emit sound. The proximity sensor function is the main application for the ultrasonic sensor. Collision avoidance systems and automated parking technology both use it. Ultrasonic sensors are used in both manufacturing technologies and robot obstacle detection systems.

2.2 ASSUMPTIONS AND CONSTRAINTS

1. In addition to the complicated environment of the roads and the surroundings, signs can be found in various states, such as old, damaged, or disoriented. As a result, identifying these signs may present one or more of the following challenges.
2. In the colour of the sign to fade over time due to long-term exposure to sunshine and the interaction of the paint with the air.
3. Meteorological conditions, such as fog, rain, clouds, and snow, impact visibility.
4. Variations in the direction and intensity, intensity, and seasonal and time-of-day variations can all impact visibility. Other factors to consider include shadows cast by nearby objects.
5. Color information is susceptible to changes in lighting circumstances, including shadows, clouds, and sunlight. It can be affected by the lighting type (daylight), illumination geometry, and viewing geometry. As shown in fig 2.8 and fig 2.9.

Fig. 2.8 Faded Sign

Fig. 2.9 Bad Weather Condition (Rain & Snow)

Fig. 2.10 The presence of barriers in the environment.

6. Objects in the scene that block the view of other objects include trees, buildings, cars, pedestrians, and signs as shown in fig 2.10.
7. We have considered 3 different sizes of traffic signs based on its geometric shape as shown in table 2.1

Table 2.1: Overview of traffic sign sizes and shape types

Shape of Traffic Sign	Size 1	Size 2	Size 3
Triangle (edge length)	630	900	1260 (140%)
Rectangle(width*height)	630 * 420	900 * 600	1260 * 840(140%)
Square (edge length)	420	600	840 (140%)
Circle (diameter)	420	600	750 (125%)

8. According to the conventional rule, the size of a speed limit sign depends on the maximum permitted speed at the place of installation as shown in table 2.2

Table 2.2: Overview of speed limit signs and sizes

Circular Shape speed limit (km/h)	Other shapes speed limit (km/h)	Size
0 to 20	20 to 50	1
20 to 80	50 to 100	2
more than 80	More than 100	3

2.3 TECHICAL STANDARDS USED

The standards applied in the suggested methodology are listed below. Prototype of IEEE Recommended Practices for traffic sign recognition. These standards address autonomous vehicle traffic detection system.

IEEE 1481-2009: Describes how to design integrated circuits for chip timing analysis. This assisted in figuring out a square wave's time value in a timer circuit.

IEEE C3 7.90: This standard creates a uniform and repeatable framework for designing and assessing relays and relay systems.. Cables are rated not to exceed 3kV or 5kV (line to line) and have a core temperature of 140°C (284°F) for nitrile, sheathed or EPDM cables. coating. Conductors, insulators, barriers (optional), assemblies with and without armor, manufacturer's test requirements and maximum cable capacity are included.

IEEE 802.15.1 : Wireless personal area networks can be built using the Bluetooth Module, a documented IEEE 802.15.1 standard (PANs). It transmits data wirelessly using Frequency Hopping Spread Spectrum (FHSS) radio technology. I use serial communication to talk to the gadget. Trees, buildings, cars, pedestrians, and signs are among the objects in the scene that obscure the view of other objects. Through a serial port, it connects with the microcontroller (USART).

2.4 CONCLUSION

In this chapter, the reader is made aware of the basic standards one must verify before making a proto-model of traffic detection system. The constraints and assumptions that we have made. After going through the above section, one can easily comprehend various components and its constraints to used in a proto-model.

CHAPTER 3

DESIGN METHODOLOGY

3.1 OVERVIEW

It is proposed to develop practical implementations using intelligent vehicles. The major objective is to increase the system's speed and efficiency by condensing the search space and showing only potential locations. Intelligent algorithms that are more reliable and faster are needed to ensure the accuracy required for traffic sign recognition. The generated image is preprocessed, improved, and segmented in accordance with the colour and form characteristics of the indicators during the detection phase to identify potential pixel regions that could be used to represent a complex traffic sign background, traffic sign images are examined. The latent objects are then normalized to a certain size and introduced into the recognition phase. In this study, only was investigated because the circular and hexagonal shapes of are commonly seen in traffic signs. Multi-layer perceptron (MLP) related backpropagation learning algorithm is an alternative approach to the symbol recognition problem in this study. The image processing tool used in this article is a free, open source, non-commercial computer vision (OpenCV) library.

3.2 CONCEPT MAP AND FLOW CHARTS

The framework we proposed is categorized into 2 components Hardware Part and the Software Part. The system's construction focuses mostly on a ready-made, inexpensive solution that uses a Raspberry Pi tiny embedded microprocessor. Deep learning techniques must be utilised with the aid of the TensorFlow and Keras platforms to deliver results that are processed quickly. The primary function of the suggested prototype is the detection of traffic signs using a convolutional neural network, such as the 40 and 60 mph speed limits, the no left, no right, and no U-turn signals. The observed traffic signals are then encrypted into distinct binary codes and stored in IoT cloud data if the CNN approach was effective in detecting the traffic signals. An Arduino UNO controls the motors in accordance with the control signal supplied to IC L293D after receiving the logged data from a Node MCU, which is continuously collecting the data. A complete block diagram of the proposed prototype is shown in Figure.3.1

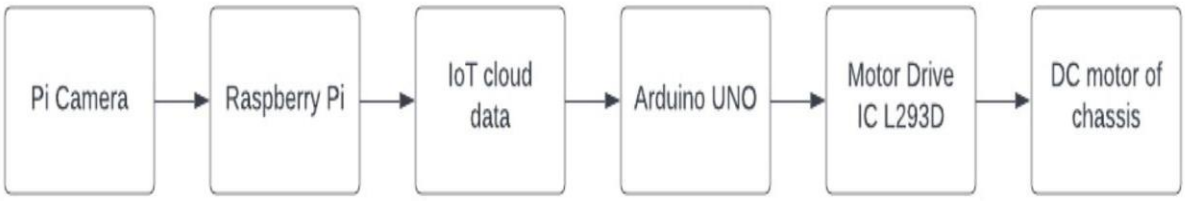

Fig. 3.1. Block diagram of Proposed Prototype

3.3 PROPOSED METHODOLOGY

3.3.1 SOFTWARE IMPLEMENTATION:

A unique deep neural network model will be developed and trained using data from traffic signs. A typical workflow is illustrated using a flow diagram as shown in Figure. 3.2. In a later implementation phase, this model will be used to detect traffic signals in real time.

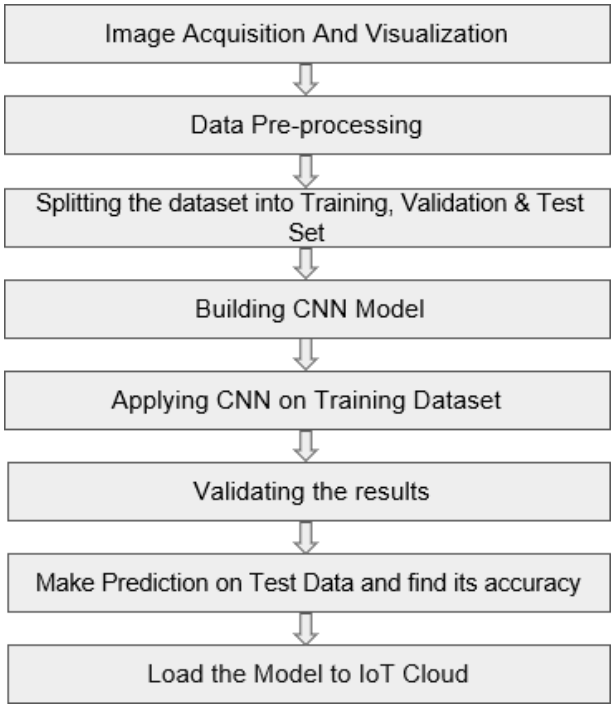

Fig. 3.2 Block diagram of Software Implementation of Traffic Sign Recognition System

3.3.2 HARDWARE IMPLEMENTATION:

The Pi camera will take pictures when it is connected to the Camera Serial Interface (CSI) port on the Raspberry Pi. Images are continually captured, and each frame is processed by a system that has been trained to recognize traffic signs when they are present. A message is then sent to the RC car in accordance with the outcome. In this case, NodeMCU continuously extracts

data from IoT data logs while Arduino is utilised for basic RC car movement, vehicle infraction enforcement when traffic signs are observed, and basic RC car movement. Systems for detecting and recognising traffic signs (TSDR) operate in two steps. detection. At this point, it is known whether there are any road signs or not recognize. Traffic signs that have been detected are identified and categorised in this step. A prototype of the proposed system for detecting sidewalk road signs is shown in Figure. 3.3.

The proposed technique can be conditionally divided into five stages.

1. Video and Frame Capturing
2. Preprocessing
3. Traffic Sign Detection
4. Character/icon extraction
5. Recognition

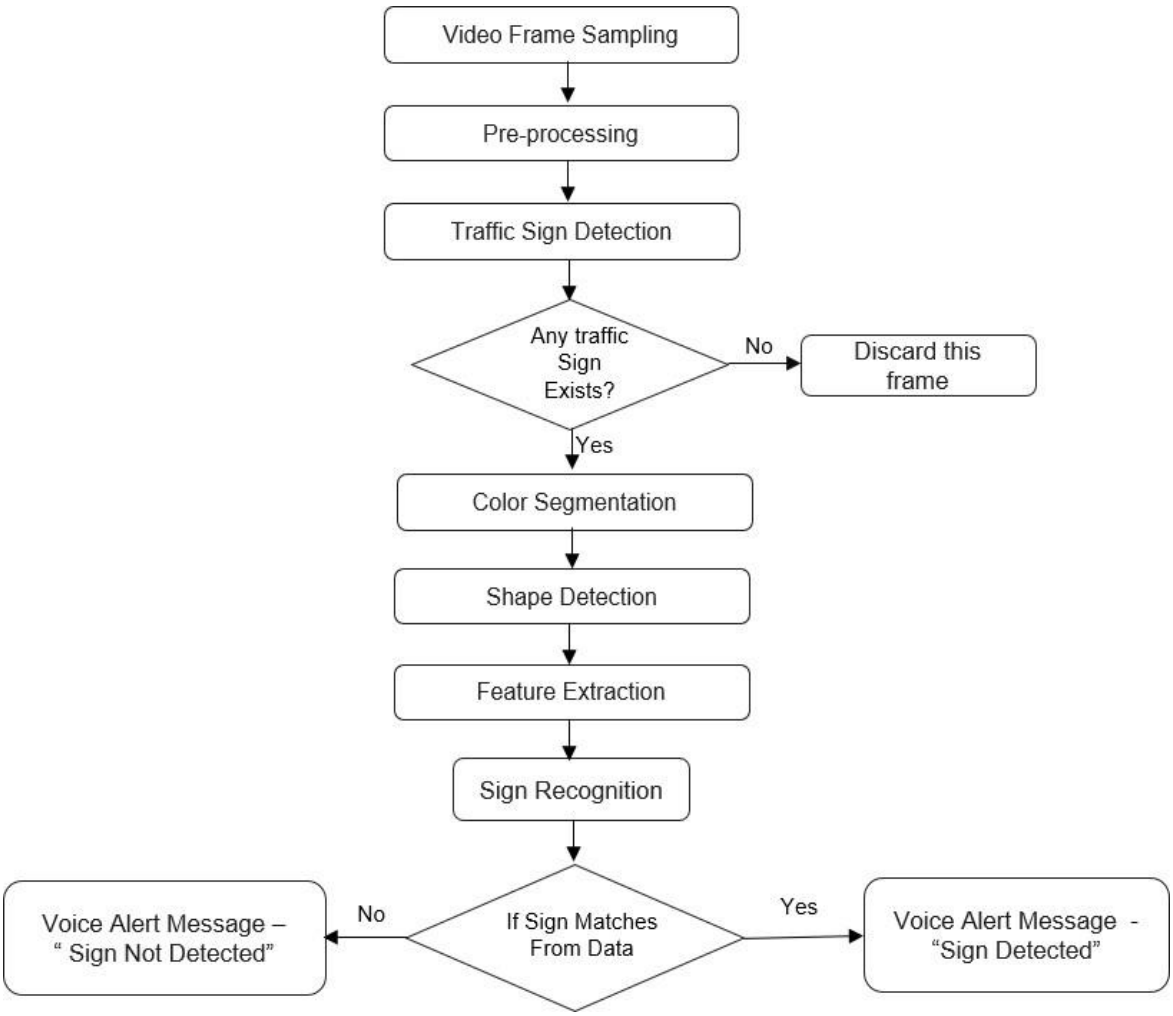

Fig. 3.3 Block diagram of Traffic Sign Detection

3.4 MATHEMATICAL ANALYSIS AND CALCULATIONS

BUILDING THE CNN MODEL

CNN models serve as a representation of a deep neural network's layers in deep learning. Convolutional neural networks, or CNNs, are made up of many layers of fully connected perceptrons. Fully linked describes the state in which every neuron in one layer is connected to every neuron in the following layer in a multi-layer perceptron.Data overflow can be efficiently avoided using this approach. Applications for it specifically include natural language processing, image and video categorization, medical image analysis, and more.

Convolutional neural network The input and output layers of a convolutional neural network are joined by a number of hidden layers (Xu, Ren, Liu, & Jia, 2014). The hidden layers also contain a series of convolutional layers based on multiplication and other scalar operations.

ReLU layer: Group layers, completely connected layers, and normalisation layers are frequently placed after the ReLU layer, which is regarded as an activation function. Because the activation function and the final convolution conceal their inputs and outputs, they are comparable to hidden layers.

A typical CNN architecture is represented as follows:
1. Convolutional layers
2. Composite layers
3. ReLU layers
4. Fully connected layers.

Group level:

It serve to reduce the spatial size of built-in features.
Pooling are of two types:
1. Max Pooling
2. Average Pooling.

MAX POOLING: The maximum value of the area of the image covered by Nucleus is returned by MAX POOLING.

AVERAGE: The average of all values in the area of the image covered by the core is what is returned by AVERAGE POOLING.

Figure 3.4 shows an example of aggregation. In practice, max aggregation performs better than average aggregation. I used max compositing to remove noisy triggers and reduce noise while reducing size. As for mean clustering, we only perform dimensionality reduction to remove noise. Therefore, max pooling is a better choice than average pooling.

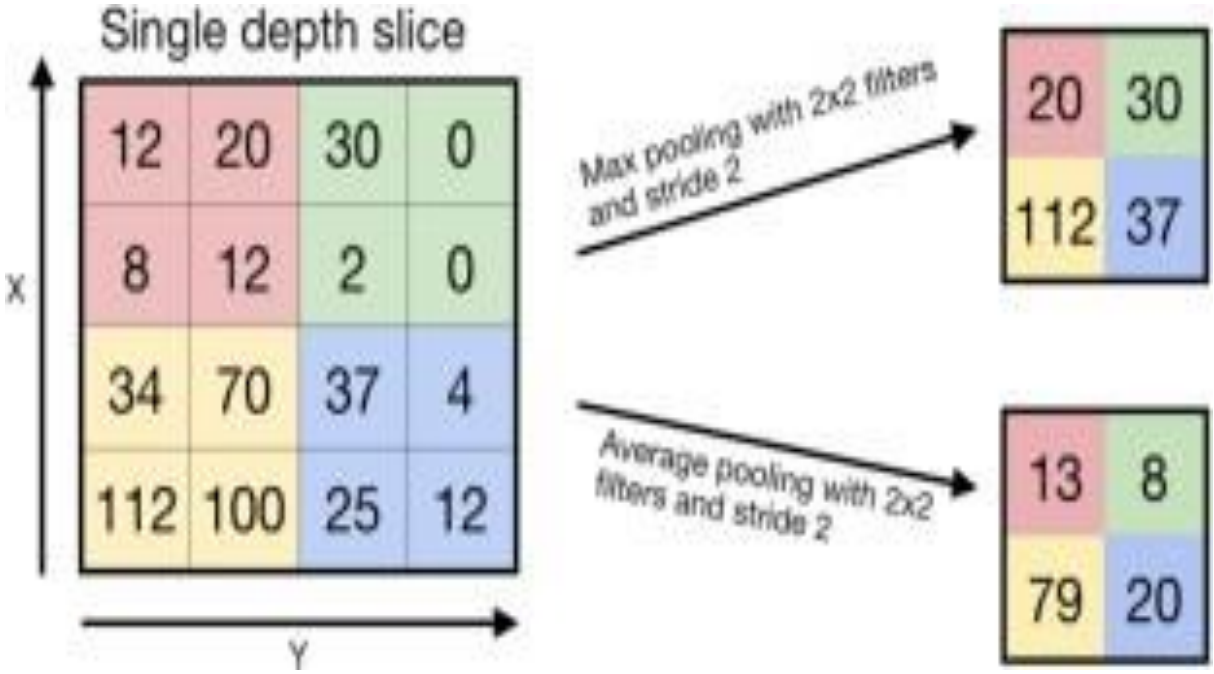

Fig. 3.4. Maximum and average pooling with a size 2*2 and stride 2 filter

ReLU layer:

A linear unit that has been changed and makes use of the unsaturated activation function. To zero out all negative values and replace them with ones in the activation map, use the formula f(x) = max(0,x). Without changing the CONV layer's receptive region, this technique enhances the decision function and total network nonlinearity. There are other ways to improve the nonlinear properties such as

- Saturated hyperbolic tangent f(x) = tanh(x), f(x) = |tanh(x)|
- Sigmoid function $\sigma(x)$ = (1 + e^ −x) ^−1.

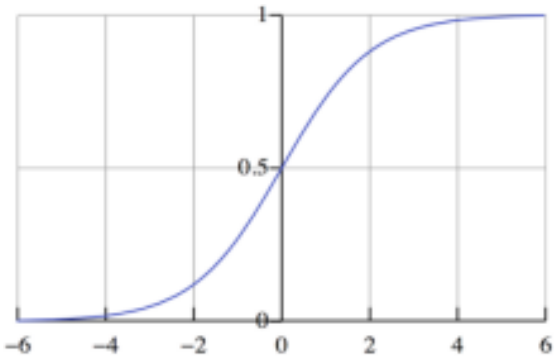

Fig. 3.5 The Sigmoid Graph

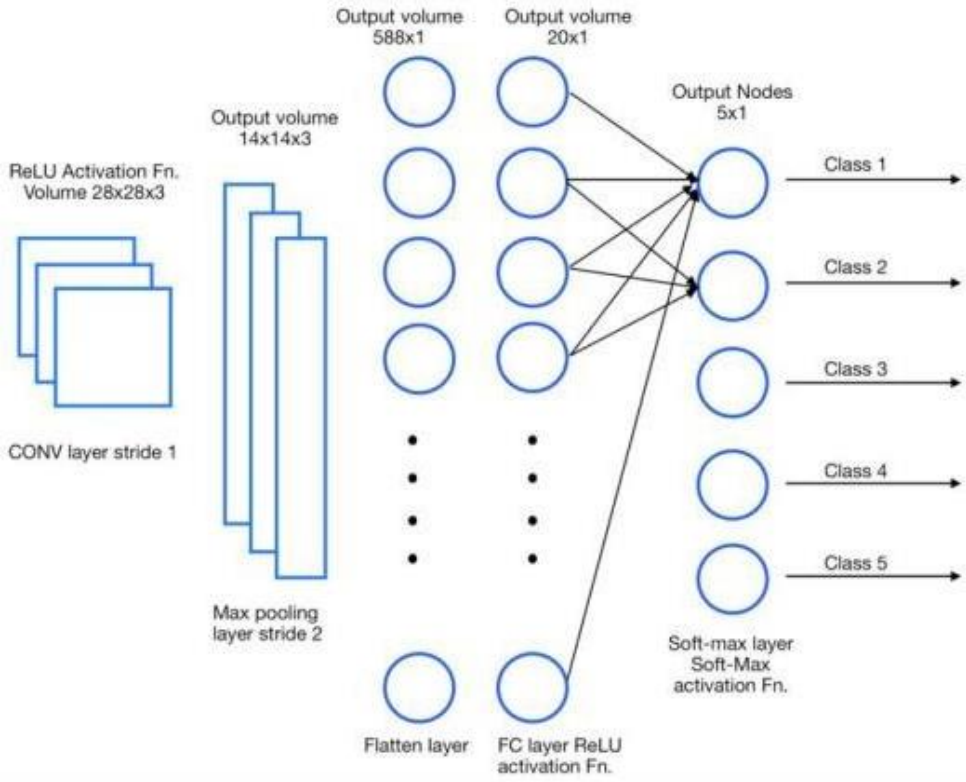

Fig. 3.6 Fully Connected Layer (FC layer)

Its quicker training pace has helped it become more well-liked than other aspects. A graph of ReLU is shown in Figure 3.5 and fully connected layer in fig 3.6.

ARCHITECUTE

The implemented CNN model has two stacks consisting of two and one convolutional layer respectively, followed by two fully convolutional layers. We used Max Pooling after each stack to remove noisy triggers and denoise while reducing sizeIn order to

avoid overfitting, the Dropout layer is additionally applied following each stack. The flowchart of the developed architecture is illustrated in Figure 3.7.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 21, 21, 32)	2432
conv2d_1 (Conv2D)	(None, 19, 19, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 9, 9, 64)	0
dropout (Dropout)	(None, 9, 9, 64)	0
conv2d_2 (Conv2D)	(None, 7, 7, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 3, 3, 64)	0
dropout_1 (Dropout)	(None, 3, 3, 64)	0
flatten (Flatten)	(None, 576)	0
dense (Dense)	(None, 256)	147712
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 43)	11051

Total params: 216,619
Trainable params: 216,619
Non-trainable params: 0

Fig. 3.7 Summary of CNN Model

```python
from keras.models import Sequential
from keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Dropout

model = Sequential()

#1st CNN layer
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu', input_shape=X_train.shape[1:]))

#2nd CNN layer
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))

#3rd CNN layer
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Flatten())

#Fully Connected layer 1st layer
model.add(Dense(256, activation='relu'))
model.add(Dropout(rate=0.5))

#Fully Connected layer 2nd layer
model.add(Dense(43, activation='softmax'))

#Compilation of the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Fig. 3.8 Code For CNN Model

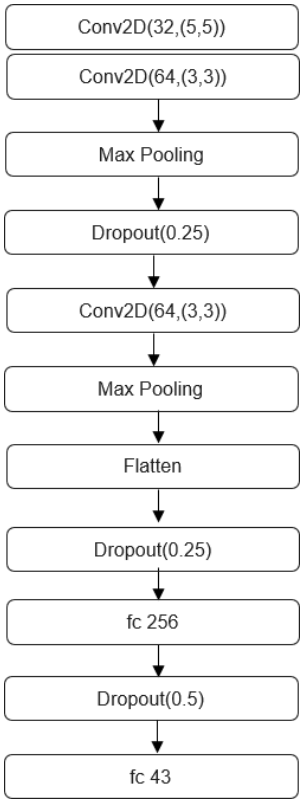

Fig 3.9 An illustration of the proposed CNN architecture (Fully connected layer is abbreviated as fc)

The first two convolutional layers in Figure 3.8 and Figure 3.9. have 32 and 64 convolutional filters with kernel sizes of 5 and 3 respectively. Grid search on different core sizes: 3, 5, 7, 9, 11, 19 and 31 for the convolution layer has been performed. The highest accuracy is achieved with core sizes 5 and 3 respectively. The next and final convolution layer has 64 convolution filters with kernel size of 3. Max pooling is then used to distil the essence of all received feature maps before flattening. A hidden layer with 256 neurons that is fully connected receives its output. The 43 neurons in the final output layer's softmax activation function are used to calculate the probability corresponding to the 43 classes. The majority of this architecture is created using random search for various parameters.

3.5 SIMULATION SET UP

1. Description of Autonomous Mode: This is how our car operates in Autonomous Mode. The Raspberry Pi is mostly utilised to calculate the distance to barriers and process photos of traffic lights that were collected by the camera. With the help of a motor driver L293D, an ultrasonic sensor regulates its movement in accordance with a predetermined programme,

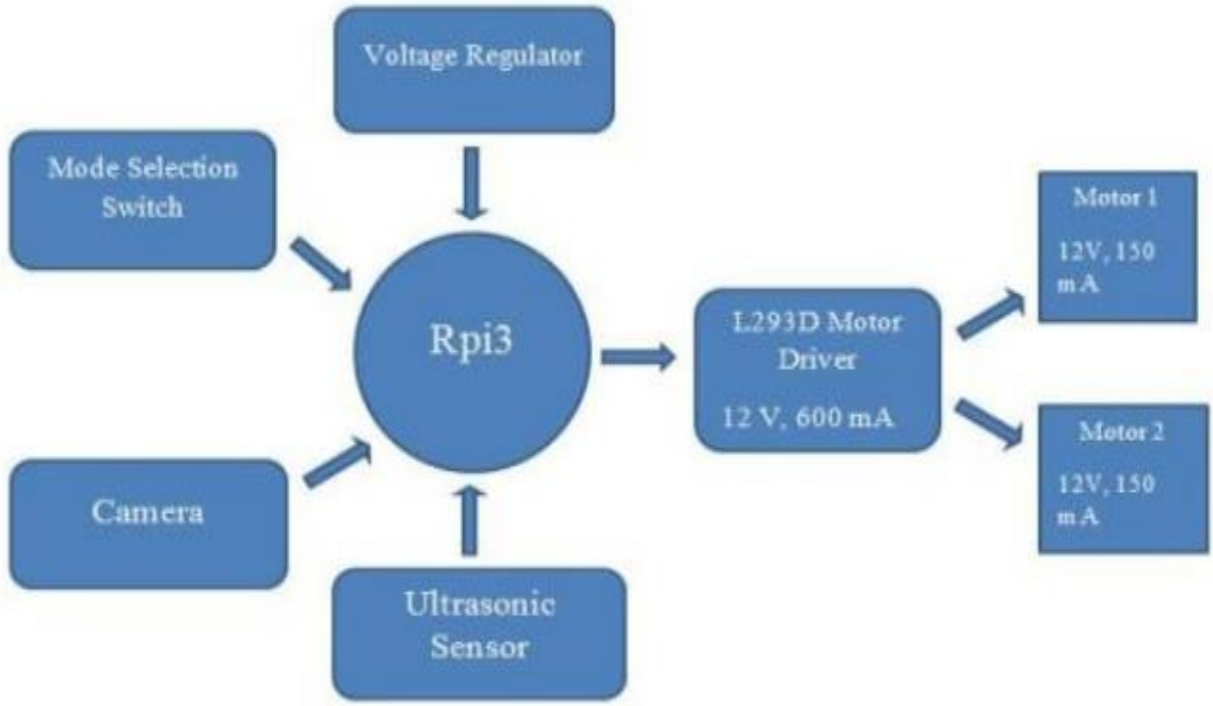

Fig. 3.10 System Architecture: Autonomous Mode

2. You can choose your vehicle's operating mode with this mode selection switch. Both autonomous mode and remote mode are available ways of operation. In autonomous mode, the processor decides how to manoeuvre the vehicle after processing the visuals that were captured. Cameras capture and process traffic images. An ultrasonic sensor measures the distance to obstacles. Camera capture and process traffic image. An ultrasonic sensor measures the distance to obstacles. The car will stop when it is within 2m. A motor driver moves the motor in the direction of the result. It supplies 12 V, 600mA to two motors.

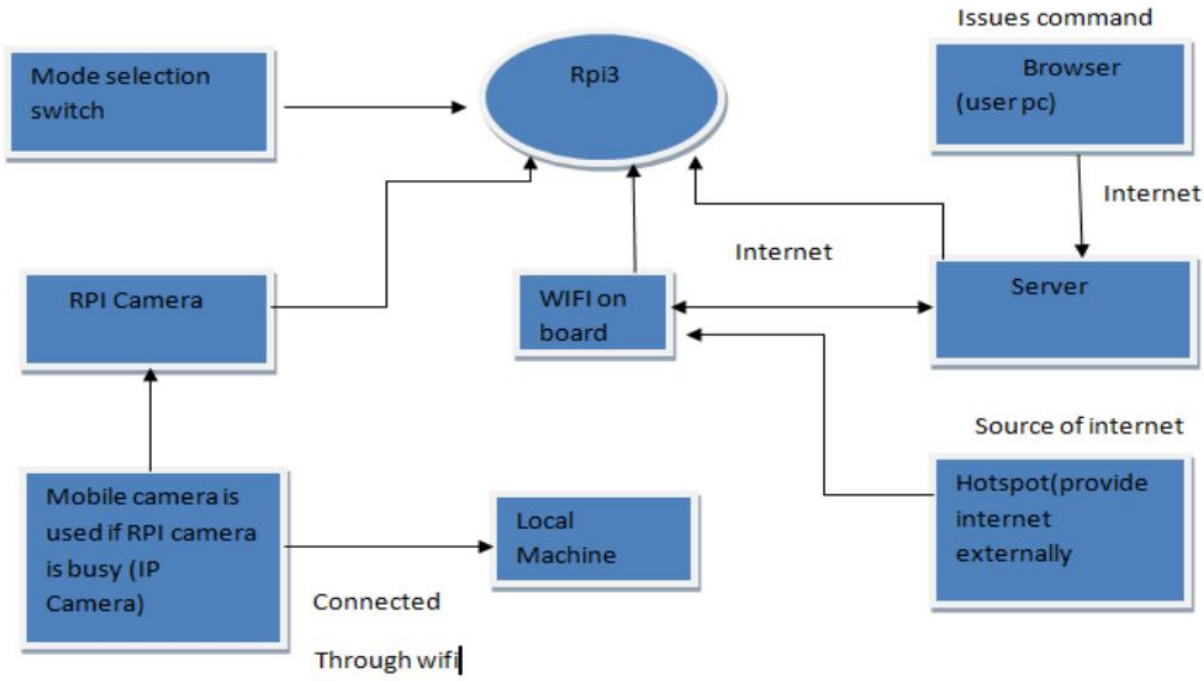

Fig. 3.11 System Architecture: Remote Mode

3. Voltage regulator: Supply 5V to Rpi3 using the Lm7805 as a voltage regulator.

4. Description of Remote Mode: Fig 3.10. In this mode, you can remotely operate the car's functions and movement, with all of the information that has been gathered being sent to a laptop or personal mobile phone via the internet of things. Additionally, you have control over movements to the left, right, forward, backward, and stop.

5. Wifi is also used. It uses the internet to connect to the server on board and gives his WiFi access to the internet via a hotspot. If the car veers off the intended path, the user can adjust the direction of the vehicle as shown in figure 3.11. Cameras attached to the local machine, i.e. mobile and mobile cameras, are used for processing even when the Pi camera is busy.

3.6 HARDWARE DESIGN

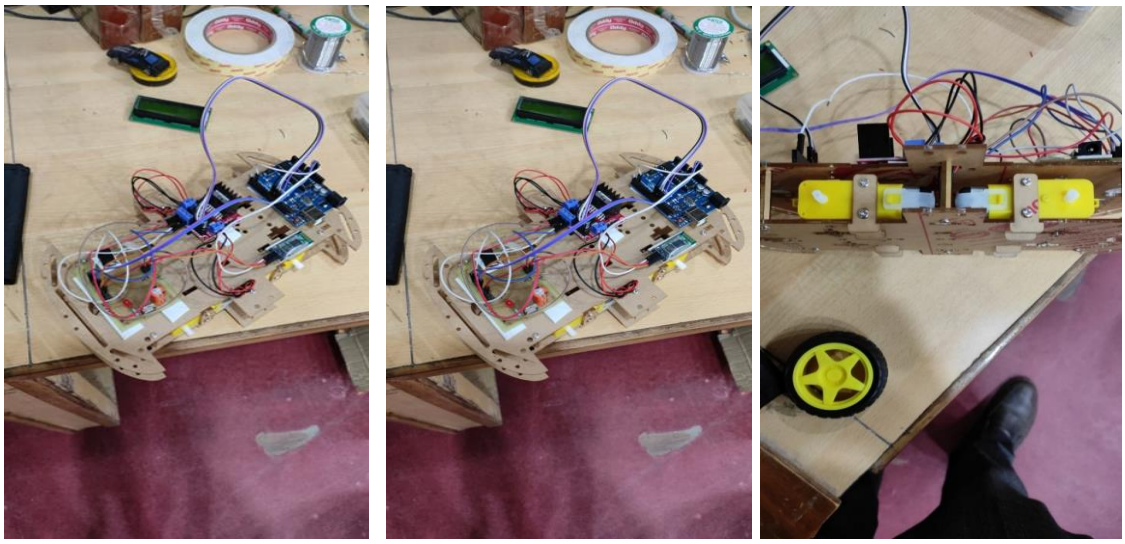

Fig 3.12 Assembling Hardware – Part I

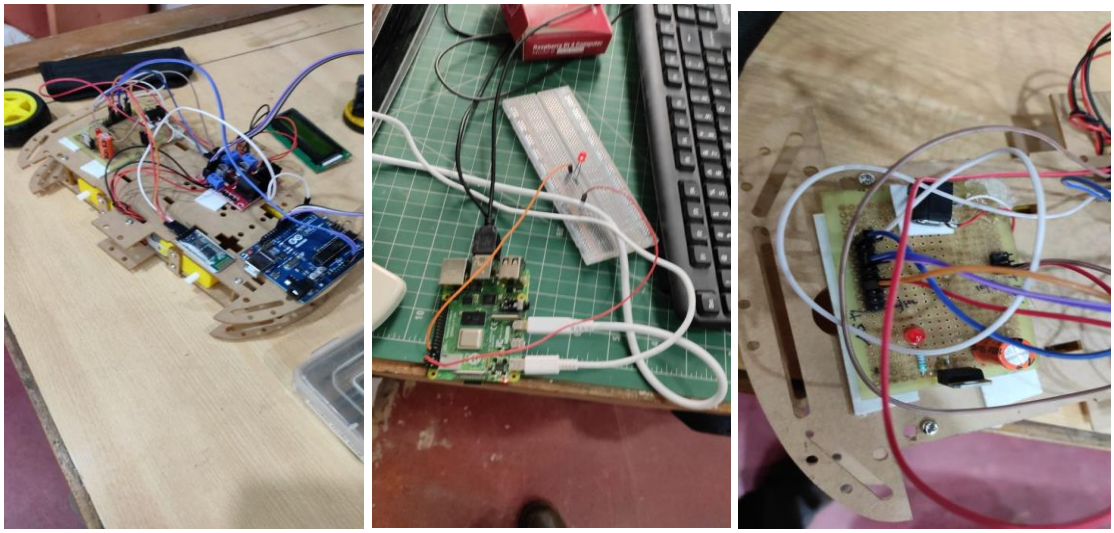

Fig 3.13 Assembling Hardware – Part II

3.7 CONCLUSION

This chapter presented the simulation and hardware results of the proposed model. We also compare the simulation keeping in consideration the different conditions as well. At the end the system is moving and following its path using different sensors.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 OVERVIEW

A multi-agent-based autonomous traffic light controller called fuzzy logic that systemizes In order to combat issues like traffic jams, accidents, and speeding, wireless sensors are used. An adaptive optimization technique based on reinforcement learning is presented by an intelligent traffic light control system, which also introduces simulation and optimization of urban traffic light control. A radio frequency emergency vehicle traffic signal control system is designed to operate when signals from emergency vehicles are received based on radio frequency (RF) transmission. This novel distributed traffic light control design uses a wireless sensor network. Wireless sensors are placed in lanes of traffic entering and exiting intersections. The latest methods are used to regulate the traffic flow sequence in an adaptive traffic control system built on the new Wireless Sensor Network (WSN) traffic infrastructure. These technologies dynamically adapt to traffic conditions at both single and multiple intersections. The system took advantage of the emergence of new technologies to make signal control smarter. They use communication, sensor networks, and more advanced algorithms to configure traffic lights.

4.2 SIMULATION RESULTS

Step 1: IMAGE ACQUISITION AND VISUALIZATION

Dataset 1: GTSRB - German Traffic Sign Recognition Benchmark

The dataset is taken from the Kaggle repository.

Link:https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

Here is the dataset for classifying 43 different classes (Figure 2.1.4) of traffic signs with total images of 39209. In addition, there are additional 12631 images in the Test folder. The ***meta*** folder should have 43 different images (ranging from 0 to 42). The detail of each folder is mentioned in Figure.2.1.3 and the types of traffic signs are shown through a bar graph in Figure 2.1.4

```
[277] print("No. of classes : ",len(os.listdir('/content/train')))

      count=0
      for path in os.listdir('/content/train'):
        for file in os.listdir(os.path.join('/content/train',path)):
          count+=1

      print("total no. of images",count)

      No. of classes :  43
      total no. of images 39209
```

Fig. 4.1 Output of Number of classes and images in GTSRB Dataset

```
Number of files in Meta Folder of GTSRB  45
Number of sub-folders in  Meta Folder of GTSRB  0
Number of files in Test Folder of GTSRB  12631
Number of sub-folders in Test Folder of GTSRB  0
Number of images in Train Folder of GTSRB  0
Number of classes in Train Folder of GTSRB  43
Number of files in  6  class of Train Folder of GTSRB  420
Number of files in  4  class of Train Folder of GTSRB  1980
Number of files in  28  class of Train Folder of GTSRB  540
Number of files in  20  class of Train Folder of GTSRB  360
Number of files in  1  class of Train Folder of GTSRB  2220
Number of files in  0  class of Train Folder of GTSRB  210
Number of files in  25  class of Train Folder of GTSRB  1500
Number of files in  42  class of Train Folder of GTSRB  240
Number of files in  32  class of Train Folder of GTSRB  240
Number of files in  12  class of Train Folder of GTSRB  2100
Number of files in  30  class of Train Folder of GTSRB  450
Number of files in  18  class of Train Folder of GTSRB  1200
Number of files in  21  class of Train Folder of GTSRB  330
Number of files in  36  class of Train Folder of GTSRB  390
Number of files in  34  class of Train Folder of GTSRB  420
Number of files in  39  class of Train Folder of GTSRB  300
Number of files in  5  class of Train Folder of GTSRB  1860
Number of files in  37  class of Train Folder of GTSRB  210
Number of files in  22  class of Train Folder of GTSRB  390
Number of files in  8  class of Train Folder of GTSRB  1410
Number of files in  31  class of Train Folder of GTSRB  780
Number of files in  35  class of Train Folder of GTSRB  1200
Number of files in  9  class of Train Folder of GTSRB  1470
Number of files in  23  class of Train Folder of GTSRB  510
Number of files in  19  class of Train Folder of GTSRB  210
Number of files in  17  class of Train Folder of GTSRB  1110
Number of files in  26  class of Train Folder of GTSRB  600
Number of files in  7  class of Train Folder of GTSRB  1440
Number of files in  16  class of Train Folder of GTSRB  420
Number of files in  24  class of Train Folder of GTSRB  270
Number of files in  3  class of Train Folder of GTSRB  1410
```

Fig 4.2 Number of images in different classes of GTSRB dataset

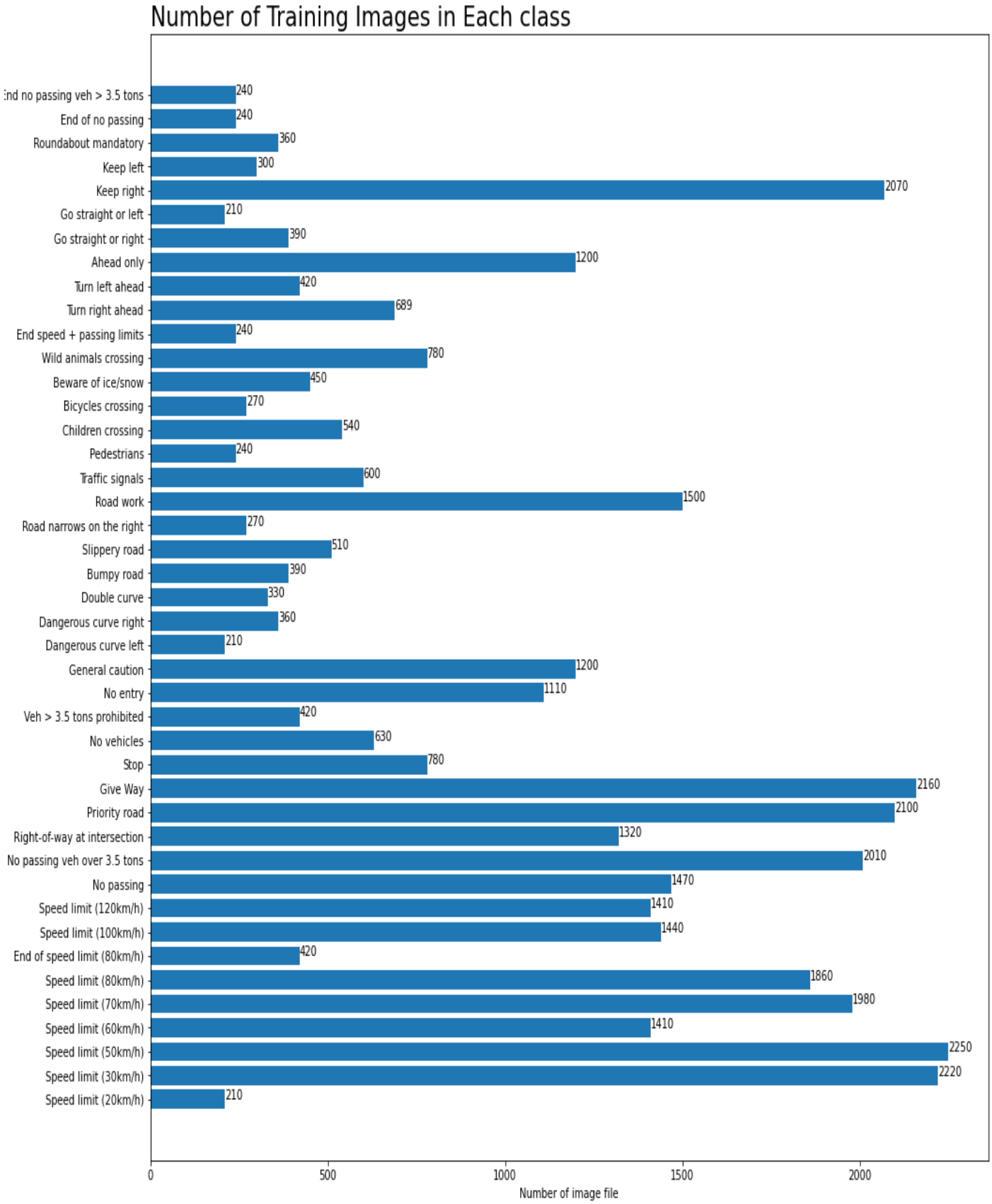
Number of Training Images in Each class
:nd no passing veh > 3.5 tons
End of no passing
Roundabout mandatory
Keep left
Keep right
Go straight or left
Go straight or right
Ahead only
Turn left ahead
Turn right ahead
End speed + passing limits
Wild animals crossing
Beware of ice/snow
Bicycles crossing
Children crossing
Pedestrians
Traffic signals
Road work
Road narrows on the right
Slippery road
Bumpy road
Double curve
Dangerous curve right
Dangerous curve left
General caution
No entry
Veh > 3.5 tons prohibited
No vehicles
Stop
Give Way
Priority road
Right-of-way at intersection
No passing veh over 3.5 tons
No passing
Speed limit (120km/h)
Speed limit (100km/h)
End of speed limit (80km/h)
Speed limit (80km/h)
Speed limit (70km/h)
Speed limit (60km/h)
Speed limit (50km/h)
Speed limit (30km/h)
Speed limit (20km/h)
240
240
360
300
2070
210
390
1200
420
689
240
780
450
270
540
240
600
1500
270
510
390
330
360
210
1200
1110
420
630
780
2160
2100
1320
2010
1470
1410
1440
420
1860
1980
1410
2250
2220
210
0
500
1000
1500
2000
Number of image file

Fig. 4.3 Bar Graph showing the number of image files in each class of Traffic signs of GTSRB dataset

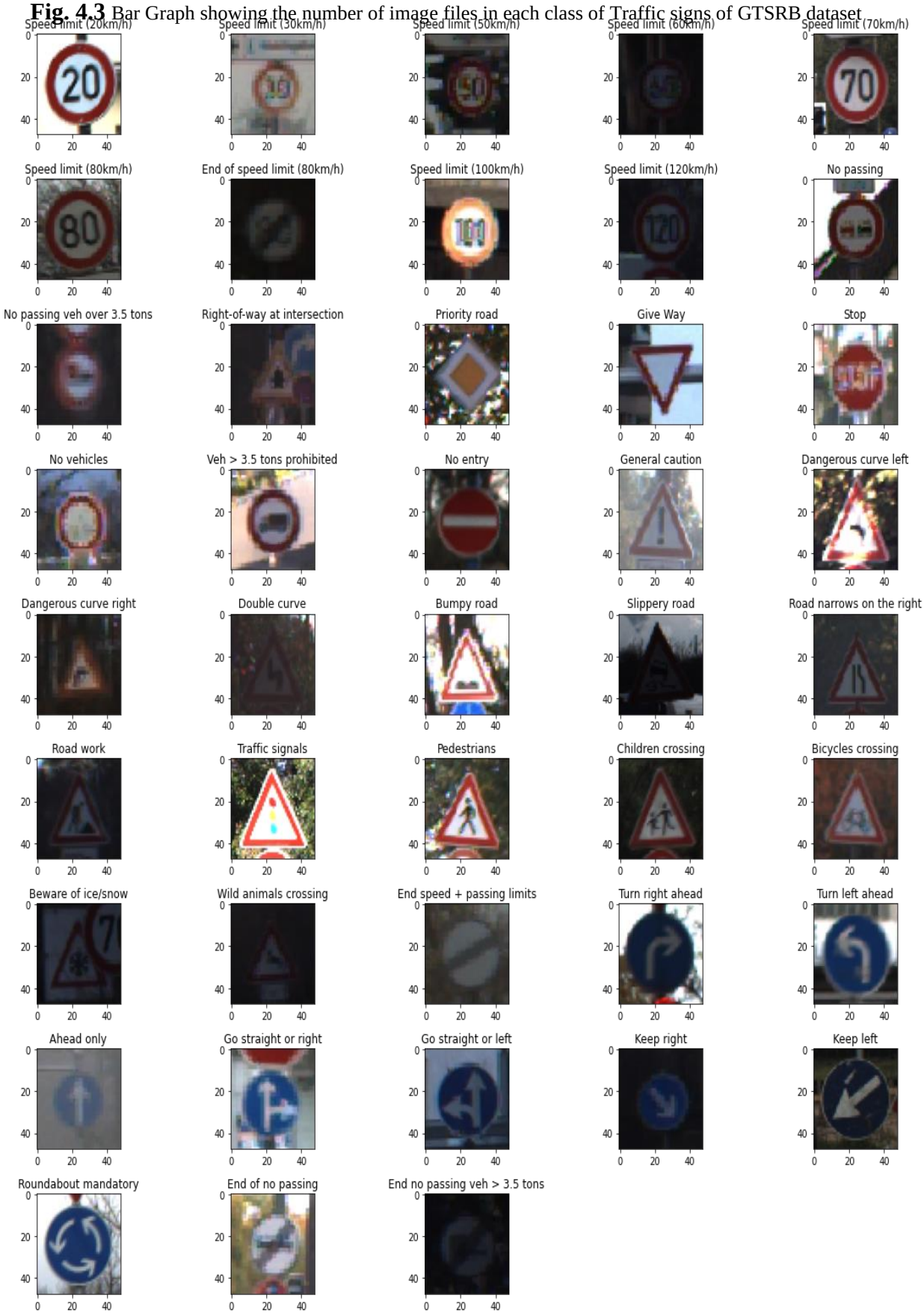

Fig. 4.4 Different classes in GTSRB dataset

DATASET 2: Traffic Sign Dataset – Classification

The dataset is taken from the Kaggle repository.

Link:https://www.kaggle.com/datasets/ahemateja19bec1025/traffic-sign-dataset-classification

Here is the dataset for classifying 57 different classes (Figure 2.1.7) of traffic signs with total images of 4168. In addition, there are additional 1994 images for testing.

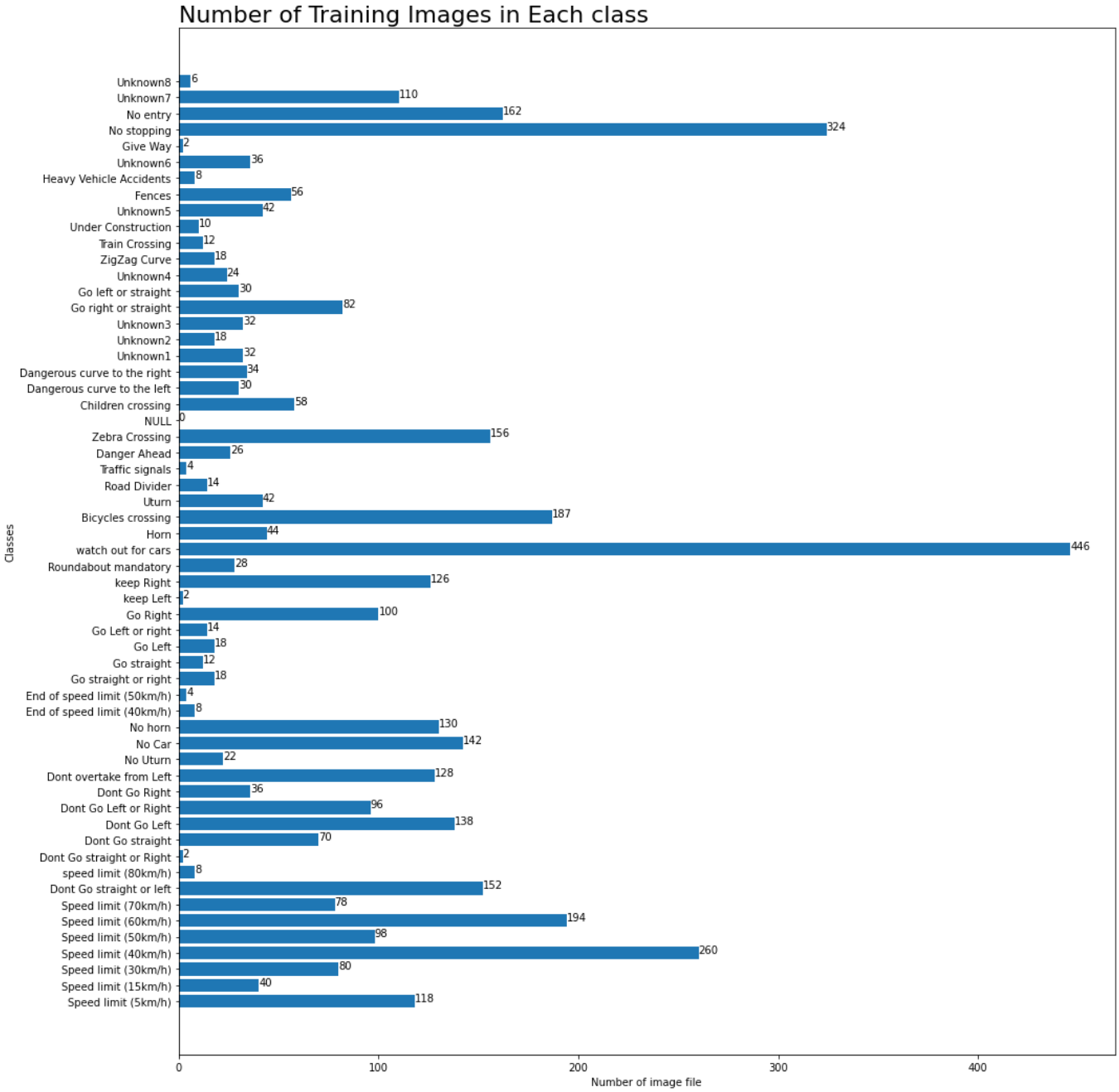

Fig. 4.5 Bar Graph showing the number of image files in each class of Traffic signs of Traffic Sign Dataset

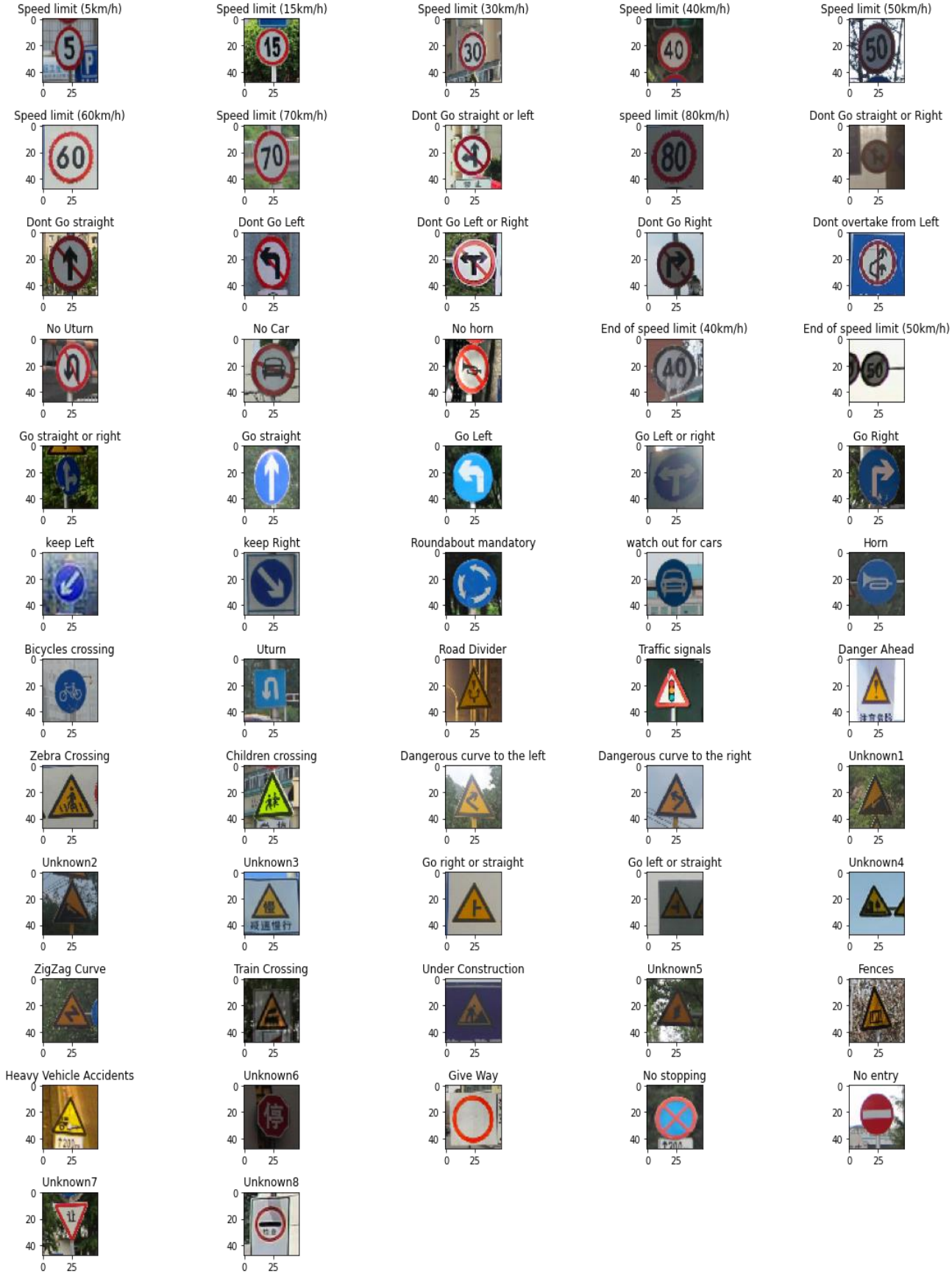

Fig. 4.6 Different classes in Traffic Sign Dataset

Step 2: DATA PREPROCESSING

Upon some initial analysis, it was noticed that there was a high-class imbalance in the dataset. Figure 2.1.4 and Figure 2.1.6 show the data count per class. If the model is trained on this dataset, it poses a problem as this makes the model biased towards the class with higher data frequency. To resolve this, data augmentation was performed. Images from low-frequency classes were picked up and random rotation and brightness variation was performed. Eventually, data was normalized for all classes. Then, standardization was performed on the data to normalize its mean and make it unit variance.

Step 3: Splitting the data into Training, Validation, and Test set

While training a model, it is important to provide random inputs of different classes to the model so that the model can generalize better. That is why we are going to use the sklearn train_test_split() function that will randomly split the data into training and validation sets in the ratio of 80:20. The list of labels ranges from 0 to 42 that represents each category but the neural network needs a different format which is one hot encoding. One hot encoding is a vector representation where all elements of the vector are 0 except one, which has a 1 value. Code for the given in Figure 4.7. Here, we don't need to split the data for the test set an additional folder containing random test images is already present.

```
#Spliting the images into train and validation sets
(X_train,X_val)=Cells[(int)(0.2*len(labels)):],Cells[:(int)(0.2*len(labels))]
X_train = X_train.astype('float32')/255
X_val = X_val.astype('float32')/255
(y_train,y_val)=labels[(int)(0.2*len(labels)):],labels[:(int)(0.2*len(labels))]

#Using one hote encoding for the train and validation labels
from tensorflow.keras.utils import to_categorical
y_train = to_categorical(y_train, 43)
y_val = to_categorical(y_val, 43)
```

Fig. 4.7 Code for Splitting the dataset into Training and Validation Set

Step 4: Build CNN Model

Model Building has been explained in Chapter 3 with flowchart and codes as well as explaining the purpose of each layers as well such as max pooling etc.

Step 5: Apply CNN on Training Dataset

To train our model, we will use the model.fit() method that works well after the successful building of model architecture. All of architectures were trained using an adam optimizer at a learning rate = 0.001 (default learning rate) with batch size = 32 and epochs = 20. However, till epoch 15 we received 98.66 accuracy so we terminated the model. As this is a classification problem, the categorical cross-entropy loss function was used. All of the experiments were performed in python Google Colab using the Keras framework. All the detail about the code and its execution is given in Figure 4.8

```
epochs = 20
history = model.fit(X_train, y_train, batch_size=32, epochs=epochs,validation_data=(X_val, y_val))
```

```
Epoch 1/20
981/981 [==============================] - 84s 84ms/step - loss: 1.2719 - accuracy: 0.6372 - val_loss: 0.2168 - val_accuracy: 0.9466
Epoch 2/20
981/981 [==============================] - 77s 79ms/step - loss: 0.2840 - accuracy: 0.9125 - val_loss: 0.1008 - val_accuracy: 0.9769
Epoch 3/20
981/981 [==============================] - 78s 80ms/step - loss: 0.1670 - accuracy: 0.9486 - val_loss: 0.0847 - val_accuracy: 0.9802
Epoch 4/20
981/981 [==============================] - 78s 80ms/step - loss: 0.1239 - accuracy: 0.9633 - val_loss: 0.0466 - val_accuracy: 0.9880
Epoch 5/20
981/981 [==============================] - 79s 81ms/step - loss: 0.1042 - accuracy: 0.9692 - val_loss: 0.0350 - val_accuracy: 0.9912
Epoch 6/20
981/981 [==============================] - 76s 78ms/step - loss: 0.0895 - accuracy: 0.9715 - val_loss: 0.0342 - val_accuracy: 0.9932
Epoch 7/20
981/981 [==============================] - 76s 77ms/step - loss: 0.0785 - accuracy: 0.9767 - val_loss: 0.0293 - val_accuracy: 0.9920
Epoch 8/20
981/981 [==============================] - 76s 78ms/step - loss: 0.0668 - accuracy: 0.9795 - val_loss: 0.0238 - val_accuracy: 0.9935
Epoch 9/20
981/981 [==============================] - 78s 79ms/step - loss: 0.0633 - accuracy: 0.9808 - val_loss: 0.0293 - val_accuracy: 0.9921
Epoch 10/20
981/981 [==============================] - 77s 78ms/step - loss: 0.0597 - accuracy: 0.9819 - val_loss: 0.0242 - val_accuracy: 0.9941
Epoch 11/20
981/981 [==============================] - 77s 79ms/step - loss: 0.0590 - accuracy: 0.9825 - val_loss: 0.0233 - val_accuracy: 0.9948
Epoch 12/20
981/981 [==============================] - 77s 78ms/step - loss: 0.0543 - accuracy: 0.9841 - val_loss: 0.0255 - val_accuracy: 0.9946
Epoch 13/20
981/981 [==============================] - 78s 79ms/step - loss: 0.0530 - accuracy: 0.9843 - val_loss: 0.0248 - val_accuracy: 0.9932
Epoch 14/20
981/981 [==============================] - 78s 79ms/step - loss: 0.0539 - accuracy: 0.9844 - val_loss: 0.0210 - val_accuracy: 0.9949
Epoch 15/20
981/981 [==============================] - 79s 80ms/step - loss: 0.0425 - accuracy: 0.9866 - val_loss: 0.0264 - val_accuracy: 0.9940
Epoch 16/20
972/981 [============================>.] - ETA: 0s - loss: 0.0557 - accuracy: 0.9846
```

Fig. 4.8 Training the CNN model in the Train dataset

Step 6: Validating the result

With the help of matplotlib functions, we will plot the graph of training and validation accuracy as shown in Figure 4.9

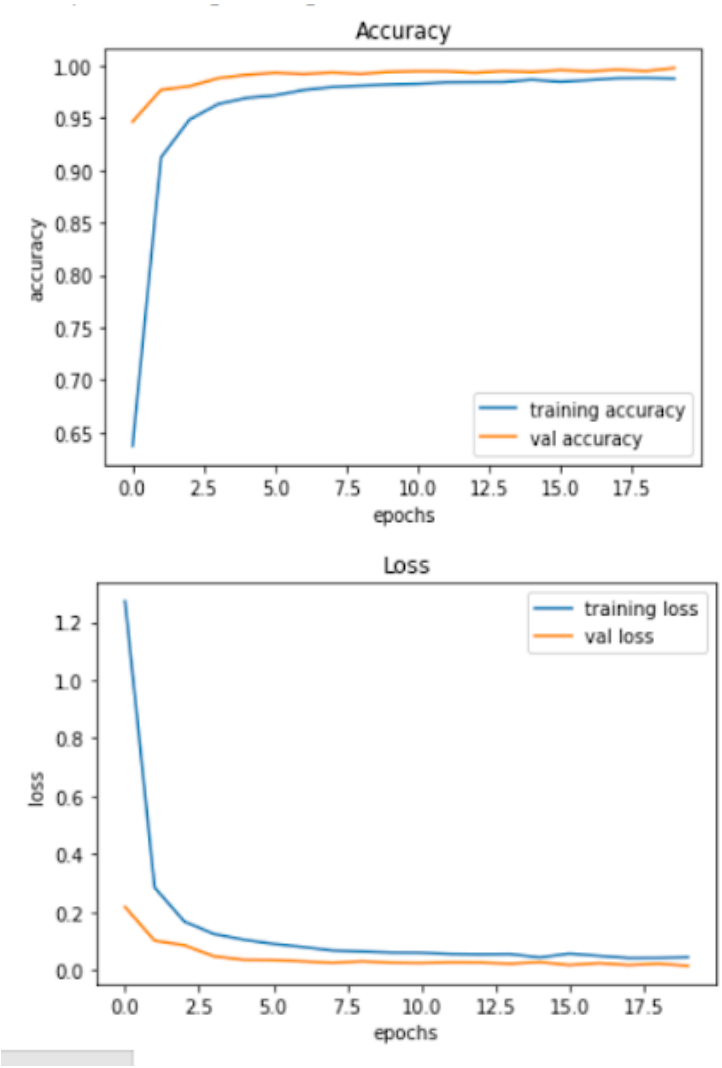

Fig. 4.9 Validation Loss and Accuracy Curve

Step 6: Make Prediction on Test Data and find its Accuracy

A folder named “test” is available in our dataset. Inside that, we got the main working comma-separated file called “test.csv”. It comprises two things, the image paths, and their respective class labels. We can use the pandas’ python library to extract the image path with corresponding labels. Total 12630 test images are there for training, Next, we need to resize our images to 25×25 pixels to predict the model and create a numpy array filled with image data. To understand how the model predicts the actual labels, we need to import accuracy_score from the sklearn. Metrics as shown in Fig. 4.10

```
from PIL import Image
from sklearn.metrics import accuracy_score
import pandas as pd
y_test = pd.read_csv('/content/Test.csv')

labels = y_test['ClassId'].values
imgs = y_test['Path'].values

data1=[]

for img in imgs:
  image = Image.open(img)
  image = image.resize((25,25))
  data1.append(np.array(image))

X_test = np.array(data1)
```

```
len(data1)
```

```
12630
```

```
predict_x=loaded_model.predict(X_test)
pred=np.argmax(predict_x,axis=1)
#pred = loaded_model.predict_classes(X_test)
```

```
#Accuracy with the test data
from sklearn.metrics import accuracy_score
accuracy_score(labels,pred)
```

```
0.7482977038796517
```

Fig. 4.10 Accuracy of Test Data and Code

4.3 HARDWARE RESULTS

The hardware prototype of the proposed idea of Autonomous vehicle with real time traffic signal detection is resulted. We used Pi camera to capture traffic signs in real time. Then the image is processed though the trained CNN model. And the vehicle operates according. The resultant model is shown in fig 4.11 and figure 4.12.

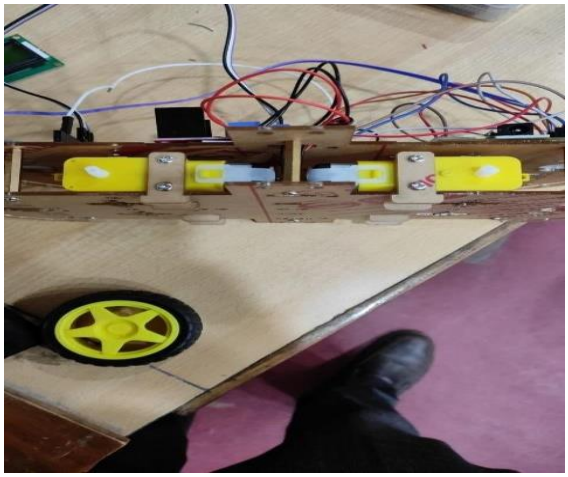

Fig. 4.11 Motor Connected to Circuit

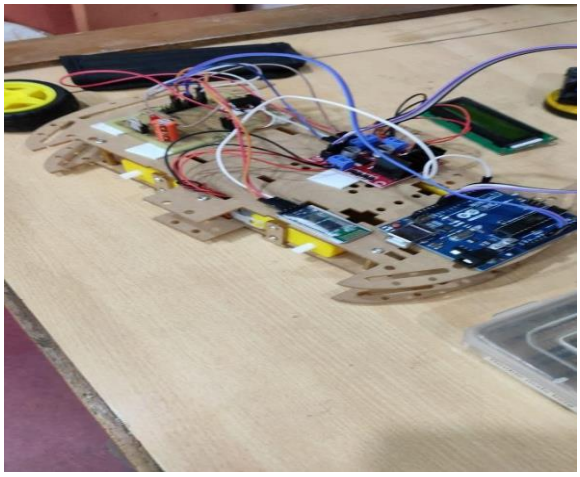

Fig. 4.12 Front View of Circuit

4.4 INFERENCES DRAWN

Using LPR data, a proposed integrated technique for signal timing parameter inference at junctions. Phase sequence inference, cycle length and cycle breakpoint inference, and green phase duration for each phase can all be categorised under the three categories of the proposed method. The efficiency and robustness of our method, which combines traffic prior knowledge (information on phase weight and average phase duration) and mathematical-statistical modelling techniques, on the signal timing parameters inferences are expected to be significantly improved, compared to the existing technologies, because the LPR data can provide comprehensive spatio-temporal information, because extensive spatio-temporal information can be provided by LPR data. A suggested integrated method for signal timing parameter inference at junctions using LPR data. Under the three categories of the proposed method, phase sequence inference, cycle length and cycle breakpoint inference, and green phase duration for each phase can all be grouped together. Because the LPR data can provide comprehensive spatio-temporal information, the efficiency and robustness of our method, which combines traffic prior knowledge (information on phase weight and average phase duration) and mathematical statistical modelling techniques, on the signal timing parameters inferences are expected to be significantly improved, compared to the existing technologies.

4.5 JUSTIFICATION OF OBJECTIVE ACHIEVED

Using a dataset of our traffic signals, this study aims to evaluate the TSR's accuracy and speed. So, to evaluate the performance of the YOLO algorithm family, we employed its most recent edition, YOLOv5. Additionally, we choose from the YOLOv5 and SSD models the one that will work the best for the TSR. In this study, we make use of a highly specialised computing skill. We also examine and contrast the two models' performance using our assessment measures.

4.6 CONCLUSION

This chapter presented the simulation and hardware results of the proposed model. We also compared the model with different result. It can be observed from Fig that the accuracy we received is around 74%. But if we compare it with the training accuracy, it happens to be 98.66%. From this, we can conclude that on the training dataset the CNN model was overfitted due to which on the test dataset we got 74% accuracy. We need to further perform data

augmentation techniques and make changes in CNN network parameters (learning rate, epoch, kernel size; etc) and layers to get better accuracy on test data.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSION

By capturing streaming data in the IoT cloud, an RC car model is integrated with traffic sign detection. Python has been used to do traffic sign detection using CNN and Keras, and the Raspberry Pi 3 has been used. The system's major goal is to improve automation because doing so will reduce human error and boost speed, efficiency, and dependability. All of the techniques and algorithms covered in this report have been applied to RC cars with positive outcomes, and each technique, algorithm, and component used has been described in great detail.

The major conclusion from this project:

1. This project if applied on the actual vehicle will certainly be able to reduce accidents on road.
2. The project will also help traffic regulation as well as controlling and regulating the crowd on the roads
3. It will help drivers while driving and keeping them mindful of the surrounding.

5.2 SCOPE OF FUTURE WORK

At the successful completion of project, the scope for further work has been identified as follows :

1. In the future, a high resolution camera can be used to accelerate and improve traffic sign detection.
2. The work can be tested with advanced deep learning algorithms, such as You Only Look Once (YOLO).
3. Automated traffic sign violation checks for a variety of additional traffic signs can be included in remote-controlled automobiles to add more automation.

CHAPTER 6

PROJECT METRICS

6.1 CHALLENGES FACED

There were numerous difficulties with this project. It was initially challenging to find a sizable data set that would be the greatest fit for our project. Then with the help of our mentor, we decided on the various limited parameters on which we will train our model and hence select a suitable data set of Indian traffic signals, firstly we were considering the entire data set of traffic signals which included traffic signs all over the world. Initially, We were also facing difficulties in choosing suitable hardware . Panel suggested us use a raspberry pi which will be the best fit CPU for our project, After deciding the type of sensors, we faced difficulty in increasing the accuracy of our trained model which was resolved later by increasing the size of our data set.

When looking at dataset 2, we see that the histogram is incorrect. After careful analysis, we found that type 18 and type 19 were the same as type 3 and 4. But after visual inspection, we concluded that the name of the sign was misspelled. Likewise, Grade 30 and Grade 36 are the same. So we merged th 2 layers into 1. The issue can be easily observed through figure 5.1 and figure 5.2

6.2 RELEVANT SUBJECTS

Table 6.1 List of components and cost analysis

S.NO	SUBJECT CODE	SUBJECT NAME
1.	UEE504	Power Electronics
2.	UEI201	Analog Electronic Device and Circuits
3.	UEI301	Digital Electronics
4.	UEE601	Machine Learning techniques

6.3 INTERDISCIPLINARY ASPECT

Various factors are covered by our project. It includes knowledge of agriculture, fundamental electricity, and power electronics. Software called Multisim has also been studied. Our investigation of alternative energy sources was prompted by the deployment of solar panels.

6.4 COMPONENTS AND COST ANALYSIS

The following table gives details on the various components used as well as their specifications in Table 6.2

Table 6.2 List of components and cost analysis

S.No.	COMPONENTS	QTY.	LINK	COST
1.	L298N Motor Driver	2	https://robu.in/product/l298n-2abased-motor-driver-module-goodquality/	270.00
2.	Ultrasonic Sensor	1	https://robu.in/product/us-100ultrasonic-sensor-distancemeasuring-module-temperaturecompensation/	419.00
3.	Male to female jumper (10 cm)	1	https://robu.in/product/male-tofemale-jumper-wires-40-pcs-10cm/	55.00
4.	Male to female jumper (30 cm)	2	https://robu.in/product/male-tofemale-jumper-wires-40-pin-30cm/	130.00
5.	SpeedStudio Grove speaker	1	https://robu.in/product/grovespeaker/	699.00
6.	NodeMCU	1	https://robu.in/product/nodemcuesp8266/	649.00
7.	Breadboard	1	https://robu.in/product/transparent830-points-solderless-breadboard/	108.00

8 .	Battery Pack	1	https://robu.in/product/orange-isr18650-li-ion-1300mah-7-4v-2s1pprotected-battery-pack-15c/	549.00
9.	Li ion Battery	4	https://robu.in/product/orange-isr18650-2000mah-10c-lithium-ionbattery/	712.00
10.	Base with wheel	1	https://robu.in/product/longerversion-4-wd-double-layer-smartcar-chassis/	599.00
11.	Shock absorber	1	https://robu.in/product/98mmmetal-front-rear-shock-absorber-forrc-car/	549.00
12.	Li battery protection board	1	https://robu.in/product/5s-25a18650-lithium-battery-protectionboard/	399.00
13.	Remote	1	https://robu.in/product/infrared-irwireless-remote-control-module-kitarduino/	129.00
13.	Raspberry Pi	1	https://robu.in/product/raspberrypi-4-model-b-2gb-starter-kit/	5930.00
14.	Pi camera	1	https://robu.in/product/raspberrypi-camera-v2/	1810.00
15.	Bluetooth module	1	https://robu.in/product/hc-05-6pinbluetooth-module-with-button/	622.00
16.	Arduino	1	https://robu.in/product/arduinouno-r3/	1079.00
	TAX			50.00
	TOTAL			**14758.00**

6.5 BRIEF ANALYTICAL NOTES

- **What sources of information did your group explore to arrive at the list of problems which could be taken as the project?**

  The team was aware of the significance of the capstone project. Before beginning the study, the group conducted a literature review on the subject. In order to learn more about the current state of farming practises in India, we read the IEEE paper. We took help from our mentor and understood the various limitations we have and hence modified our project to deliver the best possible results.

- **What analytical, computational and/or experimental methods did your project group use to obtain solutions to the problems in the project?**

  The analytical components were completed in a lab setting. The project was created and ran successfully in the lab. Our calculations' results have been approximated by adjusting the values. The simulations are all carried out using Tensorflow.

- **Did the project demand demonstration of knowledge of fundamentals, scientific and/or engineering principles? If yes, & when did you apply?**

  Yes, the project allowed us the chance to learn about a variety of subjects outside of our academic curriculum, which made the whole thing even more exciting. We have discussed the fundamentals of CNN, deep learning, and machine learning. These topics have also assisted us in selecting the elements and comprehending the fundamental operations of our project.

- **How did your group shares responsibility and communicate the information of schedule with others in team to coordinate design and manufacturing dependencies?**

  Four BE electrical engineering students make up our group, which is in the same lecture section. This makes it easier for us to coordinate well. We have been meeting in the library to discuss our challenges and all project-related issues. We also meet on Skype every fourth day of the month and in a Whatsapp group. Our twice-weekly meetings with our mentor have been really helpful in getting all of our questions answered. We have also benefited from the departmental monitoring and mentoring schedule.

- **What resources did you use to learn new materials not taught in class for the course of the project?**

  The TensorFlow software has been used for simulation purposes, Which was basically to train our model. We were never tough in class how to collect a highly efficient data set and train it for a specific optimised output. It was also the first time we used both our hardware and software knowledge to give value added results.

- **Does the project make you appreciate the need to solve problems in real life using engineering?**

  Engineering is more than just academic research; it also involves applying concepts to actual situations. We have been able to analyse the real issues and identify fresh approaches to solving them thanks to this initiative. Perhaps our modest contribution will be helpful and make people's lives easier.

- **List the engineering equipment, hardware which this project has made you able to use effectively.**

  Some of the equipment that are used are –

  Soldering station Python libraries

- **List the software tools used to analyze engineering problems in the project.**

  Tenserflow

6.6 TEAM ASSESSMENT MATRIX

Table 6.3 Team Assessment Matrix

EVALUATION BY	**EVALUATION OF**	VINAY	YASHVARDHAN	AYUSH	AKANKSHA	GURSIMAR
VINAY		4	4	3.5	4	4
YASHVARDHAN		4	4	3.5	4	3.5
AYUSH		3.5	3.5	4	3.5	4
AKANKSHA		3.5	3.5	3.5	4	2

GURSIMAR		4	4	4	4	2

6.7 WORK SCHEDULE (GANTT CHARTS)

Akanksha Singh , Vinay Jain - Collection and training of data set to deliver a high efficient model , paper work handling .

Yahvardhan Pachauri , Ayush Vashisth - prior research for the project, designing and installation of the final hardware model , paper work handling

Fig. 6.1 to 6.6 shows the Gantt chart for the work distribution and its execution for the group member.

6.7.1 GANTT CHARTS OF GROUP

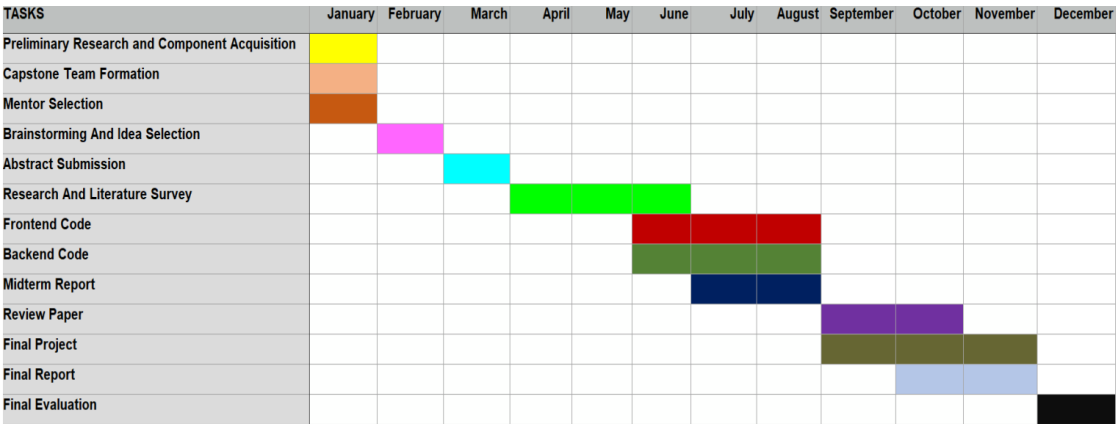

Fig 6.1 Gantt Chart of Group

6.7.2 GANTT CHARTS OF INDIVIDUAL

6.7.2.1 GANTT CHART OF Vinay Kumar Jain

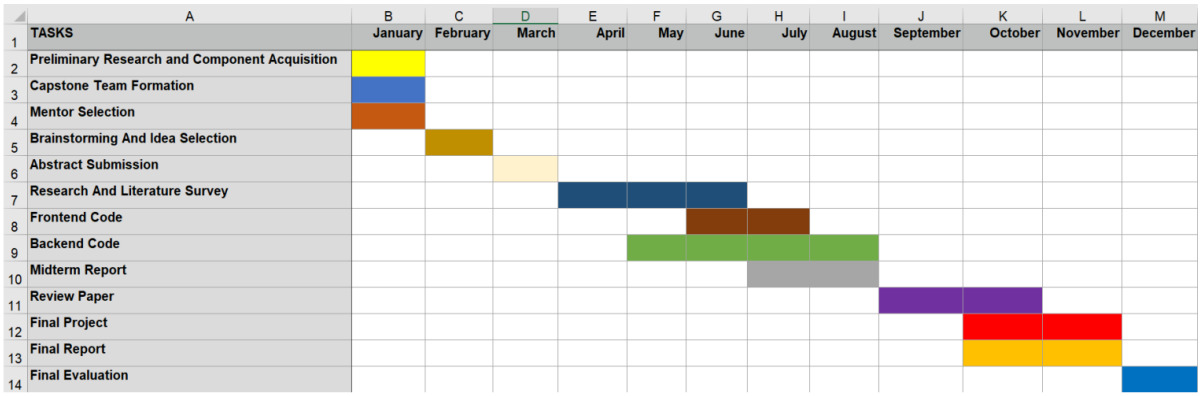

Fig 6.2 Gantt Chart of Vinay Kumar Jain

6.7.2.2. GANTT CHART OF Ayush Vashisth

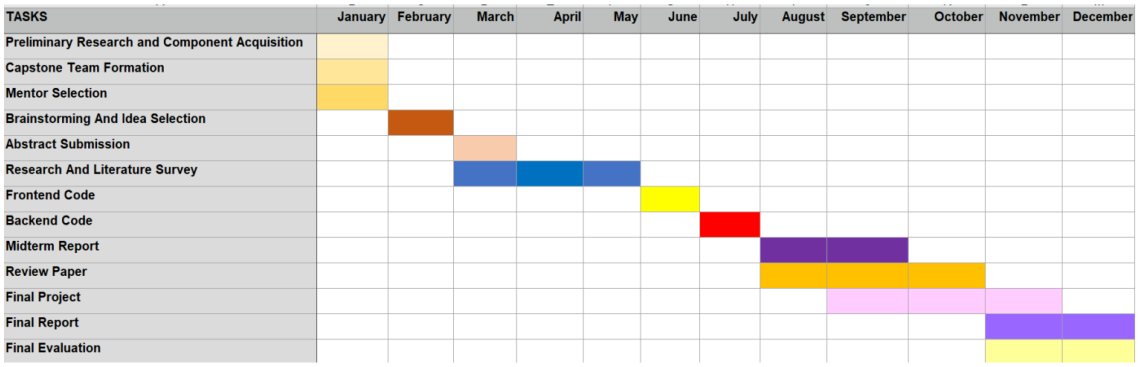

Fig 6.3 Gantt Chart of Ayush Vashisth

6.7.2.3. GANTT CHART OF Akanksha Singh

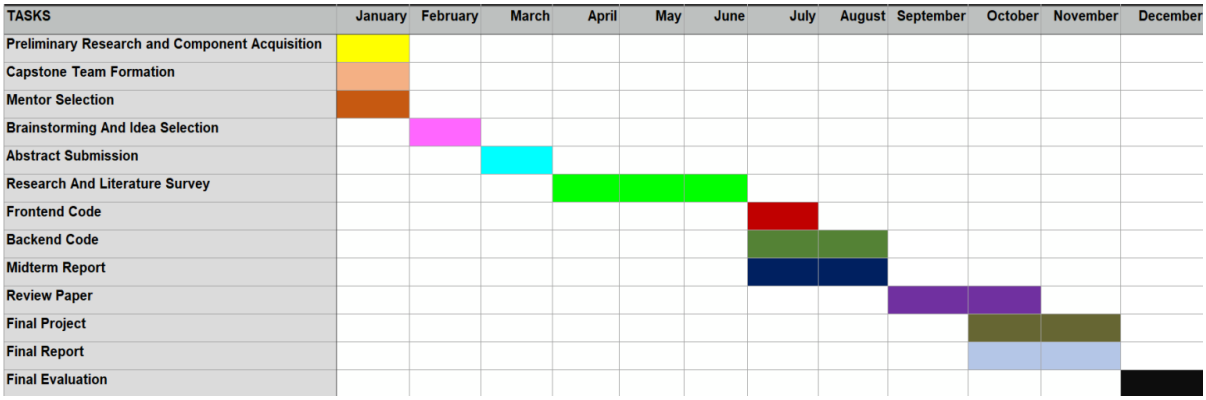

Fig 6.4 Gantt Chart of Akanksha Singh

6.7.2.4. GANTT CHART OF Yashvardhan Pachauri

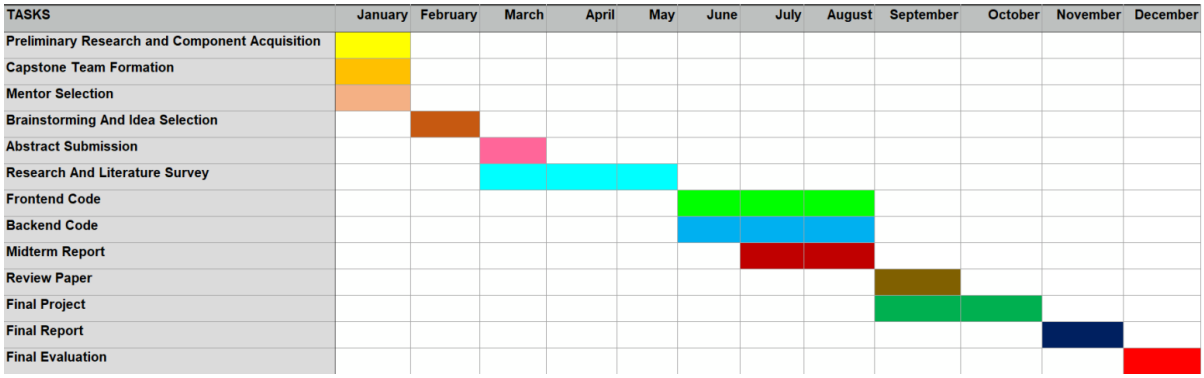

Fig 6.5 Gantt Chart of Yashvardhan Pachauri

6.7.2.5. GANTT CHART OF Gursimar Singh Nagpal

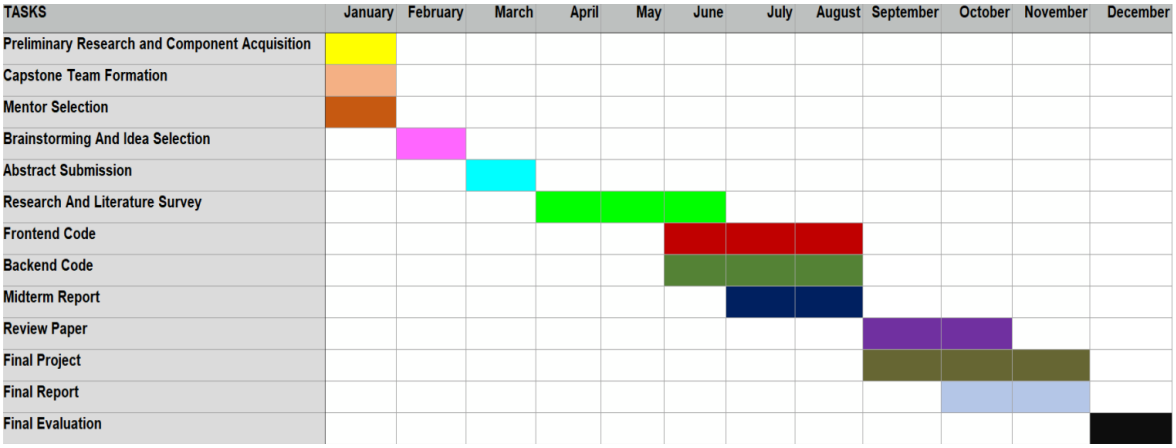

Fig 6.6 Gantt Chart of Gursimar Singh Nagpal

6.8 STUDENT OUTCOME (A-K) MAPPING

Table 6.4 Student Outcome (SOS) mapping

A2	Demonstrate and apply knowledge of fundamentals, scientific and/or engineering principles towards solving engineering problems.	This project illustrates the use of DL, sensors, raspberry- pi .
B1	Identify the constraints, assumptions and models for the experiments.	This project works under the assumption that weather will not play a major role for disturbing our model in real life.
D1	Share responsibility and information schedule with others in team.	Every individual team member is working on different part of project and discuss the status of their work
E1	Classify information to identify engineering problems.	We have acknowledged the practical problems and their corresponding solution in the need analysis of project
E3	Use analytical, computational and/or experimental methods to obtain solutions.	Calculation for data set have been done.

H1	Aware of societal and global changes due to engineering innovations.	No harm was done on the environment , all components can recycled
I1	Able to use resources to adopt new technologies not included in curriculum.	We have used Tenserflow software for simulations.
K3	Able to analyze engineering problems using software tools.	The simulation model is developed and results have been discussed.

6.9 REPORT PROGRESS

Thapar Institute of Engineering and Technology, Patiala
Electrical and Instrumentation Engineering Department
Progress Report Form
Capstone Project (Electrical Engineering)

Group No: 1
Name of the Supervisor: S.K.Jain & Vishal Srivastava
Project Title: Autonomous Vehicle Control Through Traffic Signs

First/Second/Final Evaluation (Dated: ..17/..5./...2022) **Code: UEE795**

Date	Work done/discussed	Roll No.	Student name	Self-Evaluation	Supervisor Evaluation	Supervisor's Sign with remarks
24/4/22	Everyone read different Research papers on traffic sign detection In both software and hardware part and contributed their ideas	101904104	Akanksha Singh	5	5	Work hard
		101904017	Vinay Kumar	5	5	
		101904095	Yashvardhan Pachauri	5	4.5	
		101904020	Ayush Vashisth	5	4.5	
		101904114	Gursimar Singh	5	4.5	
30/4/22	Searched for the dataset and found the dataset on Kaggle and Indepth research of raspberry pi	101904104	Akanksha Singh	5	5	Load data & understand its limitation
		101904017	Vinay Kumar	5	4.5	
		101904095	Yashvardhan Pachauri	5	4.5	
		101904020	Ayush Vashisth	5	4.5	
		101904114	Gursimar Singh	5	4.5	
5/5/22	Matching the found German dataset with Indian dataset And learning about different models of Raspberry Pi	101904104	Akanksha Singh	5	5	Basics of Raspberry Pi should be cleared
		101904017	Vinay Kumar	5	5	
		101904095	Yashvardhan Pachauri	5	4.5	
		101904020	Ayush Vashisth	5	4.5	
		101904114	Gursimar Singh	5	4.5	
10/5/22	Reading, searching and building CNN model. Studying about working of NodeMCU	101904104	Akanksha Singh	5	5	Read more about CNN
		101904017	Vinay Kumar	5	5	
		101904095	Yashvardhan Pachauri	5	5	
		101904020	Ayush Vashisth	5	5	
		101904114	Gursimar Singh	5	5	
14/5/22	Training and testing to increase the efficiency of the model	101904104	Akanksha Singh	5	5	Optimize model
		101904017	Vinay Kumar	5	5	
		101904095	Yashvardhan Pachauri	5	5	

		101904020	Ayush Vashisth	5	5	
		101904114	Gursimar Singh	5	5	

Signature of the evaluation committee member(s): – Dr Santosh Kumar

	Name of the Evaluator	Signature
Evaluator-1:	Manoj Baduni	
Evaluator-2:		
Evaluator-3:	Santanu Paul	

Average Marks :-

Akanksha – 10

Vinay – 9.9

Yash – 9.7

Ayush – 9.7

Gursimar – 9.7

Thapar Institute of Engineering and Technology, Patiala
Electrical and Instrumentation Engineering Department
Capstone Project Evaluation

Group No: 1
Branch: Electrical Engineering
Supervisor: S.K. Jain & Vishal Srivastava
Date of Evaluation: 17/05/2022

Component and Cost Analysis Form-1

S. No.	Name of the component(s)	Specifications	Cost (In INR)	Status Purchased/Ordered
1	RASPBERRY PIE	Raspberry pi 4	8000	-
2	PI CAMERA	8 MP Module V2	2700	-
3	BLUETOOTH MODULE	HC -05	700	-
4	NodeMCU	ESP32-32	250	-
5	DUAL H BRIDGE / DC motor	IC L293D	2000	-
6	HARDWARE EQUIPMENT	Tyre, connecting wires, wooden plank, etc.	1500	-
7	ULTRASONIC SENSOR	HC-SR04	2000	-
8	BLUETOOTH SPEAKER	Mini boost 3W	350	-
		Grand Total (in INR)	18000	

Project Title: Autonomous Vehicle Control through Traffic Signs Detection

Name of the Group Member(s)	Roll No.	Signature with date
1.Yashvardhan Pachauri	101904095	17/05/2022
2.Vinay Kumar	101904017	
3.Ayush Vashisth	101904020	17/5/22.
4.Akanksha Singh	101904104	17/5/22
5.Gursimar Singh	101904114	

REFERENCE

[1] Luo, Hengliang, Yi Yang, Bei Tong, Fuchao Wu, and Bin Fan. "Traffic sign recognition using a multi-task convolutional neural network." *IEEE Transactions on Intelligent Transportation Systems* 19, no. 4 (2017): 1100-1111.
DOI: 10.1109/TITS.2017.2714691

[2] Ju, Moran, Jiangning Luo, Panpan Zhang, Miao He, and Haibo Luo. "A simple and efficient network for small target detection." *IEEE Access* 7 (2019): 85771-85781.
DOI: 10.1109/ACCESS.2019.2924960

[3] Qin, Zhongbing, and Wei Qi Yan. "Traffic-sign recognition using deep learning." In *International Symposium on Geometry and Vision*, pp. 13-25. Springer, Cham, 2021.
DOI : 10.1007/978-3-030-72073-5_2

[4] Said, Riadh Ayachi1 Yahia ElFahem, and Mohamed Atri. "To Perform Road Signs Recognition for Autonomous Vehicles Using Cascaded Deep Learning Pipeline." *Artificial Intelligence Advances*.
DOI: 10.30564/aia.v1i1.569

[5] Zhang, Shiwen. *Traffic sign detection for vision-based driver's assistance in land-based vehicles*. Technical report, School of Aeronautics and Astronautics-Stanford University, 2016.

[6] Wan, Haifeng, Lei Gao, Manman Su, Qinglong You, Hui Qu, and Qirun Sun. "A novel neural network model for traffic sign detection and recognition under extreme conditions." *Journal of Sensors* 2021 (2021).
DOI: 10.1155/2021/9984787

[7] Nikhitha, M., K. P. Swetha, L. Nagamani Swathi, A. Sushma, and E. Sai Santhoshini. "TRAFFIC SIGN BOARD IDENTIFICATION AND VOICE VIGILANCE SYSTEM." *Journal homepage: www. ijrpr. com ISSN* 2582: 7421

[8] Bailke, Preeti, and Kunjal Agrawal. "Traffic Sign Classification Using CNN."
DOI: 10.22214/ijraset.2022.40224

[9] Alghmgham, Danyah A., Ghazanfar Latif, Jaafar Alghazo, and Loay Alzubaidi. "Autonomous traffic sign (ATSR) detection and recognition using deep

CNN." *Procedia Computer Science* 163 (2019): 266-274.
DOI: 10.1016/j.procs.2019.12.108

[10] Alam, Altaf, and Zainul Abdin Jaffery. "Indian traffic sign detection and recognition." *International Journal of Intelligent Transportation Systems Research* 18, no. 1 (2020): 98-112.
DOI: 10.1007/s13177-019-00178-1

[11] Sudha, S. K., and L. Mahathi. "A Real-Time System for Detection and Recognition of Traffic Signs."

[12] Shafiei, Sepehr Foroughi, Hafiza Rimsha Rauf, and Yashdeep Singh. "TRAFFIC SIGN DETECTION USING DEEP LEARNING BY GROUP 36."

[13] De la Escalera, Arturo, J. Ma Armingol, and Mario Mata. "Traffic sign recognition and analysis for intelligent vehicles." *Image and vision computing* 21, no. 3 (2003): 247-258.
DOI: 10.1016/S0262-8856(02)00156-7

[14] Lorsakul, Auranuch, and Jackrit Suthakorn. "Traffic sign recognition for intelligent vehicle/driver assistance system using neural network on opencv." In *The 4th International Conference on Ubiquitous Robots and Ambient Intelligence*. 2007.

[15] Tabernik, Domen, and Danijel Skočaj. "Deep learning for large-scale traffic-sign detection and recognition." *IEEE transactions on intelligent transportation systems* 21, no. 4 (2019): 1427-1440.
DOI : 10.1109/TITS.2019.2913588

[16] Garcia-Garrido, Miguel Angel, Miguel Angel Sotelo, and Ernesto Martin-Gorostiza. "Fast traffic sign detection and recognition under changing lighting conditions." In *2006 IEEE Intelligent Transportation Systems Conference*, pp. 811-816. IEEE, 2006. DOI:10.1109/ITSC.2006.1706843

[17] Huang, Shu-Chun, Huei-Yung Lin, and Chin-Chen Chang. "An in-car camera system for traffic sign detection and recognition." In *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, pp. 1-6. IEEE, 2017.
DOI: 10.1109/IFSA-SCIS.2017.8023239

[18] Wali, Safat B., Mahammad A. Hannan, Aini Hussain, and Salina A. Samad. "An automatic traffic sign detection and recognition system based on colour segmentation, shape matching, and SVM." *Mathematical Problems in Engineering* 2015 (2015). DOI: 10.1155/2015/250461

[19] Damavandi, Y. Baleghi, and Karim Mohammadi. "Speed limit traffic sign detection and recognition." In *IEEE Conference on Cybernetics and Intelligent Systems, 2004.*, vol. 2, pp. 797-802. IEEE, 2004.
DOI: 0.1109/ICCIS.2004.1460690

ANNEXURE

FULL SOURCE CODE

https://colab.research.google.com/drive/1GYb3clx-7EfV-1QuSXA_w_ABUgPVlYn7?usp=sharing

CODE

```
import os
os.getcwd()
from google.colab import drive
drive.mount('/content/gdrive', force_remount=True)
os.getcwd()
#install Kaggle
!pip install -q kaggle
from google.colab import files
files.upload()
#create a kaggle folder
! mkdir ~/.kaggle
#copy the kaggle.json to folder created
! cp kaggle.json  ~/.kaggle
#Permission for the json to act
! chmod 600  ~/.kaggle
#Permission for the json to act
! chmod 600 '/content/gdrive/MyDrive/traffic sign content/data split/kaggle.json'
# Importing the dataset from the kaggle
# 1. GTSRB - German Traffic Sign Recognition Benchmark
!kaggle datasets download -d meowmeowmeowmeowmeow/gtsrb-german-traffic-sign
!unzip gtssrb-german-traffic-sign.zip
#copy the kaggle.json to folder created
! cp -r '/content/meta' '/content/gdrive/MyDrive/traffic sign content'
!unzip traffic-sign-dataset-classification.zip
# Studying the dataset1
dir = "/content/meta"
```

```
os.listdir(dir) #this function will list all the files in the directory #mentioned
path = '38.png'
os.path.join(dir, path) #this function joins paths together
# REFERENCE: https://www.techiedelight.com/list-all-subdirectories-in-directory-python/
initial_count_file = 0
initial_count_folder = 0
dir = "/content/meta"
for path in os.listdir(dir):
    if os.path.isfile(os.path.join(dir, path)):
        initial_count_file += 1
    if os.path.isdir(os.path.join(dir, path)):
        initial_count_folder += 1
print("Number of files in Meta Folder of GTSRB ",initial_count_file)
print("Number of sub-folders in  Meta Folder of GTSRB ",initial_count_folder)
initial_count_file = 0
initial_count_folder = 0
dir = "/content/test"
for path in os.listdir(dir):
    if os.path.isfile(os.path.join(dir, path)):
        initial_count_file += 1
    if os.path.isdir(os.path.join(dir, path)):
        initial_count_folder += 1
print("Number of files in Test Folder of GTSRB ",initial_count_file)
print("Number of sub-folders in Test Folder of GTSRB ",initial_count_folder)
initial_count_file = 0
initial_count_folder = 0
dir = "/content/train"
for path in os.listdir(dir):
    if os.path.isfile(os.path.join(dir, path)):
        initial_count_file += 1
    if os.path.isdir(os.path.join(dir, path)):
        initial_count_folder += 1
print("Number of images in Train Folder of GTSRB ",initial_count_file)
```

```
print("Number of classes in Train Folder of GTSRB ",initial_count_folder)
class_count={}
dir = "/content/train"
for path in os.listdir(dir):
    initial_count_file = 0
    sub_dir = os.path.join(dir, path)
    for file in os.listdir(sub_dir):
      if os.path.isfile(os.path.join(sub_dir, file)):
        initial_count_file += 1
    class_count[int(path)]=initial_count_file
    print("Number of files in ",path," class of Train Folder of GTSRB ",initial_count_file)
#Sorting the classes dictionary with respect to keys
sorted(class_count.keys())
class_count
# Creates a sorted dictionary (sorted by key)
from collections import OrderedDict
print(class_count)
class_count = OrderedDict(sorted(class_count.items()))
print(class_count)
class_count.values()
print("No. of classes : ",len(os.listdir('/content/train')))
count=0
for path in os.listdir('/content/train'):
  for file in os.listdir(os.path.join('/content/train',path)):
    count+=1

print("total no. of images",count)
# Reference : https://www.adamsmith.haus/python/answers/how-to-display-the-value-of-each-bar-in-a-bar-chart-using-matplotlib-in-python
#Reference : https://www.geeksforgeeks.org/bar-plot-in-matplotlib/
#Figure Size
plt.figure(figsize=(16,16))
#Horizontal Bar Plot
```

```
plt.barh(list(classes.values()), list(class_count.values()))
#Add Plot Title
plt.title('Number of Training Images in Each class ', loc ='left',fontsize= 22 )
plt.xlabel('Number of image file')
plt.ylabel('Classes')
# Add annotation to bars
for index,value in enumerate(list(class_count.values())):
    plt.text(value, index, str(value))
m=list(enumerate(list(class_count.values())))
#reference : https://www.delftstack.com/howto/matplotlib/how-to-add-title-to-subplots-in-matplotlib/
#If you use Matlab-like style in the interactive plotting, then you could use plt.gca()
#to get the reference of the current axes of the subplot and combine set_title() or title.set_text() method
#to set title to the subplots in Matplotlib.
import matplotlib.pyplot as plt
from keras.preprocessing.image import load_img, img_to_array
picture_size = 48
folder_path = '/content/train/'
plt.figure(figsize = (16,18))
i=1
for key in classes:
  plt.subplot(9,5,i)
  img = load_img(folder_path+str(key)+"/"+os.listdir(folder_path+ str(key))[3], target_size=(picture_size, picture_size))
  plt.imshow(img)
  plt.gca().set_title(classes[key])
  i = i+1
plt.tight_layout()
plt.show()
folder_path = "/content/train"
plt.figure(figsize = (12,12))
for path in os.listdir(folder_path):
```

```
  sub_class = os.path.join(folder_path,path)
for i in range(2, 27, 1):
  plt.subplot(5,9,i-1)
  img = load_img(folder_path+"meta/"+os.listdir(folder_path+'meta')[i], target_size=(picture_
size, picture_size))
  print(img)
  plt.imshow(img)
plt.show()
# Studying dataset2
source_folder= '/content/traffic_Data/DATA/36'
destination_folder ='/content/traffic_Data/DATA/30'
for path in os.listdir(source_folder):
  src_path = os.path.join(source_folder,path)
  dst_path = os.path.join(destination_folder,path)
  shutil.move(src_path,dst_path)
shutil.move('/content/traffic_Data/DATA/036_0018.png', '/content/traffic_Data/DATA/30')
shutil.move(src_path,dst_path)
classes1 = {0:'Speed limit (5km/h)',
            1:'Speed limit (15km/h)',
            2:'Speed limit (30km/h)',
            3:'Speed limit (40km/h)',
            4:'Speed limit (50km/h)',
            5:'Speed limit (60km/h)',
            6:'Speed limit (70km/h)',
            8:'Dont Go straight or left',
            7:'speed limit (80km/h)',
            9:'Dont Go straight or Right',
            10:'Dont Go straight',
            11:'Dont Go Left',
            12:'Dont Go Left or Right',
            13:'Dont Go Right',
            14:'Dont overtake from Left',
            15:'No Uturn',
```

16:'No Car',

17:'No horn',

18:'End of speed limit (40km/h)',

19:'End of speed limit (50km/h)',

20:'Go straight or right',

21:'Go straight',

22:'Go Left',

23:'Go Left or right',

24:'Go Right',

25:'keep Left',

26:'keep Right',

27:'Roundabout mandatory',

28:'watch out for cars',

29:'Horn',

30:'Bicycles crossing',

31:'Uturn',

32:'Road Divider',

33:'Traffic signals',

34:'Danger Ahead',

35:'Zebra Crossing',

36:'NULL',

37:'Children crossing',

38:'Dangerous curve to the left',

39:'Dangerous curve to the right',

40:'Unknown1',

41:'Unknown2',

42:'Unknown3',

43:'Go right or straight',

44:'Go left or straight',

45:'Unknown4',

46:'ZigZag Curve',

47:'Train Crossing',

48:'Under Construction',

```
            49:'Unknown5',
            50:'Fences',
            51:'Heavy Vehicle Accidents',
            52:'Unknown6',
            53:'Give Way',
            54:'No stopping',
            55:'No entry',
            56:'Unknown7',
            57:'Unknown8'}
initial_count_file = 0
initial_count_folder = 0
dir = "/content/traffic_Data/TEST"
for path in os.listdir(dir):
    if os.path.isfile(os.path.join(dir, path)):
        initial_count_file += 1
    if os.path.isdir(os.path.join(dir, path)):
        initial_count_folder += 1
print("Number of files in Test Folder of traffic_Data ",initial_count_file)
print("Number of sub-folders in Test Folder of traffic_Data ",initial_count_folder)
class_count1={}
dir = "/content/traffic_Data/DATA"
for path in os.listdir(dir):
    initial_count_file = 0
    sub_dir = os.path.join(dir, path)
    for file in os.listdir(sub_dir):
      if os.path.isfile(os.path.join(sub_dir, file)):
        initial_count_file += 1
    class_count1[int(path)]=initial_count_file
    print("Number of files in ",path," class of Train Folder of traffic_Data ",initial_count_file)
print("No. of classes : ",len(os.listdir('/content/traffic_Data/DATA')))
count=0
for path in os.listdir('/content/traffic_Data/DATA'):
  for file in os.listdir(os.path.join('/content/traffic_Data/DATA',path)):
```

```
    count+=1
print("total no. of images",count)
# Creates a sorted dictionary (sorted by key)
from collections import OrderedDict
print(class_count1)
class_count1 = OrderedDict(sorted(class_count1.items()))
print(class_count1)
# Reference : https://www.adamsmith.haus/python/answers/how-to-display-the-value-of-each-bar-in-a-bar-chart-using-matplotlib-in-python
#Reference : https://www.geeksforgeeks.org/bar-plot-in-matplotlib/
#Figure Size
plt.figure(figsize=(16,18))
#Horizontal Bar Plot
plt.barh(list(classes1.values()), list(class_count1.values()), height=0.8)
#Add Plot Title
plt.title('Number of Training Images in Each class ', loc ='left',fontsize= 22 )
plt.xlabel('Number of image file')
plt.ylabel('Classes')
# Add annotation to bars
for index,value in enumerate(list(class_count1.values())):
    plt.text(value, index, str(value))
#reference : https://www.delftstack.com/howto/matplotlib/how-to-add-title-to-subplots-in-matplotlib/
#If you use Matlab-like style in the interactive plotting, then you could use plt.gca()
#to get the reference of the current axes of the subplot and combine set_title() or title.set_text() method
#to set title to the subplots in Matplotlib.
import matplotlib.pyplot as plt
from keras.preprocessing.image import load_img, img_to_array
picture_size = 48
folder_path = '/content/traffic_Data/DATA/'
plt.figure(figsize = (16,18))
i=1
```

```
for key in classes1:
  plt.subplot(12,5,i)
  if(key==36):
    continue
  img = load_img(folder_path+str(key)+"/"+os.listdir(folder_path+ str(key))[0], target_size=(
picture_size, picture_size))
  plt.imshow(img)
  plt.gca().set_title(classes1[key])
  i = i+1
plt.tight_layout()
plt.show()
m = list(classes1.values())
n =list(class_count1.values())
for i in range(len(m)):
  print(i, " : ",m[i]," : ",n[i])
list(enumerate(list(class_count1.values())))
# **COMBINING THE TWO DATASET**
import shutil
#No entry
source_folder= '/content/traffic_Data/DATA/55'
destination_folder ='/content/Train/17'
for path in os.listdir(source_folder):
  src_path = os.path.join(source_folder,path)
  shutil.copy(src_path,destination_folder)
#give way
import shutil
source_folder= '/content/traffic_Data/DATA/53'
destination_folder ='/content/traffic_Data/DATA/13'
for path in os.listdir(source_folder):
  src_path = os.path.join(source_folder,path)
  dst_path = os.path.join(destination_folder,path)
  shutil.move(src_path,dst_path)
picture_size = 48
```

```
folder_path_ = "/content/"
#/content/sample_data/Traffic Sign dataset/Meta/0.png
plt.figure(figsize = (12,12))
for i in range(1, 26, 1):
  plt.subplot(5,5,i)
  img = load_img(folder_path_+"batch/"+os.listdir(folder_path_+'batch')[i+25], target_size=(
picture_size, picture_size))
  plt.imshow(img)
plt.show()
#-------------------------
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
img = mpimg.imread('/content/traffic_Data/DATA/36/036_0018.png')
plt.imshow(img)
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os
import tensorflow as tf
from PIL import Image
#Importing Deep Learining Libraries
from keras.preprocessing.image import load_img, img_to_array
from keras.preprocessing.image import ImageDataGenerator
# DATA EXTRACTION
categ = {'1': '30 speed limit', '4':'70 speed limit', '15':'no entry', '18':'go slow', '33':'right hand c
urve', '34':'left hand curve' }
import pandas as pd
import numpy as np
df = pd.read_csv('/content/gdrive/MyDrive/traffic sign content/Meta.csv
# Making Training and Validation Data
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import keras
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPool2D, Flatten, Dropout
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import adam_experimental
from sklearn.metrics import classification_report,confusion_matrix
import tensorflow as tf

import cv2
import os
import numpy as np
pip install split-folders
from google.colab.patches import cv2_imshow
i = cv2.imread('/content/drive/MyDrive/traffic sign content/data split/train/15/00015_00000_
00000.png')
plt.imshow(i)
#https://www.kaggle.com/questions-and-answers/102677
categ = {'1': '30 speed limit', '4':'70 speed limit', '15':'no entry', '18':'go slow', '33':'right hand c
urve', '34':'left hand curve' }
import splitfolders
splitfolders.ratio('/content/sample_data/Traffic Sign dataset/train_sign',output="/content/drive
/MyDrive/traffic sign content/data split", seed=1337,ratio=(0.8,0.2))
test_file = pd.read_csv('/content/Test.csv')
# Putting input images into numpy array
import cv2
data = []
labels = []
folder_path = '/content/train'
height = 25
width = 25
channels = 3
classes = 43
n_inputs = height * width * channels
```

```
for i in range(classes) :
  path = os.path.join(folder_path,str(i))
  print(path)
  class_path = os.listdir(path)
  for add in class_path:
    try:
      address = os.path.join(path,add)
      img = cv2.imread(address)
      image_from_array = Image.fromarray(img,'RGB')
      size_image = image_from_array.resize((height,width))
      data.append(np.array(size_image))
      labels.append(i)
    except AttributeError: print(" ")
Cells = np.array(data)
labels = np.array(labels)
#Randomize the order of the input images
s = np.arange(Cells.shape[0])
np.random.shuffle(s)
Cells=Cells[s]
labels=labels[s]
#Saving the Cells and labels file
np.save("/content/gdrive/MyDrive/traffic sign content/Cells.npy",Cells)
np.save("/content/gdrive/MyDrive/traffic sign content/labels.npy",labels)
Cell1 = np.load("/content/gdrive/MyDrive/traffic sign content/Cells.npy")
labels1 = np.load("/content/gdrive/MyDrive/traffic sign content/labels.npy")
#Checking if 2 arrays are equal or not
if (Cell1 == Cells).all():
  print("t")
if (labels1 == labels).all():
  print("m")
print('T')
#Function to compare 2 files are equal or not
import filecmp.
```

```
filecmp. cmp('/content/gdrive/MyDrive/traffic sign content/Cells.npy', Cells)
type(labels), type(Cells), labels.shape, Cells.shape
#Spliting the images into train and validation sets
(X_train,X_val)=Cells[(int)(0.2*len(labels)):],Cells[:(int)(0.2*len(labels))]
X_train = X_train.astype('float32')/255
X_val = X_val.astype('float32')/255
(y_train,y_val)=labels[(int)(0.2*len(labels)):],labels[:(int)(0.2*len(labels))]
#Using one hote encoding for the train and validation labels
from tensorflow.keras.utils import to_categorical
y_train = to_categorical(y_train, 43)
y_val = to_categorical(y_val, 43)
#Definition of the CNN model
from keras.models import Sequential
from keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Dropout
model = Sequential()
#1st CNN layer
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu', input_shape=X_train.sha
pe[1:]))
#2nd CNN layer
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
#3rd CNN layer
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Flatten())
#Fully Connected layer 1st layer
model.add(Dense(256, activation='relu'))
model.add(Dropout(rate=0.5))
#Fully Connected layer 2nd layer
model.add(Dense(43, activation='softmax'))
#Compilation of the model
```

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
epochs = 20
history = model.fit(X_train, y_train, batch_size=32, epochs=epochs,validation_data=(X_val,
y_val))
score = model.evaluate(X_train, y_train, verbose=0)
score
print("%s: %.2f%%" % (model.metrics_names[1], score[1]*100))
# serialize model to JSON
model_json = model.to_json()
with open("/content/gdrive/MyDrive/traffic sign content/model.json", "w") as json_file:
  json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("/content/gdrive/MyDrive/traffic sign content/model.h5")
print("Saved model to disk")
from keras.models import model_from_json
# load json and create model
json_file = open('/content/gdrive/MyDrive/traffic sign content/model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
# load weights into new model
loaded_model.load_weights("/content/gdrive/MyDrive/traffic sign content/model.h5")
print("Loaded model from disk")
loaded_model.compile( loss='categorical_crossentropy', optimizer='adam',metrics=['accuracy
'])
score_save = loaded_model.evaluate(X_train, y_train,verbose=0)
print("%s: %.2f%%" % (loaded_model.metrics_names[1], score_save[1]*100))
import pickle
# save the model to disk
filename = '/content/gdrive/MyDrive/traffic sign content/model.sav'
pickle.dump(model, open(filename, 'wb'))
import pickle
# convert the history.history dict to a pandas DataFrame:
```

```
hist_df = pd.DataFrame(history.history)

# save to json:
hist_json_file = '/content/gdrive/MyDrive/traffic sign content/history.json'
with open(hist_json_file, mode='w') as f:
    hist_df.to_json(f)
# or save to csv:
hist_csv_file = '/content/gdrive/MyDrive/traffic sign content/history.csv'
with open(hist_csv_file, mode='w') as f:
    hist_df.to_csv(f)
# load the model from disk
loaded_model_history = pickle.load(open(filename, 'rb'))
 with open('/trainHistoryDict', 'wb') as file_pi:
        pickle.dump(history.history, file_pi)
#Display of the accuracy and the loss values
import matplotlib.pyplot as plt
plt.figure(0)
plt.plot(loaded_model.loaded_model['accuracy'], label='training accuracy')
plt.plot(loaded_model.loaded_model['val_accuracy'], label='val accuracy')
plt.title('Accuracy')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.legend()
plt.figure(1)
plt.plot(loaded_model.loaded_model['loss'], label='training loss')
plt.plot(loaded_model.loaded_model['val_loss'], label='val loss')
plt.title('Loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.legend()
#Display of the accuracy and the loss values
import matplotlib.pyplot as plt
```

```
plt.figure(0)
plt.plot(history.history['accuracy'], label='training accuracy')
plt.plot(history.history['val_accuracy'], label='val accuracy')
plt.title('Accuracy')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.legend()
plt.figure(1)
plt.plot(history.history['loss'], label='training loss')
plt.plot(history.history['val_loss'], label='val loss')
plt.title('Loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.legend()
# **Plotting Accuracy & Loss**
plt.style.use('dark_background')
plt.figure(figsize=(20,10))
plt.subplot(1,2,1)
plt.suptitle('Optimizer : Adam', fontsize=10)
plt.ylabel('Loss', fontsize=16)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label = 'Validation Loss')
plt.legend(loc = 'upper right')
plt.subplot(1, 2, 2)
plt.ylabel('Accuracy', fontsize=16)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label = 'Validation Accuracy')
plt.legend(loc = 'lower right')
plt.show()
plt.style.use('dark_background')
plt.figure(figsize=(20,10))
plt.subplot(1,2,1)
plt.suptitle('Optimizer : Adam', fontsize=10)
```

```
plt.ylabel('Loss', fontsize=16)
plt.plot(save_model.save_model['loss'], label='Training Loss')
plt.plot(save_model.save_model['val_loss'], label = 'Validation Loss')
plt.legend(loc = 'upper right')

plt.subplot(1, 2, 2)
plt.ylabel('Accuracy', fontsize=16)
plt.plot(save_model.save_model['accuracy'], label='Training Accuracy')
plt.plot(save_model.save_model['val_accuracy'], label = 'Validation Accuracy')
plt.legend(loc = 'lower right')
plt.show()
# Testing our model with test dataset
from keras.models import model_from_json
# load json and create model
json_file = open('/content/gdrive/MyDrive/traffic sign content/model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
# load weights into new model
loaded_model.load_weights("/content/gdrive/MyDrive/traffic sign content/model.h5")
print("Loaded model from disk")
from PIL import Image
from sklearn.metrics import accuracy_score
import pandas as pd
y_test = pd.read_csv('/content/Test.csv')

labels = y_test['ClassId'].values
imgs = y_test['Path'].values
data1=[]
for img in imgs:
  image = Image.open(img)
  image = image.resize((25,25))
  data1.append(np.array(image))
```

```
X_test = np.array(data1)
predict_x=loaded_model.predict(X_test)
pred=np.argmax(predict_x,axis=1)
#pred = loaded_model.predict_classes(X_test)
#Accuracy with the test data
from sklearn.metrics import accuracy_score
accuracy_score(labels,pred)
```

ch1_ch6

ORIGINALITY REPORT

12%	4%	4%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Thapar University, Patiala Student Paper	5%
2	Qinghai Lin, Jinyong Chen, Guilong Li, Zhaocheng He. "Signal timing parameters inference method at intersections using license plate recognition data", IET Intelligent Transport Systems, 2022 Publication	1%
3	Md. Sayem Miah, Robin Acharjee, Mrinal Kanti Dhar, Ifthekhar Ahammad, Md. Thasfiquzzaman, Md. Niaz Morshedul Haque. "Traffic signal interactive autonomous vehicle: An approach for intelligent path planning and steering control", 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017 Publication	1%
4	Baratsanjeevi T, Deepakkrishna S, Harrine M P, Sharan S, Prabhu E. "IoT based Traffic Sign Detection and Violation Control", 2020 Second International Conference on Inventive	1%

Thapar Institute of Engineering and Technology, Patiala
Electrical and Instrumentation Engineering Department
Capstone Project Evaluation

Group No: 1
Branch: Electrical Engineering
Supervisor: S.K. Jain & Vishal Srivastava
Date of Evaluation: 17/05/2022

Component and Cost Analysis Form-1

S. No.	Name of the component(s)	Specifications	Cost (In INR)	Status Purchased/Ordered
1	RASPBERRY PIE	Raspberry pi 4	8000	-
2	PI CAMERA	8 MP Module V2	2700	-
3	BLUETOOTH MODULE	HC -05	700	-
4	NodeMCU	ESP32-32	250	-
5	DUAL H BRIDGE / DC motor	IC L293D	2000	-
6	HARDWARE EQUIPMENT	Tyre, connecting wires, wooden plank, etc.	1500	-
7	ULTRASONIC SENSOR	HC-SR04	2000	-
8	BLUETOOTH SPEAKER	Mini boost 3W	350	-
		Grand Total (in INR)	18000	

Project Title: Autonomous Vehicle Control through Traffic Signs Detection

Name of the Group Member(s)	Roll No.	Signature with date
1.Yashvardhan Pachauri	101904095	17/05/2022
2.Vinay Kumar	101904017	
3.Ayush Vashisth	101904020	17/5/22.
4.Akanksha Singh	101904104	17/5/22
5.Gursimar Singh	101904114	

Thapar Institute of Engineering and Technology, Patiala
Electrical and Instrumentation Engineering Department
Progress Report Form
Capstone Project (Electrical Engineering)

Group No: 1
Name of the Supervisor: S.K.Jain & Vishal Srivastava
Project Title: Autonomous Vehicle Control Through Traffic Signs

First/Second/Final Evaluation (Dated: 17/5/2022) **Code: UEE795**

Date	Work done/discussed	Roll No.	Student name	Self-Evaluation	Supervisor Evaluation	Supervisor's Sign with remarks
24/4/22	Everyone read different Research papers on traffic sign detection In both software and hardware part and contributed their ideas	101904104	Akanksha Singh	5	5	Work hard
		101904017	Vinay Kumar	5	5	
		101904095	Yashvardhan Pachauri	5	4.5	
		101904020	Ayush Vashisth	5	4.5	
		101904114	Gursimar Singh	5	4.5	
30/4/22	Searched for the dataset and found the dataset on Kaggle and Indepth research of raspberry pi	101904104	Akanksha Singh	5	5	Load data & understand its limitation
		101904017	Vinay Kumar	5	4.5	
		101904095	Yashvardhan Pachauri	5	4.5	
		101904020	Ayush Vashisth	5	4.5	
		101904114	Gursimar Singh	5	4.5	
5/5/22	Matching the found German dataset with Indian dataset And learning about different models of Raspberry Pi	101904104	Akanksha Singh	5	5	Basics of Raspberry Pi should be cleared
		101904017	Vinay Kumar	5	5	
		101904095	Yashvardhan Pachauri	5	4.5	
		101904020	Ayush Vashisth	5	4.5	
		101904114	Gursimar Singh	5	4.5	
10/5/22	Reading, searching and building CNN model. Studying about working of NodeMCU	101904104	Akanksha Singh	5	5	Read more about CNN
		101904017	Vinay Kumar	5	5	
		101904095	Yashvardhan Pachauri	5	5	
		101904020	Ayush Vashisth	5	5	
		101904114	Gursimar Singh	5	5	
14/5/22	Training and testing to increase the efficiency of the model	101904104	Akanksha Singh	5	5	Optimize model
		101904017	Vinay Kumar	5	5	
		101904095	Yashvardhan Pachauri	5	5	

		101904020	Ayush Vashisth	5	5	
		101904114	Gursimar Singh	5	5	

Signature of the evaluation committee member(s): – Dr Santosh Sarar

	Name of the Evaluator	Signature
Evaluator-1:	Manoj Bardwaj	
Evaluator-2:		
Evaluator-3:	Santanu Paul	

Average Marks :-

Akansha – 10

Vinay – 9.9

Yash – 9.7

Ayush – 9.7

Gursimar – 9.7

Thapar Institute of Engineering and Technology, Patiala
Electrical and Instrumentation Engineering Department
Progress Report Form
Capstone Project (Electrical Engineering)

Group No: 1 **Name of the Supervisor:** S.K. Jain & Vishal Srivastava
Project Title: Autonomous Vehicle Control through Traffic Sign

First/Second/Final Evaluation (Dated: 01/12/2022**)** **Code: UEE795**

Date	Work done/discussed	Roll No.	Student name	Self-Evaluation	Supervisor Evaluation	Supervisor's Sign with remarks
08/08/22	Everyone read research Papers on traffic signal hardware and software.	101904017	Vinay	5	5	Shrotare Keep work hard Sanjay
		101904029	Ayush	5	4	
		101904114	Gursimar	5	4	
		101904104	Akanksha	5	5	
		101904095	Yashvardhan	5	5	
22/08/22	search for dataset and found the data on Kaggle.	101904017	Vinay	5	5	Shrotare meet me on regular basis Sanjay
		101904029	Ayush	5	3	
		101904114	Gursimar	5	3	
		101904104	Akanksha	5	5	
		101904095	Yashvardhan	5	5	
30/09/22	matching the found German dataset with Indian dataset	101904017	Vinay jain	5	4	Shrotare try to simulate the result Sanjay
		101904029	Ayush	5	4	
		101904114	Gursimar	5	4	
		101904104	Akanksha	5	4	
		101904095	Yashvardhan	5	4	
20/09/22	building CNN model. Study about working of Node CU	101904017	Vinay	5	5	Shrotare Keep work hard Sanjay
		101904029	Ayush	5	5	
		101904114	Gursimar	5	5	
		101904104	Akanksha	5	5	
		101904095	Yashvardhan	5	5	
21/10/22	Train and Testing To increase the efficiency	101904017	Vinay	5	5	Shrotare Keep work hard Sanjay
		101904029	Ayush	5	5	
		101904114	Gursimar	5	5	
		101904104	Akanksha	5	5	
		101904095	Yashvardhan	5	5	

Signature of the evaluation committee member(s):

	Name of the Evaluator	Signature
Evaluator-1:	Dr, Souvik Ganguli	Ganguli
Evaluator-2:	Dr. Amit Kumar	Amit Kumar
Evaluator-3:		

Thapar Institute of Engineering and technology, Patiala

Electrical and Instrumentation Engineering Department

Capstone Project: Daily Diary-2022

NAME OF THE STUDENT: Akanksha Singh, Yashvardhan Pachauri, Ayush Vashisth, Vinay Kumar Jain, Gursimar Singh Nagpal

Group and Roll No.: (Group - 01) | 101904104, 101904095, 101904020, 101904017, 101904114

March		April	
Date	**Description**	**Date**	**Description**
01-03-2022	Study on both topics	01-04-2022	MST
02-03-2022	Study on both topics	02-04-2022	MST
03-03-2022	Study on both topics	03-04-2022	MST
04-03-2022	Meeting with Faculty Mentor	04-04-2022	MST
05-03-2022	Documentation for evaluation 1	05-04-2022	MST
06-03-2022	Documentation for evaluation 1	06-04-2022	MST
07-03-2022	Documentation for evaluation 1	07-04-2022	MST
08-03-2022	Documentation for evaluation 1	08-04-2022	MST
09-03-2022	Documentation for evaluation 1	09-04-2022	MST
10-03-2022	Evaluation 1 day, but it got postponed	10-04-2022	MST
11-03-2022	Waited for evaluation 1	11-04-2022	MST
12-03-2022	Waited for evaluation 1	12-04-2022	MST
13-03-2022	Waited for evaluation 1	13-04-2022	MST
14-03-2022	Evaluation 1 day	14-04-2022	MST
15-03-2022	Meeting with Faculty Mentor	15-04-2022	MST
16-03-2022	Worked on feedback of evaluation 1	16-04-2022	MST
17-03-2022	Worked on feedback of evaluation 1	17-04-2022	MST
18-03-2022	Worked on feedback of evaluation 1	18-04-2022	Studied various research papers and topologies
19-03-2022	Worked on feedback of evaluation 1	19-04-2022	Studied various research papers and topologies
20-03-2022	Worked on feedback of evaluation 1	20-04-2022	Studied various research papers and topologies
21-03-2022	Worked on feedback of evaluation 1	21-04-2022	Studied various research papers and topologies
22-03-2022	Worked on feedback of evaluation 1	22-04-2022	Studied various research papers and topologies
23-03-2022	Search of new project ideas	23-04-2022	Studied various research papers and topologies
24-03-2022	Search of new project ideas	24-04-2022	Studied various research papers and topologies
25-03-2022	Search of new project ideas	25-04-2022	Studied various research papers and topologies
26-03-2022	Search of new project ideas	26-04-2022	Studied various research papers and topologies
27-03-2022	Search of new project ideas	27-04-2022	Meeting with Faculty Mentor
28-03-2022	Search of new project ideas	28-04-2022	Studied various research papers and topologies
29-03-2022	Meeting with Faculty Mentor regarding new topic	29-04-2022	Studied various research papers and topologies
30-03-2022	MST	30-04-2022	Studied various research papers and topologies
31-03-2022	MST		

Meeting date and time with Supervisor(s)	1. 24/04/2022 at 1730 Hrs.	Meeting date and time with Supervisor(s)	1. 27/04/2022 at 1730 Hrs.
	2. 30/04/2022 at 1730 Hrs.		2
	3		3
	4		4

Signature of Student

Signature of Student

Thapar Institute of Engineering and technology, Patiala

Electrical and Instrumentation Engineering Department

Capstone Project: Daily Diary-2022

NAME OF THE STUDENT: Akanksha Singh, Yashvardhan Pachauri, Ayush Vashisth, Vinay Kumar Jain, Gursimar Singh Nagpal

Group and Roll No.: (Group - 01) | 101904104, 101904095, 101904020, 101904017, 101904114

May		June	
Date	**Description**	**Date**	**Description**
01-05-2022	Studied various research papers and topologies	01-06-2022	Hardware Implementation
02-05-2022	Studied various research papers and topologies	02-06-2022	Hardware Implementation
03-05-2022	Studied various research papers and topologies	03-06-2022	Hardware Implementation
04-05-2022	Studied various research papers and topologies	04-06-2022	Hardware Implementation
05-05-2022	Meeting with Faculty Mentor	05-06-2022	Hardware Implementation
06-05-2022	Started learning MATLAB simulink	06-06-2022	Hardware Implementation
07-05-2022	Started learning MATLAB simulink	07-06-2022	Hardware Implementation
08-05-2022	Started learning MATLAB simulink	08-06-2022	Hardware Implementation
09-05-2022	Started learning MATLAB simulink	09-06-2022	Hardware Implementation
10-05-2022	Started learning MATLAB simulink	10-06-2022	Hardware Implementation
11-05-2022	Started learning MATLAB simulink	11-06-2022	Hardware Implementation
12-05-2022	Started learning MATLAB simulink	12-06-2022	Hardware Implementation
13-05-2022	Meeting with Faculty Mentor	13-06-2022	Hardware Implementation
14-05-2022	Worked on Simulation and Documentation for Eval-2	14-06-2022	Hardware Implementation
15-05-2022	Worked on Simulation and Documentation for Eval-2	15-06-2022	Hardware Implementation
16-05-2022	Worked on Simulation and Documentation for Eval-2	16-06-2022	Hardware Implementation
17-05-2022	Evaluation-2 day	17-06-2022	Hardware Implementation
18-05-2022	Hardware Implementation	18-06-2022	Hardware Implementation
19-05-2022	Hardware Implementation	19-06-2022	Hardware Implementation
20-05-2022	Hardware Implementation	20-06-2022	Hardware Implementation
21-05-2022	Hardware Implementation	21-06-2022	Hardware Implementation
22-05-2022	Hardware Implementation	22-06-2022	Hardware Implementation
23-05-2022	Hardware Implementation	23-06-2022	Hardware Implementation
24-05-2022	Hardware Implementation	24-06-2022	Hardware Implementation
25-05-2022	Hardware Implementation	25-06-2022	Hardware Implementation
26-05-2022	Hardware Implementation	26-06-2022	Hardware Implementation
27-05-2022	Hardware Implementation	27-06-2022	Hardware Implementation
28-05-2022	Hardware Implementation	28-06-2022	Hardware Implementation
29-05-2022	Hardware Implementation	29-06-2022	Hardware Implementation
30-05-2022	Hardware Implementation	30-06-2022	Hardware Implementation
31-05-2022	Hardware Implementation		

Meeting date and time with Supervisor(s)	
	1. 05/05/2022 at 1730 Hrs.
	2. 10/05/2022 at 1730 Hrs.
	3
	4

Signature of Student

Meeting date and time with Supervisor(s)	
	1
	2
	3
	4

Signature of Student

Thapar Institute of Engineering and technology, Patiala

Electrical and Instrumentation Engineering Department

Capstone Project: Daily Diary-2022

NAME OF THE STUDENT: Akanksha Singh, Yashvardhan Pachauri, Ayush Vashisth, Vinay Kumar Jain, Gursimar Singh Nagpal

Group and Roll No.: (Group - 01) | 101904104, 101904095, 101904020, 101904017, 101904114

July		August	
Date	**Description**	**Date**	**Description**
01-07-2022	Hardware Implementation	01-08-2022	Placement Prep and Project Compilation
02-07-2022	Hardware Implementation	02-08-2022	Placement Prep and Project Compilation
03-07-2022	Hardware Implementation	03-08-2022	Placement Prep and Project Compilation
04-07-2022	Hardware Implementation	04-08-2022	Placement Prep and Project Compilation
05-07-2022	Hardware Implementation	05-08-2022	Placement Prep and Project Compilation
06-07-2022	Hardware Implementation	06-08-2022	Placement Prep and Project Compilation
07-07-2022	Hardware Implementation	07-08-2022	Placement Prep and Project Compilation
08-07-2022	Hardware Implementation	08-08-2022	Placement Prep and Project Compilation
09-07-2022	Hardware Implementation	09-08-2022	Placement Prep and Project Compilation
10-07-2022	Hardware Implementation	10-08-2022	Placement Prep and Project Compilation
11-07-2022	Hardware Implementation	11-08-2022	Placement Prep and Project Compilation
12-07-2022	Hardware Implementation	12-08-2022	Placement Prep and Project Compilation
13-07-2022	Hardware Implementation	13-08-2022	Placement Prep and Project Compilation
14-07-2022	Hardware Implementation	14-08-2022	Placement Prep and Project Compilation
15-07-2022	Hardware Implementation	15-08-2022	Placement Prep and Project Compilation
16-07-2022	Hardware Implementation	16-08-2022	Placement Prep and Project Compilation
17-07-2022	Hardware Implementation	17-08-2022	Placement Prep and Project Compilation
18-07-2022	Hardware Implementation	18-08-2022	Placement Prep and Project Compilation
19-07-2022	Hardware Implementation	19-08-2022	Placement Prep and Project Compilation
20-07-2022	Hardware Implementation	20-08-2022	Placement Prep and Project Compilation
21-07-2022	Hardware Implementation	21-08-2022	Placement Prep and Project Compilation
22-07-2022	Hardware Implementation	22-08-2022	Placement Prep and Project Compilation
23-07-2022	Hardware Implementation	23-08-2022	Placement Prep and Project Compilation
24-07-2022	Hardware Implementation	24-08-2022	Placement Prep and Project Compilation
25-07-2022	Hardware Implementation	25-08-2022	Placement Prep and Project Compilation
26-07-2022	Hardware Implementation	26-08-2022	Placement Prep and Project Compilation
27-07-2022	Hardware Implementation	27-08-2022	Placement Prep and Project Compilation
28-07-2022	Hardware Implementation	28-08-2022	Placement Prep and Project Compilation
29-07-2022	Hardware Implementation	29-08-2022	Placement Prep and Project Compilation
30-07-2022	Hardware Implementation	30-08-2022	Placement Prep and Project Compilation
31-07-2022	Hardware Implementation	31-08-2022	Placement Prep and Project Compilation

Meeting date and time with Supervisor(s)	1	Meeting date and time with Supervisor(s)	1. 08/08/2022 at 1730 Hrs
	2		2. 22/08/2022 at 1730 Hrs
	3		3
	4		4

Signature of Student

Signature of Student

Thapar Institute of Engineering and technology, Patiala

Electrical and Instrumentation Engineering Department

NAME OF THE STUDENT: Akanksha Singh, Yashvardhan Pachauri, Ayush Vashisth, Vinay Kumar Jain, Gursimar Singh Nagpal

Group and Roll No.: (Group - 01) | 101904104, 101904095, 101904020, 101904017, 101904114

Group and Roll No.:

September

Date	Description
01-09-2022	3rd Evaluation Preparation
02-09-2022	3rd Evaluation Preparation
03-09-2022	3rd Evaluation Preparation
04-09-2022	3rd Evaluation Preparation
05-09-2022	3rd Evaluation Preparation
06-09-2022	3rd Evaluation Preparation
07-09-2022	3rd Evaluation Preparation
08-09-2022	3rd Evaluation Preparation
09-09-2022	3rd Evaluation Preparation
10-09-2022	3rd Evaluation Preparation
11-09-2022	3rd Evaluation Preparation
12-09-2022	3rd Evaluation Preparation
13-09-2022	3rd Evaluation Preparation
14-09-2022	3rd Evaluation Preparation
15-09-2022	3rd Evaluation Preparation
16-09-2022	3rd Evaluation Preparation
17-09-2022	MST
18-09-2022	MST
19-09-2022	MST
20-09-2022	MST
21-09-2022	MST
22-09-2022	MST
23-09-2022	MST
24-09-2022	MST
25-09-2022	MST
26-09-2022	MST
27-09-2022	MST
28-09-2022	MST
29-09-2022	MST
30-09-2022	MST

Meeting date and time with Supervisor(s)	
	1. 10/09/2022 at 1730 Hrs.
	2. 20/09/2022 at 1730
	3
	4

Signature of Student

Group and Roll No.:

October

Date	Description
01-10-2022	MST
02-10-2022	MST
03-10-2022	Testing and Improvisation
04-10-2022	Testing and Improvisation
05-10-2022	Testing and Improvisation
06-10-2022	Testing and Improvisation
07-10-2022	Testing and Improvisation
08-10-2022	Testing and Improvisation
09-10-2022	Testing and Improvisation
10-10-2022	Report Finalization
11-10-2022	Report Finalization
12-10-2022	Report Finalization
13-10-2022	Report Finalization
14-10-2022	Report Finalization
15-10-2022	Report Finalization
16-10-2022	Report Finalization
17-10-2022	Report Finalization
18-10-2022	Report Finalization
19-10-2022	Report Finalization
20-10-2022	Report Finalization
21-10-2022	Report Finalization
22-10-2022	Report Finalization
23-10-2022	Report Finalization
24-10-2022	Report Finalization
25-10-2022	Report Finalization
26-10-2022	Report Finalization
27-10-2022	Report Finalization
28-10-2022	Report Finalization
29-10-2022	Report Finalization
30-10-2022	Report Finalization
31-10-2022	Report Finalization

Meeting date and time with Supervisor(s)	
	1. 03/10/2022 at 1730 Hrs
	2. 10/10/2022 at 1730 Hrs
	3. 21/10/2022 at 1730 Hrs
	4

Signature of Student

Final Report For Capstone - Group 1

3 messages

AKANKSHA SINGH <asingh16_be19@thapar.edu> Tue, 6 Dec 2022 at 12:34
To: vsrivastava@thapar.edu, sk jain <skjain@thapar.edu>

VISHAL SRIVASTAVA <vsrivastava@thapar.edu> Tue, 6 Dec 2022 at 12:36
To: AKANKSHA SINGH <asingh16_be19@thapar.edu>

You can take a print out.

On Tue, Dec 6, 2022 at 12:34 PM AKANKSHA SINGH <asingh16_be19@thapar.edu> wrote:

SANJAY KUMAR JAIN <skjain@thapar.edu> Tue, 6 Dec 2022 at 12:38
To: AKANKSHA SINGH <asingh16_be19@thapar.edu>, Vishal Shrivastava <vshrivastava@thapar.edu>

Akanksha

I understand that you have incorporated all the suggestions. Please go ahead for print and submit it.

Regards

Sanjay

Dr. Sanjay K. Jain
Professor, EIED
Thapar Institute of Egg. & Tech. (TIET), Patiala
Punjab (India)

[Quoted text hidden]