JSS MAHAVIDYAPEETHA

JSS SCIENCE AND TECHNOLOGY UNIVERSITY

JSS TECHNICAL INSTITUTIONS CAMPUS

MYSURU – 570006 KARNATAKA, INDIA

A Project Report titled

"Automatic Speed Limit Detection and Vehicle Retardation System"

Submitted in partial fulfilment of the curriculum requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION

Submitted by

AMITH PRAKASH S P	**GAGAN S**
01JST16EC013	**01JST17EC407**
KEERTHAN S	**SUMUKH S**
01JST16EC042	**01JST16EC097**

Under the guidance of

Mr. Halesh M R

(ASSISTANT PROFESSOR)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

AUGUST 2020

Educate Elevate Enlighten

JSS Mahavidyapeetha

JSS Science And Technology University

(Established Under JSS Science and Technology University Act No. 43 of 2013)

(Formerly Known as SJCE)

JSS SCIENCE AND TECHNOLOGY UNIVERSITY MYSURU

CERTIFICATE

Certified that the project report entitled “Automatic Speed Limit Detection and Vehicle Retardation System” is a bonafide work carried out by **Amith Prakash S P** bearing USN **01JST16EC013, Gagan S** bearing USN **01JST17EC407, Keerthan S** bearing USN **01JST16EC042, and Sumukh S** bearing USN **01JST16EC097,** in partial fulfilment of the requirements for the award of **Bachelor of Engineering (BE)** in Electronics and Communication Engineering under JSS Science and Technology University (JSSS&TU) during the year 2019 - 2020.

....	....	....
Signature of the Guide	Signature of the HOD	Signature of Principal

Date :
Mysuru

Examiners

1.

2.

3.

Educate Elevate Enlighten

JSS Mahavidyapeetha

JSS Science And Technology University

(Established Under JSS Science and Technology University Act No. 43 of 2013)

(Formerly Known as SJCE)

JSS SCIENCE AND TECHNOLOGY UNIVERSITY MYSURU

DECLARATION

This is to certify that the dissertation entitled "Automatic Speed Limit Detection and Vehicle Retardation System" has been carried out by the undersigned and the same has not been submitted to any other University/institution elsewhere for the award of Degree/Diploma.

Date:
Place: Mysuru

Amith Prakash S P
(01JST16EC013)

Gagan S
(01JST17EC407)

Keerthan S
(01JST16EC042)

Sumukh S
(01JST16EC097)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accomplices the successful completion of any task would be incomplete without the mention of people who made it possible.

We take the opportunity to express our sincere thanks to following persons without whom this project would not have been possible and whose firm believe in our capabilities and moral boosting to bring out full potential towards the forefront has played a major role in each and every step of our endeavor to success.

First and Foremost we ought to pay due regards to our institution Sri Jayachamarajendra College of Engineering Mysuru,which provided us a great platform providing us an opportunity for carrying out this project work.

We are very much thankful to **Dr.T. N. Nagabhushan** , Principal,Sri Jayachamarajendra College of Engineering Mysuru. We are extremely grateful **Dr. Shankaraiah**, Professor and Head of the Department of Electronics and Communication Engineering,Sri Jayachamarajendra College of Engineering Mysuru.

We take this opportunity to express our gratitude and sincere thanks to our project guide **Halesh M R** , Assistant Professor , Department of Electronics and Com- munication Engineering,Sri Jayachamarajendra College of Engineering Mysuru , for his kind consent , sustained encouragement, timely valuable guidance ,immense help and co-operation throughout the tenure of the project.

We intended to express our sincere gratitude to all staff members and non teaching staff of Electronics and Communication Engineering, Sri Jayachamarajendra College of Engineering Mysuru, for their help and support.

Amith Prakash S P
Keerthan S
Sumukh S
Gagan S

ABSTRACT

With the increase in number of vehicles on road, there is also increase in the number of violations of traffic rules day by day. Though many of the of the drivers follow the rules, there are some people who doesn't abide to any such rules. One such case is not driving under speed-limit in speed-limit zone. Because of which, mishaps on the road are taking one's life and causing economic damage to the country. Speed-limit are set at certain areas for the reasons like banking of the roads, dense area etc. driving within the limit gives you lot time to respond to the unexpected events and control your vehicle.

Most of the time it is driver negligence towards the caution of speed limit board which leads to disasters. So, in this project we have implemented to automatically reduce the speed of the vehicle when it is in a speed limit zone and avoid collision with the rear vehicle. The designed system has a camera set up to read the sign and extract the speed limit from the board and alert the driver to decrease the speed, and if the driver doesn't slow down the vehicle the system will automatically reduce the speed and brings under the specified limit. The vehicle following the other may not have any idea about its front vehicles speed and reduction in speed can sometimes results in collision. To avoid such cases the system also has the capability of communicating with the rear vehicle system to inform regarding speed reduction and warns the driver to slow down to avoid collision. And if there are any other vehicles (having our system installed) behind the second vehicle the channel continues warning the drivers to slow down.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automobiles are one of the most important parts for the human life in the modern world. The development of automobiles effects on people's way of life throughout much of world. Probably no other discovery, invention and technological advance has created more changes in society. Automobiles have given many people incredible freedom of movement. The automobiles influence where people live and work and how they spend their time. The automobile itself has become safer over the years because of advance in its design and manufacturing process. Due to various advancements in the engineering field and the automobile industries, automobiles are now fast, efficient and comfortable. But in the recent years, many casualties and injuries were caused by reckless driving of automobiles. By not obeying traffic rules, some reckless drivers not only put their lives at risk but also the lives of other innocent passengers travelling in the same road. The government and law enforcing bodies decide on speed limits for areas of roads everywhere. These speed limits are then made into boards and placed along the side of all major roads. The drivers travelling in those areas are to strictly adhere to those set limits and keep their vehicles under that limit. Breaking those rules i.e. over-speeding is a finable offense. But due to poor enforcement of such rules in India, most of the drivers break these rules. It is difficult for the law enforcement officers to catch these culprits and punish or fine them. This can be due to unavailability of proper equipment also. This project deals with developing a system that detects these installed speed limit boards placed alongside the roads using a camera mounted on the car. The mounted camera constantly captures video and sends it to a control unit for processing. The ECU detects the speed limit board and interprets it to deduce the numerical speed. After this speed is interpreted, the ECU checks whether the current speed of the vehicle is more than the detected limit. If it is, the ECU send signals to the motor control of the electric car to reduce the speed gradually bringing it down to the prescribed limit. Simultaneously, the vehicle also sends out information regarding the detected speed limit to similarly equipped vehicles nearby. This also means that the control unit is also ready to receive similar signals from other vehicles to control its speed.

1.1 Motivation

Each year over **40,000** fatalities and **2,788,000** non-fatal injuries occur due to automobile accidents on the road. In addition, it is predicted that hospital bills, damaged properties and additional accident-related costs will add up to approximately one to three percent of the world's gross domestic product. This can have owed to the fact that most drivers of automobiles knowingly or unknowingly break the rules of traffic. The carelessness of these certain driver cost the world many valuable lives. Over-speeding is one the most broken traffic rule. This is the motivation behind this project. This project aims to create a system for electric cars through which the car reads speed limit boards placed by the government along the road and automatically limits the speed of the vehicle overriding the driver's control.

1.2 Problem Statement

Over speeding on roads is unlawful and also dangerous. This has been known to be the major cause in accidents and fatalities in highway accidents by a number of studies. A system is to be devised which automatically detects speed limit boards and limits the speed of the electric vehicle. The proposed system also involves attaching cameras to the front of a vehicle to detect and interpret the speed limit boards and gradually decelerate the vehicle while also communicating the obtained information to other vehicles in the vicinity. The challenge lies in real time detection and interpretation of the sign boards which is to be achieved using machine learning algorithms.

1.3 Objectives

- To design an automatic speed limiter system for electric cars that detects and interprets speed limiter signs placed alongside roads efficiently. The system is also supposed to communicate with other similar automobiles in its vicinity and inform them of the detected speed limit. Similarly, the system is supposed to receive similar information from other vehicles and limit its speed.

- To test the system for various test cases, verify its functionalities and find ways to fix its faults/errors is any.

1.4 Mode of Demonstration

The demonstration of the real time system is shown through a prototype models of two bots, where the two bots depicts the vehicle in the real time. Our main focus is to show the vehicle to vehicle communication upon detecting the speed limit board. The bots are interfaced with a camera, which captures the images and process the images. Based on the attributes taken out from the processed images the bots make the decision. The bot upon detecting the sign board calculates its current speed by using the speed sensor and then the speed is compared with the limiting speed of the board. If the speed of the bot exceeds the limiting speed, then the micro controller sends a signal to decelerate by controlling the PWM of the wheels. Then the corresponding is communicated to the rear bot which receives the data through Bluetooth and the motors will be actuated to the limiting speed. Hence ensuring safety to the vehicles behind.

1.5 Organization of the report

The complete report consists of five chapters where the first chapter encompasses the Synopsis like Introduction to the project, Motivation, Objective, Methodology, Problem statement and their solutions. The second chapter embodies the exhaustive Literature survey of the available road safety technologies and communication. The third Chapter comprises the block diagram, Software and Hardware Requirements. The fourth chapter includes flow-chart and detailed information about the implementation of our project. The fifth chapter includes the results and discussions of the implemented project along with the Conclusion and Future scope of this project.

Chapter 2

Literature Survey

To contribute to this field of development, it becomes important to understand and evaluate previously existing methods. Hence, the literature survey has been carried out for the project work where several authors and their contributions have been analyzed. The next section posits the previous researches and attempts on speed-limit board detection, vehicle retardation methods and inter-vehicular communication.

2.1 Previous Research

The distance between the vehicles was measured using an Ultrasonic sensor interfaced with the Arduino UNO [1] and alerts the driver to react to the condition. This paper in the beginning gives a detailed explanation on various types of collision avoidance and warning systems. It covers a wide range of techniques implemented or proposed. Their prototype also included GPS and GSM modules to track down the vehicle in case of any emergencies. This is initiated by the trigger of the airbags during accidents. Though the paper gave a good idea about collision avoidance, it didn't mention vehicle to vehicle communication in order to prevent collision.

A cost-effective sensor-based development of an embedded system [2] which alerts the driver to avoid collision by measuring the probability of collision and assisting him the steps to be taken in such situations. It also mentions the disadvantages of using image processing for this domain. The proposed method here is suitable even for existing vehicles without any additional infrastructure required. This proposed method also uses Arduino UNO and Ultrasonic sensors for detecting obstacles. The algorithm estimates the probability of collision by considering the vehicle's speed and its direction along with measuring the distance of the front obstacle. The distance between the vehicles is constantly monitored if it falls below the critical value then the algorithm would raise an alert for the driver.

Dedicated Short Range Communications (DSRC) [3] protocol allows cars on the road to form a dynamic wireless network that aid in driving with the overall goal of reducing car accidents. It treats every car as a router which allows for sending/receiving messages over multi-hop systems to/from other vehicles. The routing algorithm is based on the position of the vehicles and is able to handle fast changes of the network topology. The system is capable of making automatic or semi-automatic decisions, providing warnings. The DSRC is designed to create a short-range cellular network that is based on a 3G and 4G cellular network. In this scheme, every vehicle is also a router and the complete system acts as a dynamic wireless network.

Vehicle to Vehicle communication through LI-Fi technology [4] is developed in this research paper where both the vehicles in line can communicate and synchronize for necessary conditions. It presents the initial designs and results of a small-scale prototype of a vehicle to vehicle communication system using light fidelity (Li-Fi) technology, which still needs more investigations on its sustainability for outdoor vehicular networks. Vehicle to vehicle communication is that the best solution that has been utilized in order to scale back vehicles' accidents. The proposed use of Li-Fi technology during this paper comprises mainly LED (LED) bulbs as means of connectivity by sending data through the sunshine spectrum as an optical wireless medium for signal propagation.

An intelligent transport system (ITS); with optical vehicle-to-vehicle (V2V) provides a way to communicate with the vehicles. The ITS offers various applications which are targeted to scale back traffic accidents, facilitate emergency management and flow of traffic. V2V communication is possible using optical wireless communication (OWC) [5] technology. The vehicles and the roadside units act as communicating nodes, providing each-other the information regarding vehicles and infrastructure in the vicinity. The LED based optical wireless communication turns to be effective briefly range line-of-sight communication and thus overcoming the restrictions incurred thanks to isotropic nature of radio waves. However, there are significant constraints to OWC used for ITS, which are thanks to impairments like adverse atmospheric turbulence conditions which degrades the link performance thanks to irradiance fluctuations. The vehicles in ITS are equipped with optical transceivers. The rear end of the vehicle is assumed to have three directional transmitting apertures while the front end of the vehicle has three directional receiving apertures. The roadside nodes act as a relay used to forward the data received from the leading vehicle to the preceding vehicle.

In this paper, the author envisioned future scenarios that might come up if those devices within a vehicle are able to communicate and interact with each other. Therefore, discusses the characteristics of wireless infrastructure-based and ad-hoc networks for communication between mobile devices in vehicles in order to establish a local and integrated information system. Wireless communication systems are very interesting for communication in vehicles as the heterogeneity of devices bring a plethora of different plugs and interfaces for wired communication[6]. The paper describes the Bluetooth de-facto standard in detail, which is one encouraging technology for those future scenarios. Bluetooth is a communication technology which optimized for communication between small devices in mobile and wireless ad-hoc networks.

In this paper analysis has been done keeping in mind various perspectives of Bluetooth technology. The analysis starts with a description of the technology in terms of its network infrastructure, software and hardware. Then it is continued by the Error corrections and retransmission. The analysis is done on macro analytical view including the business implications, advantages of Bluetooth technology, its role in the global third generation (3G) wireless schemes. The finally it concludes with the applications and future potentials of Bluetooth[7].

Traffic signboards can detect in many ways, the author [8] proposes the Sobel edge detector as one of the important parameters to perform this operation along with smoothing images. This paper deals with the camera-based driver alert system of automatic traffic sign board identification which uses image processing techniques. The proposed system uses a camera, mounted on top of a vehicle and interfaced to the processor. The camera captures real-time road signs images, the processor processes these images using image processing techniques and gives the audio information about road signs to the driver. The main advantage of the proposed system is that drivers need not concentrate on the road signs while driving and due to which accidents can be avoided. The acquiring images are pre-processed and enriched with respect to different backgrounds, these include grayscale conversion, image resizing, cropping, histogram equalization and noise removal. To detect the region of interest Threshold segmentation, Sobel edge detection and blob detector methods are used. Once the image detects then it will compare with the pre-stored images in the database with the help of SVM classifier and displays the output.

The methodology explained by the author [9] is to detect and recognize traffic signs in video sequences recorded by an on-board vehicle camera. The techniques such as Gaussians filter and Canny edge detection are used to enhance the image, later on, get converted to a black and white mode. The binary threshold method is used to find out the threshold values of a black and white image and then converts into a binary image. Ellipse and contour drawing operations are performed in order to check whether the image is a traffic sign board or not. The detected image is resized and performs the grayscale conversion. The dominated pixels are compared with the set of trained images and compare with obtained results with feature vector classification. A fast real-time and robust automatic traffic sign detection and recognition can support and disburden the driver and also significantly increase driving safety and comfort. Automatic recognition of traffic signs is important for automated intelligent driving vehicle or driver assistance systems. This paper [9] presents a study to recognize traffic sign patterns using OpenCV technique.

The paper [10] presents an automatic traffic sign recognition system using the videos recorded from an on-board Dashcam. The images captured from the dashcam are processed with the histogram of oriented gradients to form feature vectors, followed by support vector machines to detect the traffic signs. A road sign detection and recognition algorithm based on the image gradient information. It adopts the histogram of gradients cooperated with the support vector machine to perform the initial traffic sign detection. The exact position of the traffic sign is then obtained by BCT, VBT and colour information. Finally, a neural network is adopted in order to identify the traffic sign information.

Traffic road sign detection and recognition are important to transport systems with robotic eyes or cameras while driving on the road. The paper [11] presents an overview of the traffic road signs detection and recognition. The signboards are detected by converting the image into grayscale and highlight the red pixels. Once the red pixels are identified then after it is verified as a signboard or not by breadth-first search method. If it is detected as a signboard, then the Adaboost algorithm is used as a classifier between the different sign boards and displays the results. The paper [11] is based upon a major approach to detect the direction. The main idea proposed by the paper [11] is to design and construct a computer-based system which can automatically detect the direction of the road sign.

A traffic sign recognition system with Circular Hough Transform (CHT) can aid in the development of Intelligent Speed Adaptation. The paper [12] is based on extracting the speed limit sign from the traffic scene with the aid of colour and non-colour information of the traffic sign. The digits of the speed limit sign board are then extracted and classified using SVM classifier which is trained for this purpose. In general, the system detects the prohibited traffic sign in the first place and then specifies whether the detected sign is a speed limit sign and also determines the allowed speed in case the detected sign is a speed limit sign.

There are several environmental conditions under which human beings do not find productive to work. So, there is a need for a machine that will not be affected by these conditions during the course of carrying out the human-like tasks. A prototype of a robotic vehicle for pick and place have been designed and implemented in this paper. The proposed system works by using a Bluetooth module for receiving the Bluetooth command being sent by the operator[13]. The method adopted in this study uses a DC motor to move the robotic vehicle to the appropriate direction using commands. The robotic car will move in the appropriate direction sent to him by the operator through the Bluetooth for execution purpose.

Intelligent Transportation Systems (ITS) have been created as a convergence of modern mobile computing, wireless networks and transportation technologies[14].Vehicular Ad hoc Networks (VANETs) is emerging as the backbone of ITS. In VANETs, vehicles are capable of sending, receiving and routing the traffic information. Vehicles communicate with each other using either Vehicle-toVehicle (V2V) communication using single hop or multi-hop communications.The goal in this work is to develop an automatic information dissemination framework for vehicular ad hoc network. It also present a comparative study of different short range wireless technologies including Wi-Fi (IEEE 802.11a/b/g/n), Bluetooth (IEEE 802.15.1), ZigBee (802.15.4), and Dedicated Short Range Communication (IEEE 802.11p) through on board devices which are suitable for Vehicular Ad hoc Networks (VANETs).

2.2 Summary of Literature Review

Paper NO.	PROS	CONS	REMARK
[1]	• Gave a detailed explanation on various collision avoidance system techniques. • Involved GPS and GSM as an extra benefit of alerted concerned personality in case of accident.	• Didn't't provide any mean to intercommunicate with the vehicles. • Ultra-sonic sensors couldn't detect slow moving vehicles.	We were able to understand various collision techniques implemented around the world. We gathered the idea from their prototype for building our prototype using Raspberry-Pi and Xbee.
[2]	• Developed an intelligent system supported by efficient algorithm to detect probability of collision happening instead of randomly alerting the driver. • The proposed idea also discusses the way in which a system could assist the driver in causing of collision.	• The proposed model had just a speed limit monitoring and alerted. There was no way to reduce speed automatically. • The vehicle under the collision avoidance had no way to communicate with rear vehicle to avoid chain collision if vehicle is within critical distance.	We were able to gather the advantages of algorithm they have implemented to avoid collision.
[3]	Mentions a detailed information about the V2V communication with their necessary security and privacy if a network of vehicles are connected.	Not widely deployed and Not suitable for existing vehicles, requires changes in vehicle architecture itself.	The paper seems to be a perfect model as they have developed a new protocol which can be implemented in upcoming new vehicles .

Paper NO.	PROS	CONS	REMARK
[4]	The usage of LED eliminates the need of complex wireless networks and protocols.	The use of Li-Fi restricts the range of communication to line of sight.	This paper showed that the vehicle to vehicle communication can also be done without creating any network. It discussed that, to avoid the collision it is enough to signal the rear vehicle and not essentially build a network.
[5]	• LED for OWC has greater advantages as they are of lower cost and has longer operating life. • The developed model proved to be a high energy efficient. • Camera is used as a optical receiver which provides the non-interference communication capability to OWC system.	The atmospheric conditions deteriorate the system performance also degrades.	This is one of the advance papers of this domain. It shows potential of becoming impart technology of self-driven cars as it seems to be an fine idea for communicating between vehicles.
[8]	Blob detector uses wavelet responses for feature description and this makes the image detection as more efficient compared to others.	Threshold segmentation is less efficient among other segmentation methods to simplify the representation of an image into something that is more a meaningful object of interest and easier to analyse.	Since the database is needed to store the pre-valued images this method is not efficient for the real-time video processing system.

[9]	The processing speed is higher since the images are resized frequently.	If the signboard is in triangle size, then this method won't detect a signboard.	This method works good in real-time application but due to the image resize operation, this effects in identifying the numbers in signboard during video processing.
[10]	SVM with the HOG feature helps to detect the traffic sign board irrespective of their background colour.	Since BCT and VBT are sensitive to the gradient orientation of edge points, the resized images with distortion might lead to poor results.	This method is best for video processing. Sobel edge detection method, bilateral Chinese transform (BCT) and vertex and bisector transform (VBT) algorithms are efficient in order to find signboards.
[11]	The breadth-first search method searches all the space area to locate the region of the area, this makes the searching process more efficient.	For real-time video processing, it is very difficult to process and search the data at every instant of time.	The algorithm that is used in the preprocessed unit is not so effective so this makes the identification difficult during night time. But the breadth-first search method makes the image processing efficient compare to others.
[12]	• There is no colour segmentation invoked. • Circular Hough Transform is invoked to detect the circular shapes.	Data set should be provided and it should be trained for several images taken at all conditions.	A simple and fast segmentation technique was invoked to separate the candidate traffic sign from the traffic scene and a circular Hough transform was employed to detect the prohibited traffic signs. Speed limit digits were segmented and their property curves were used to train SVM classifier.

Table 2.1: Summary of Literature Survey

Chapter 3

System Overview

3.1 Proposed block diagram

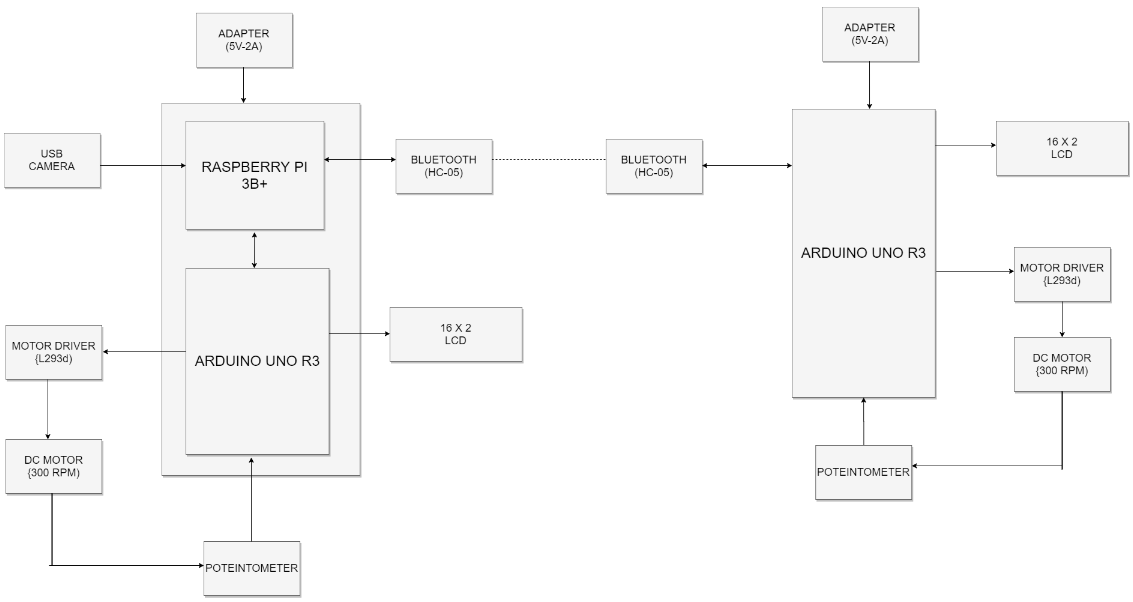


Figure 3.1: Proposed Block Diagram

In the above block diagram each module of Arduino and motors depicts as the dummy of vehicle. The camera input is constantly fed to the Raspberry pi on detecting the speed-limit sign board the image is captured and through image processing the speed-limit is identified and the speed of the motor is varied by altering the PMW. The speed is constantly read by potentiometer and it is displayed on the display for user knowledge. When the speed limit read is found to be less than the actual speed of the vehicle the Arduino initiates the retardation of vehicle and simultaneously sending signals to the rear vehicle through the Bluetooth. If there is any vehicle following the speed retarded

vehicle, it picks up the signal and the controller by varying the current flow to the motors decreases the vehicle speed in order to avoid collision.
The speed limiter board were placed near the road-side is detected and differentiated among other boards by the process of machine learning image classifier. The group of image data sets are collected and then they are trained with the help of python libraries. The trained images detect the sign board number. The number which is detected send to the Arduino through processor serially, this alerts the subsystem and then reduce the speed of the motor to a specified speed that is mentioned in the board. Once the speed is reducing then Bluetooth send the alert message to the preceding vehicle which contains Bluetooth to reduce the speed then the controller controls the motor speed and reduce the speed or just acknowledge it.

3.2 Hardware requirements

Raspberry Pi

The Raspberry Pi Foundation works to put the power of computing and digital making into the hands of people all over the world. It does this by providing low-cost, high-performance computers that people use to learn and solve problems. The Raspberry Pi is a very cheap credit card-sized single-board computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins that allows us to control electronic components for physical computing and explore the Internet of Things (IoT).

There have been four generations of Raspberry Pis: Pi 1, Pi 2, Pi 3and pi 4 and there has generally been a Model A and a Model B of most generations. All models use the same SoC (system on chip combined CPU and GPU). Central processing unit speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. The operating systems are stored in secure digital (SD) and program memory in either the SDHC or Micro SDHC sizes. Most boards have one to four USB slots, HDMI and composite video output. Lower level output is provided by a number of General purposes input output pins which support common protocols like IC. The Pi 3 has on board Bluetooth and Wi-Fi 802.11n.

- Broadcom BCM2837B0,Cortex-A53(ARMv8),64-bit SoC @ 1.4GHz,1GB LPDDR2 SDRAM.
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE.
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps).
- Extended 40-pin GPIO header.
- Full-size HDMI.
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera.

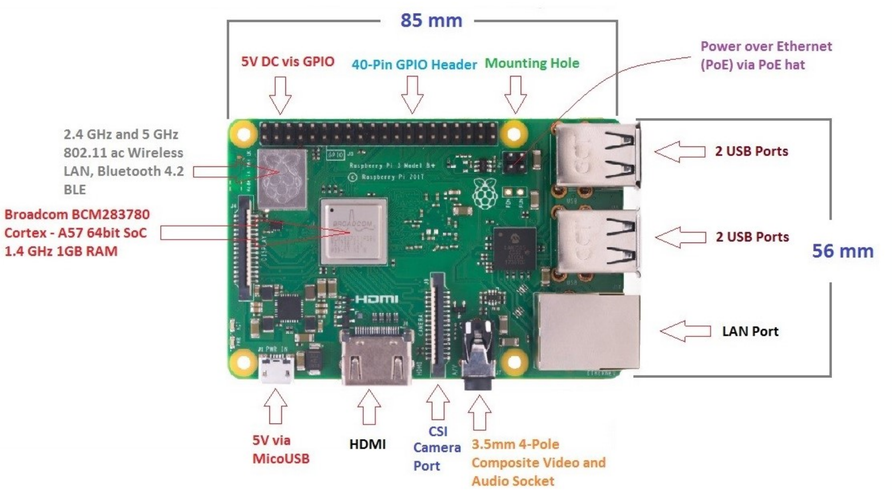


Figure 3.2: Raspberry pi 3B+

- 4-pole stereo output and composite video port.
- Micro SD port for loading your operating system and storing data.
- 5V/2.5A DC power input.
- Power-over-Ethernet (PoE) support (requires separate PoE HAT).

Arduino Uno R3

The Arduino has a large support community and an extensive set of support libraries and hardware add-on “shields, making it a great introductory platform for embedded electronics. The Arduino Uno R3 is a microcontroller board based on a removable, dual-inline-package (DIP) ATmega328 AVR microcontroller. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. Programs can be loaded on to it from the easy-to-use Arduino computer program.
It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs). The Arduino has an extensive support community, which makes it a very easy way to get started working with embedded electronics.

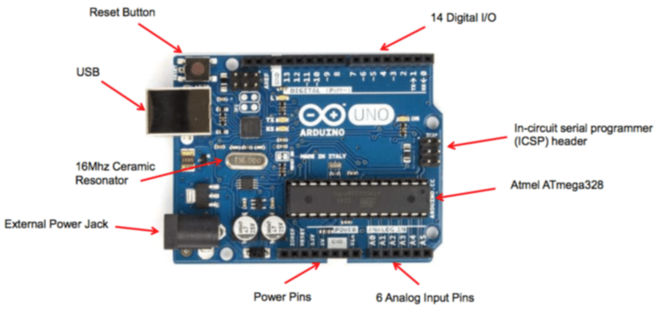


Figure 3.3: Arduino uno R3

Bluetooth(HC-05)

Bluetooth is a popular low powered wireless technology that is designed for sharing data between two devices over a short distance. HC-05 is a Bluetooth device used for wireless communication with Bluetooth enabled devices (like smartphone). It communicates with micro controllers using serial communication (USART).Default settings of HC-05 Bluetooth module can be changed using certain AT commands.As HC-05 Bluetooth module has 3.3 V level for RX/TX and micro controller can detect 3.3 V level.

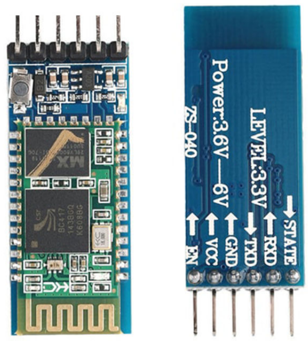


Figure 3.4: HC-05 Bluetooth

Motor Driver(L298N)

A motor controller is an electronic device that helps microcontroller to control the motor. Motor controller acts as an intermediate device between a microcontroller, a power supply or batteries, and the motors.Although the microcontroller decides the speed and direction of the motors, it cannot drive them directly because of its very limited power (current and voltage) output. The motor controller, on the other hand, can provide the current at the required voltage but cannot decide how the motor should run.Thus,

the microcontroller and the motor controller have to work together in order to make the motors move appropriately. Usually, the microcontroller can instruct the motor controller on how to power the motors via a standard and simple communication method such as UART or PWM. Also, some motor controllers can be manually controlled by an analogue voltage (usually created with a potentiometer).The physical size and weight of a motor controller can vary significantly. The size of a motor controller is usually related to the maximum current it can provide. Larger current means larger size.
L298N is called as a dual bidirectional motor driver which is based on dual H-Bridge Motor driver IC. The L298 is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It controls two dc motors independently in either direction and it is easy to use and interface the L298 with an Arduino or a Raspberry Pi. It also provides an onboard 5V regulator. It is a high voltage, high current dual full-bridge driver and designed to accept standard TTL logic levels and drive inductive loads such as DC and stepping motors, relays.

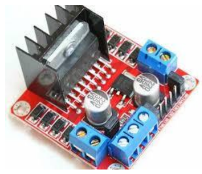


Figure 3.5: L298N Motor Driver

DC Motor

motors are used to actuate wheels, legs, tracks, arms, fingers, sensor turrets, camera, or weapon systems in robot and also used for locomotion of wheels in autonomous industry. Motors and actuators are the devices which make the vehicles movable. Motors and actuators convert electrical energy into physical motion.There are different types of electric motors but here we use brushed dc motors for movement of vehicles. A brushed DC motor is one which uses two brushes to conduct current from source to armature.The electric motor operated by dc is called DC motor.
Features:

- 300RPM 12V DC motors with Gearbox.
- 125gm weight.
- 0.35kgcm torque.

Figure 3.6: 300RPM DC Motor

Liquid Crystal Display(LCD)

Computers, calculators, television sets, mobile phones, digital watches use some kind of display to display the time.
An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 162 LCD display is a very basic module commonly used in DIYs and circuits. The 162 translates o a display 16 characters per line in 2 such lines.we are using LCD with each character is displayed in a 57 pixel matrix.

figure

LCD

Potentiometer
A Potentiometer (also known as a pot or potmeter) is defined as a 3 terminal variable resistor in which the resistance is manually varied to control the flow of electric current. A potentiometer acts as an adjustable voltage divider. Potentiometers work by varying the position of a sliding contact across a uniform resistance. we will be able to control the speed of DC motor with potentiometer and we can adjust the speed by rotating the knob of Potentiometer with the help of PWM.

Figure 3.7: Potentiometer

USB Webcam

A webcam is a video camera that feeds or streams an image or video in real time to or through a computer to a computer network
Image resolution interpolated to 25 mega pixels with 6 light sensors ; 16 MP Image Resolution ; USB Interface ; Night Vision ; USB Cable Length: 1m ; Focus Range: 4 cm to infinity:

- Anti-flicker 50Hz, 60Hz or outdoor.
- Resolution hardware: 500K pixels.
- Image quality: RGB24 or I42.

Figure 3.8: USB Camera

3.3 Software requirements

Python

Python is an interpreted high-level programming language for general-purpose programming.Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management.It supports multiple programming paradigms, including object-oriented, imperative,functional and procedural, and has a large and comprehensive standard library.Python is a programming language that lets us work quickly and integrate systems more.
Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Keras, TensorFlow, and Scikit-learn for machine learning.
- NumPy for high-performance scientific computing and data analysis.
- SciPy for advanced computing.
- Pandas for general-purpose data analysis.

- Seaborn for data visualization.
- Scikit-learn features various classification, regression, and clustering algorithms, including support vector machines.

PuTTy

PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port.

Raspbian

The operating systems Linux OS or Raspbian available for Raspberry Pi, Raspbian comes out on top as being the most user-friendly, best-looking, has the best range of default software and optimized for the Raspberry Pi hardware.Raspbian is a free operating system based on Debian GNU/Linux and optimised for the Raspberry Pi hardware. Raspbian comes with over 35,000 packages, or pre-compiled software bundled in a nice format for easy installation on a Raspberry Pi.

Ardunio IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

3.3.1 Libraries

Opencv

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.
The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image

database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

TensorFlow

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9, 2015. TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.. It is a symbolic math library, and is also used for machine learning applications such as neural networks. TensorFlow API is used to develop and train machine learning models. TensorFlow APIs are arranged hierarchically, with the high-level APIs built on the low-level APIs. For Machine learning researche , the low-level API are used to create and explore new machine learning algorithms. A high-level API named tf.keras to define and train machine learning models and to make predictions. tf.keras is the TensorFlow variant of the open-source Keras API.

Keras

Keras is an open source neural network library written in Python. It is capable of running on top of TensorFlow or Theano. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is Francois Chollet, a Google engineer. Keras libraries were installed using TensorFlow as backhend by us. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel. In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.
Keras abstracts away much of the complexity of building a deep neural network, leaving us with a very simple, nice, and easy to use interface to rapidly build, test, and deploy deep learning architectures. When it comes to Keras you have two choices for a backend engine - either TensorFlow or Theano. Theano is older than TensorFlow and was originally the only choice when selecting a backend for Keras. We are using TensorFlow as it isextremely flexible, allowing to deploy network computation to multiple CPUs, GPUs, servers, or even mobile systems without having to change a single line of code. This makes TensorFlowan excellent choice for training distributed deep learning networks in an architecture agnostic way, something that Theano does not (currently) provide.
Advantages of using keras in deep learning:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolution networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms..It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB. SciPy makes use of Matplotlib. Matplotlib 2.0.x supports Python versions 2.7 through 3.6. Pyplot is a Matplotlib module which provides a MATLAB-like interface. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source.

Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.NumPy can also be used as an efficient multi-dimensional container of generic data. NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions.
- Tools for integrating C/C++ and Fortran code.
- useful linear algebra, Fourier transform, and random number capabilities.

Pandas

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language.
Pandas is well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet.
- Ordered and unordered (not necessarily fixed-frequency) time series data.
- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels.
- Any other form of observational / statistical data sets. The data actually need not be labeled at all to be placed into a pandas data structure.

Chapter 4

Design and Implementation

This section explains the design and implementation of our project.Initially, with the help of camera and code-dumped processor we detect the speed limit board number and then data is transferred serially to the microcontroller.The speed of the vehicle is calculated using sensors and compared with the results of processor data. Depend on results, poteintometer control the speed of motor through motor driver and microcontroller. and then with the help of Bluetooth, warning message is been acknowledged to preceding vehicle and also controls its motor.

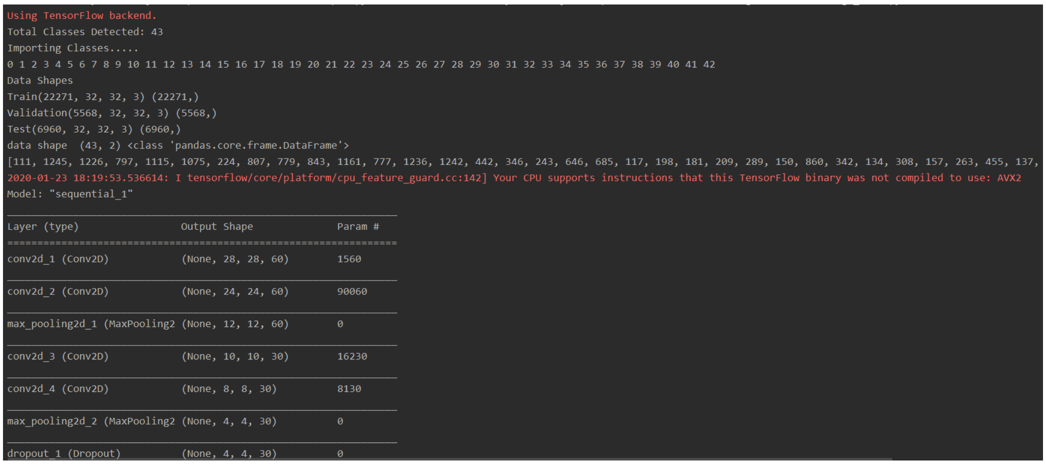
4.1 Speed Limit board detection

The Heart of the whole project is image processing part that is speed limit sign number detection. Normal contour drawing and segmentation in not enough to detect the board numbers, since there are different types of traffic sign boards in respective to signs, numbers, shapes, etc... and also the accuracy of an image processing part will get reduced.Hence we use CNN and ML to detect the numbers.
Our approach to building this speed limit sign classification model is discussed in four steps:

- Explore the dataset
- Build a CNN model
- Train and validate the model
- Test the model with test dataset

4.1.1 Explore the dataset

The detection and classification of the speed signs is performed by a completely trainable CNN. rather than specializing in improved features and algorithms of image processing, we focused on learning from an out-sized dataset.



```
Using TensorFlow backend.
Total Classes Detected: 43
Importing Classes.....
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
Data Shapes
Train(22271, 32, 32, 3) (22271,)
Validation(5568, 32, 32, 3) (5568,)
Test(6960, 32, 32, 3) (6960,)
data shape  (43, 2) <class 'pandas.core.frame.DataFrame'>
[111, 1245, 1226, 797, 1115, 1075, 224, 807, 779, 843, 1161, 777, 1236, 1242, 442, 346, 243, 646, 685, 117, 198, 181, 209, 289, 150, 860, 342, 134, 308, 157, 263, 455, 137,
2020-01-23 18:19:53.536614: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 60)        1560
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 60)        90060
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 60)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 30)        16230
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 30)          8130
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 30)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 4, 30)          0
```

Figure 4.1: Dataset Characteristics

The dataset consists of 20000 images with 12 different classes. The images are distributed unevenly between these classes rely on the parameter and hence the model may predict some classes more accurately than other classes. The total size of the dataset is around 300 MB. The dataset has a train folder which contains images inside each class and a test folder which we will use for testing our model and 20% of training data sets are used in test folder.Validation folder is used to check the training working process. The dataset is splits into training, test and validation sets, with the following characteristics:

- Images are 32 (width) x 32 (height) x 3 (RGB colour channels).
- Training set is composed of 22271 images.
- Validation set is composed 5568 of images.
- Test set is composed of 6960 images.
- There are 12 classes (e.g. Speed Limit 20km/h, Background, etc.).

Populating the dataset with various image modifying techniques such as rotation, colour distortion or blurring the image and training the original dataset results in high accuracy.

Figure 4.2: Speed limit board numbers dataset

The classes of training image dataset are distributed as:

CLASS	TRAINING	TESTING	TOTAL
Background	2800	700	3500
20 Km/hr max	162	18	180
30 Km/hr max	1584	396	1980
40 Km/hr max	1120	280	1400
50 Km/hr max	1608	402	2010
60 Km/hr max	1200	300	1500
70 Km/hr max	1346	354	1770
80 Km/hr max	1320	330	1650
90 Km/hr max	1660	404	2020
100 Km/hr max	1032	258	1290
110 Km/hr max	1200	300	1500
120 Km/hr max	1008	252	1260
			22700

Table 4.1: Dataset Numbers

Training with real-world images with natural variations results in invariant to light conditions, small scale differences and small rotations. Hence, real world images are collected. Using only speed limit sign image is not sufficient to train; a good set of background images is also required. A representative background training class prevents false detections .Background pictures like road side scenes, traffic sign boards are collected and is therefore it is important.

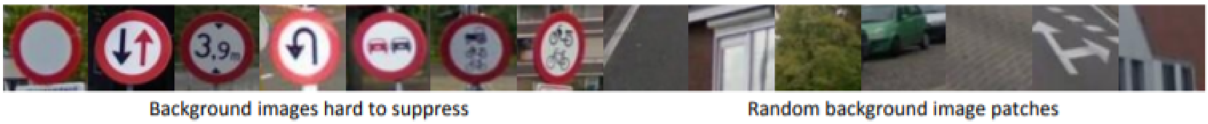


Figure 4.3: Background image datasets

The Images were chosen because of the following:

- They represent different traffic signs that we currently classify.
- They vary in shape and color.
- They are under different lighting conditions (the 4th one has sunlight reflection).
- hey are under different orientations (the 3rd one is slanted).
- They have different background.
- The last image is actually a design, not a real picture, and we wanted to test the model against it.

4.1.2 Build CNN Model

Our image processing pipeline consists of following steps: convolution, max-pooling, and finally a fully-connected network. The essential idea is to start with a large image and continually boil it down, step-by-step, until it incorporates a single result. The more convolution steps, the model learns more. the primary convolution step is to learn to recognize shape of boards, the second convolution step might recognize single number using its knowledge of shape, the third step might recognize final number using its knowledge of individual numbers, etc.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 60)        1560
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 60)        90060
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 60)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 30)        16230
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 30)          8130
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 30)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 4, 30)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 480)               0
_________________________________________________________________
dense_1 (Dense)              (None, 500)               240500
_________________________________________________________________
dropout_2 (Dropout)          (None, 500)               0
_________________________________________________________________
dense_2 (Dense)              (None, 43)                21543
=================================================================
Total params: 378,023
Trainable params: 378,023
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.4: CNN models

We initially apply two pre-processing steps to our images:

- **Grayscale**
  We convert our 3 channel image to a single grayscale image so that even day/night images can be able to detect.

- **Image normalization**
  We center the distribution of the image dataset by subtracting each image by the dataset mean and divide by its variance. This helps our model treating images uniformly. One of the constraints of the CNN model is that they can't be trained on a different dimensions of images. So, it's mandatory to possess same dimension images within the dataset. So it converted to same size using imagedatagenerator.

Architecture

The network is composed of 3 convolution layers — kernel size is 3x3, with depth doubling at next layer — using ReLU as the activation function, each followed by a 2x2 max pooling operation. The last 3 layers are fully connected, with the final layer producing 12 results (the total number of possible classes) are computed.

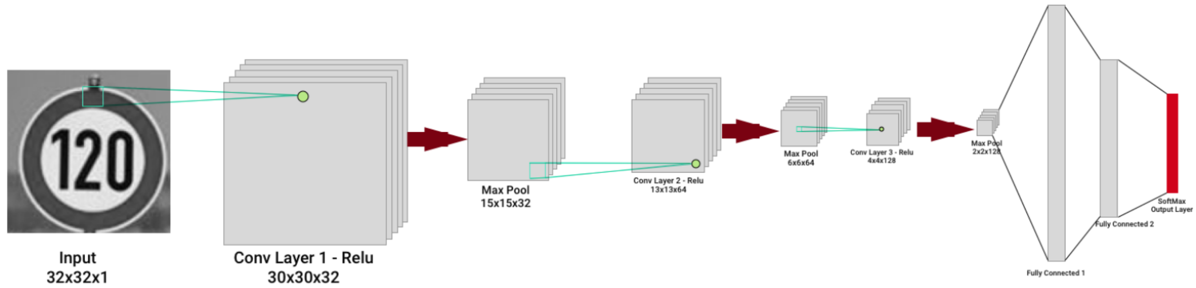


Figure 4.5: Architecture of CNN model

How convolution works?

1. Break the image into overlapping image tiles.
2. Feed each image tile into a small neural network.
3. Save the results from each tile into a new array.
4. Down sampling
5. Make a prediction

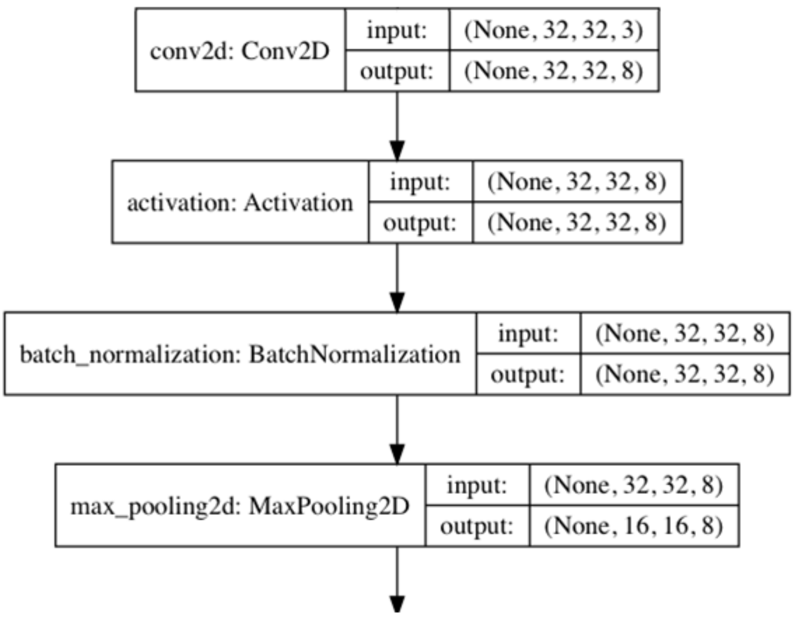


Figure 4.6: Framework of CNN model

The architecture of our model is:

- 2 Conv2D layer (filter=32, kernel size=(5,5), activation="relu")
- MaxPool2D layer (pool size=(2,2))
- Dropout layer (rate=0.25)
- 2 Conv2D layer (filter=64, kernel size=(3,3), activation="relu")
- MaxPool2D layer (pool size=(2,2))
- Dropout layer (rate=0.25)
- Flatten layer to squeeze the layers into 1 dimension
- Dense Fully connected layer (256 nodes, activation="relu")
- Dropout layer (rate=0.5)
- Dense layer (43 nodes, activation="softmax")

4.1.3 Training the dataset

The training algorithm that is used to learn the CNN coefficients is the **on-line mode of error back-propagation**. The main purpose to choose of this training algorithm is that it performs well with large datasets of training data. The basic idea of error back propagation is to efficiently calculate the partial derivatives of the output error in function of the resulted weights for a given input pattern. These partial derivatives are used to perform small corrections to the weights to the negative direction of the error derivatives.if training data is prepared before the training procedure then that dataset is considered for training. First the dataset is converted to greyscale images; the main motivation is the fact that colour representations are not consistent between day and night, smog and dusty conditions. For training CNN, a fixed size window as input layer is required. In the experiment a 32x32 pixel input is used; this results in a final application that can detect speed-limit signs of 32x32 pixels.
Steps for training datset:

1. Loading our training and testing split from the dataset.
2. Pre-processing the images.
3. Training our model.
4. Evaluating our model's accuracy.
5. Serializing the model to disk so that we can use it later to make predictions on new traffic sign data.

We also noticed that some images in the test set are distored.We are therefore used data augmentation techniques in an attempt to:

- Extend dataset and provide additional pictures in different lighting settings and orientations.
- Improve model's ability to become more generic.
- Improve test and validation accuracy, specifically on distorted images.

The distribution of images does not change significantly, but we do apply grayscale, histogram equalization and normalisation pre-processing steps to our images so that the accuracy will improved even in different lighting settings.We train for 2000 epochs with dropout (p-conv=0.6, p-fc=0.5) and achieve **97.86%** accuracy on the test set.

4.1.4 Testing data set

Running trained CNN scripts in the first time for captured video, it is found to be having lot of false detections. And the reason was that, there are infinite number of patterns missing from the background class and it was not possible to include all the patterns with in the class. The solution to avoid mistakes in detection was to include iterative boosting algorithm to recognise the pattern. The basic principle of this algorithm is to consider only those patterns which are recognized incorrectly and appending to the training set that makes the system to learn from its past errors. To automate the boosting algorithm a set of new dataset with empty road signs are collected. Initially for the algorithm the test was ran on newly collected dataset with large threshold. The threshold was selected such that the algorithm detects only those images with probability of 0.9. The details to the image path and class files, are kept in test.csv file and test folders are placed together as a dataset. Using pandas, we extract the path to image and class labels. NumPy array consists of all the pixels data of the image extracted and resized to 30x30 pixels, which was reduced from the original size in-order to predict the model. **97.86%** accuracy was achieved in this model, and for this we have imported accuracy score from sklearn.metrics and observed the model for its prediction on the class labels.

4.2 Sensing and Retardation

The detected sign board number using image processing is send to the Arduino serially and displayed on LCD. As per the detected number, microcontroller reduces the speed of the motor using poteintometer with the help of PWM technique. once the speed reduced, the reduced speed will show on LCD.
DC motors are used for movement of wheels. DC motor can be control by the microcontroller but it must not connect directly to Arduino because it can burn hence motor driver are used to control the motor. L293 is a dual H bridge IC motor driver. So, by this microcontroller two motors can be control by the single IC.

Input 1	Input 2	Result
0	0	Stop
0	1	Anti-clockwise
1	0	Clockwise
1	1	Stop

Table 4.2: Motor Driver operations

Rotation of motor depends on Enable Pins. When Enable 1/2 is HIGH , motor connected to left part of IC will rotate according to following manner:
The motors and Motor Driver connection is as follows:
Pin 1: When Enable1/2 is HIGH, Left part of IC will work, i.e motor connected with pin 3 and pin 6 will rotate.
Pin 2: Input 1, when this pin is HIGH the current will flow though output 1.
Pin 3: Output 1, this pin is connected with one terminal of motor.
Pin 4/5: GND pins
Pin 6: Output 2, this pin is connected with one terminal of motor.
Pin 7: Input 2, when this pin is HIGH the current will flow though output 2.
Pin 8: VSS, this pin is used to give power supply to connected motors from 5V to 36V maximum depends on Motor connected.
Pin 9: When Enable 3/4 is HIGH, Right part of IC will work, i.e motor connected with pin 11 and pin 14 will rotate.
Pin 10: Input 4, when this pin is HIGH the current will flow though output 4.
Pin 11: Output 4, this pin is connected with one terminal of motor.
Pin 12/13: GND pins
Pin 14: Output 3, this pin is connected with one terminal of motor.
Pin 15: Input 3, when this pin is HIGH the current will flow though output 3.
Pin 16: VCC, for supply power to IC i.e 5V.

Connections of Arduino and motor driver connection is as follows:

1. Module 5V (VCC) – Arduino 5V.
2. Module GND – Arduino GND.
3. Module 1 – Arduino D8.
4. Module 2 – Arduino D9.
5. Module 3 – Arduino D10.
6. Module 4 – Arduino D11.
7. Module Motor terminals – DC motors.
8. 8. Module VSS power terminal- External power source of 9V.

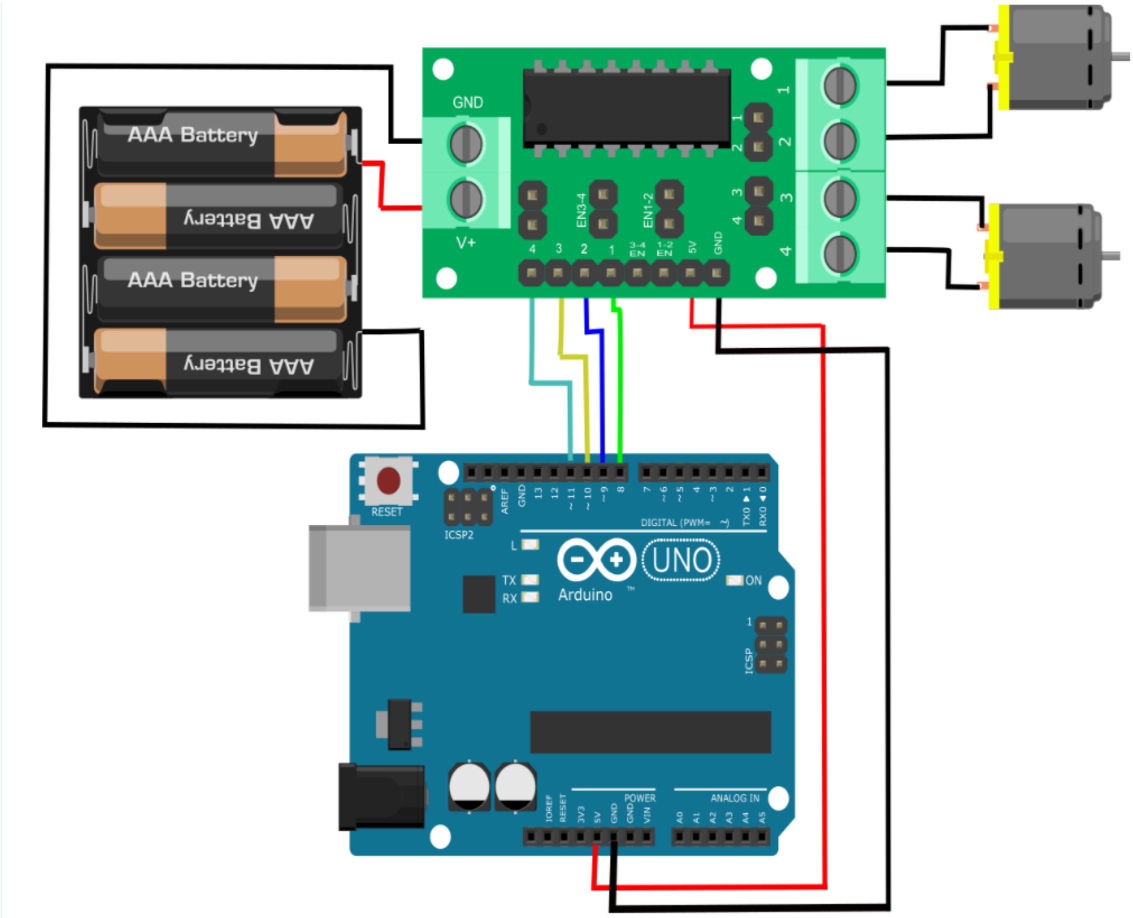


Figure 4.7: Motors-Motor Driver-Arduino Connection

The speed of DC motor can be controlled by potentiometer, PWM technique is used to control the speed of the dc motor. PWM is a technique by using we can control the voltage or power.

% Duty cycle = (TON/(TON + TOFF)) *100
Where, TON = HIGH time of the square wave
TOFF = LOW time of square wave

In this circuit, for controlling the speed of DC motor, we use a 10K ohm potentiometer to change the duty cycle of the PWM signal. potentiometer is connected to the analog input pin A0 of the Arduino UNO and the DC motor is connected to the 12th pin of the Arduino (which is the PWM pin). The working of Arduino program is works as follows, as it reads the voltage from the analog pin A0. The voltage at analog pin is varied by using the potentiometer. After doing PWM calculation, the duty cycle is adjusted according to it. And then updated data is displayed on LCD.

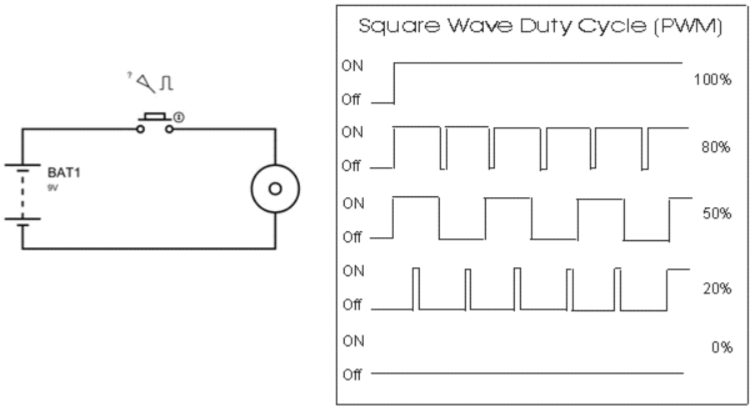


Figure 4.8: Pulse Width Modulation(PWM)

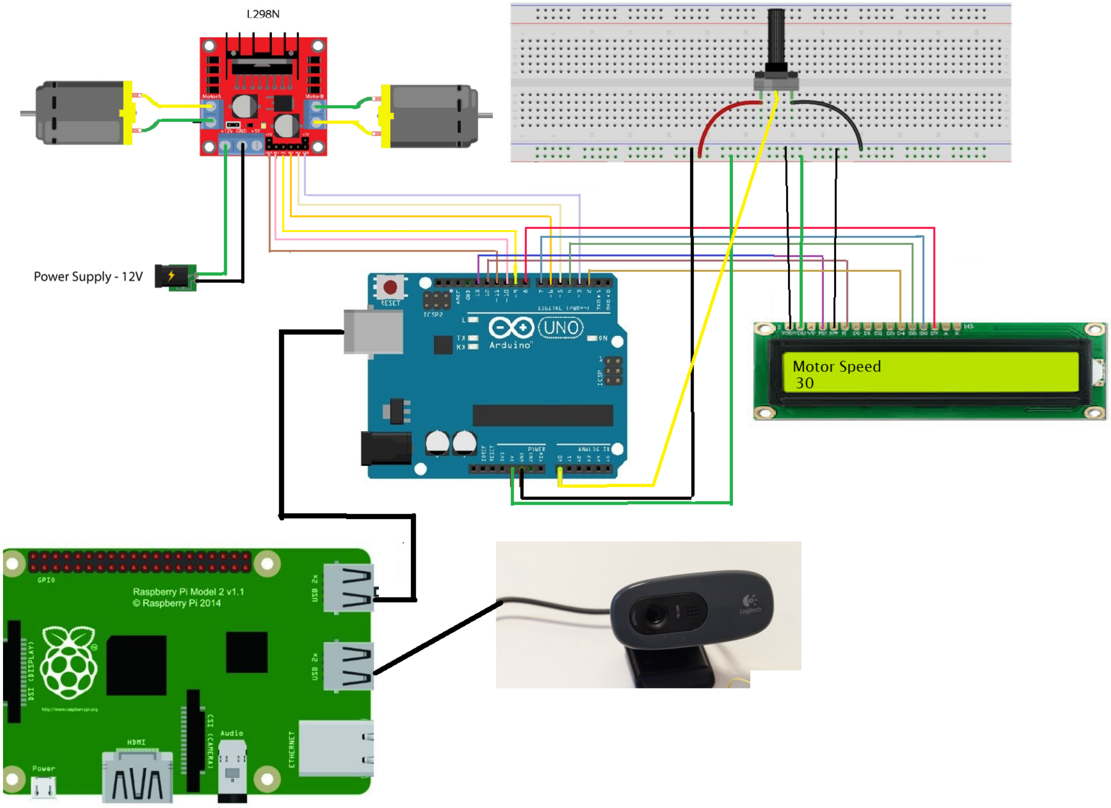


Figure 4.9: Sensing and Retardation circuit connection

4.3 Inter vehicular communication

Bluetooth is a wireless technology standard for exchanging data between devices.Bluetooth operates at frequencies between 2402 and 2480 MHz, or 2400 and 2483.5 MHz including guard bands 2 MHz wide at the bottom end and 3.5 MHz wide at the top.Bluetooth uses a radio technology called frequency-hopping spread spectrum. Bluetooth divides transmitted data into packets and transmits each packet on one of 79 designated Bluetooth channels. Each channel has a bandwidth of 1 MHz. It usually performs 1600 hops per second, with adaptive frequency-hopping (AFH) enabled.Bluetooth Low Energy uses 2 MHz spacing, which accommodates 40 channels.
Bluetooth is a packet-based protocol with a master/slave architecture. One master may communicate with up to seven slaves in a piconet. All devices share the master's clock. Packet exchange is based on the basic clock, defined by the master, which ticks at 312.5 microseconds intervals. Two clock ticks make up a slot of 625 microseconds, and two slots make up a slot pair of 1250 microseconds. In the simple case of single-slot packets, the master transmits in even slots and receives in odd slots. The slave, conversely, receives in even slots and transmits in odd slots. Packets may be 1, 3 or 5 slots long, but in all cases the master's transmission.
The range is dependent on the transmission power which is divided into four classes as: begins in even slots and the slave's in odd slots.

Max Permitted Power			
Class	(mW)	(dBm)	(m)
1	100	20	100
2	2.5	4	10
3	1	0	1
4	0.25	-3	0.5

Table 4.3: Bluetooth Classes,Power and Range

4.3.1 Bluetooth Connection Overview

The procedure involved in connecting a Bluetooth slave and a Master is described in this section.

- **Inquiry** The purpose of the Inquiry process is for a Bluetooth Master to check all the Bluetooth Slaves that are within its radio range that might provide some interesting Service. A Bluetooth Classic Slave sits in state called Inquiry Scan - i.e. a listening only state - until it hears a Bluetooth Master broadcast an Inquiry Request message. The Slave Application is responsible for putting the Stack into the Inquiry Scan state using the correct Stack API. Upon hearing an Inquiry Request the Slave will broadcast an Extended Inquiry Response (EIR) packet that contains its Name, Bluetooth Address (BDADR) and list of Services. These responses are handled completely by the Controller part of the Stack.

- **Page / Connect** The Paging process is used for a Bluetooth Master to connect to a Bluetooth Slave. The Master is "Paging" the Slave device. A Bluetooth Classic Slave sits in state called Page Scan - i.e. a listening only state - until a Bluetooth Master initiates the connection process by sending a Page Request. The Slave is responsible for putting the Stack into the Page Scan state using the correct Stack API. A Slave can - and often will be - in both the Page Scan and Inquiry Scan modes at the same time, meaning a Master can initiate a connection to a Slave without Inquiring if it already knows of the existence of the Slave from a previous connection.

- **Discover the Services using Service Discovery Protocol (SDP)** A simple conceptual model of a Bluetooth Classic device is a Server that is running one or more Services that are attached to Ports. This is the same model that we use in IP Networking. The SDP has a database embedded in it that contains a list of Services and what Port each one is running on. The SDP Protocol allows both sides of a connection to query the SDP database.

- **Pair and Bond** The whole Bluetooth communication system depends on having a shared symmetric encryption key called the Link Key. Bluetooth Classic uses a process called Secure Simple Pairing that exchanges enough information for the Link Key to be created. (There are other legacy Pairing methods, but they are largely obsolete at this point). The Secure Simple Pairing process was designed to minimize the chances that the communication link could be compromised by an eavesdropper or by a man-in-the-middle. Bonding is just saving the BDADDR/Link Key into non-volatile memory so that it can be reused to speed up re-initiating a connection.

- **Exchange Data with the Serial Port Profile** Once Service Discovery is complete, the Bluetooth Master knows the Port number that it should use to connect to the Serial Port Pro

  le (SPP). The SPP is just one of these Servers (from the last section) that acts like a serial port. The Bluetooth Master then opens a connection to the SPP Server running on the Bluetooth Slave.

Chapter 5

Results and Discussions

This chapter deals with the discussion about the results obtained from the demo of the project created. It talks about the hardware and the software results along while also discussing its performance, accuracy, merits and demerits. The software results include the generation of the convolution neural network using training set, as well as detection of the speed limit board in real time. The hardware results describe the demo setup and its performance.

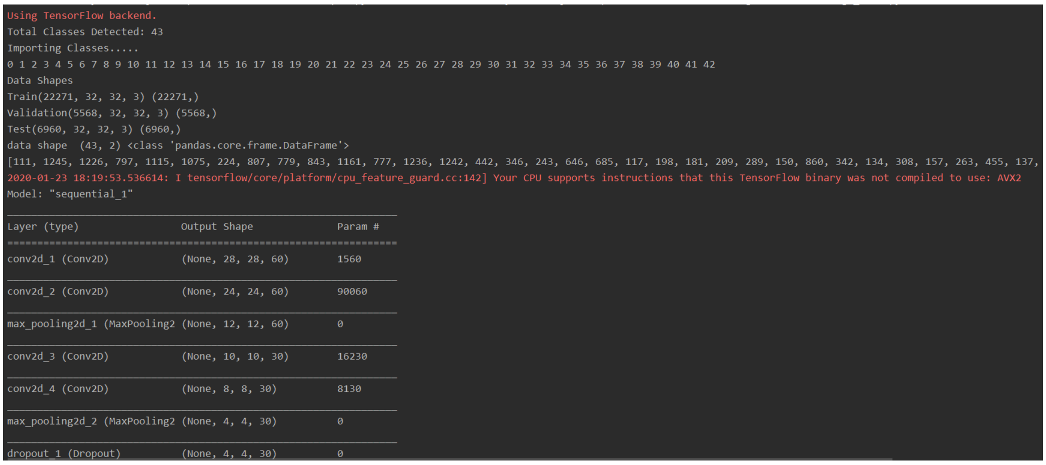
5.1 Software Results

5.1.1 Testing and validation

As explained in the earlier chapters, the python file test.py was used to generate a convolution neural network and generate a ".p" file. This file's functions were called upon by the main.py file to detect and interpret a speed limit board that the camera captured. Around 20000 images were used to train the network. These images were of speed limit boards of various values captured from various orientations and in different external conditions so as to increase the spectrum of detection as well as enhance its efficiency.

The Keras library is integrated to TensorFlow library and it is used in the backend to generate the neural network. Various classes or types of speed limit boards are defined and the required dataset for these classes are acquired and stored in the required directory. When the code starts to run, the number of classes is detected and displayed. And then the dataset for these classes stored as images is imported into the code. As seen in image 5.1, a total of 22271 images were used as dataset to train this network.

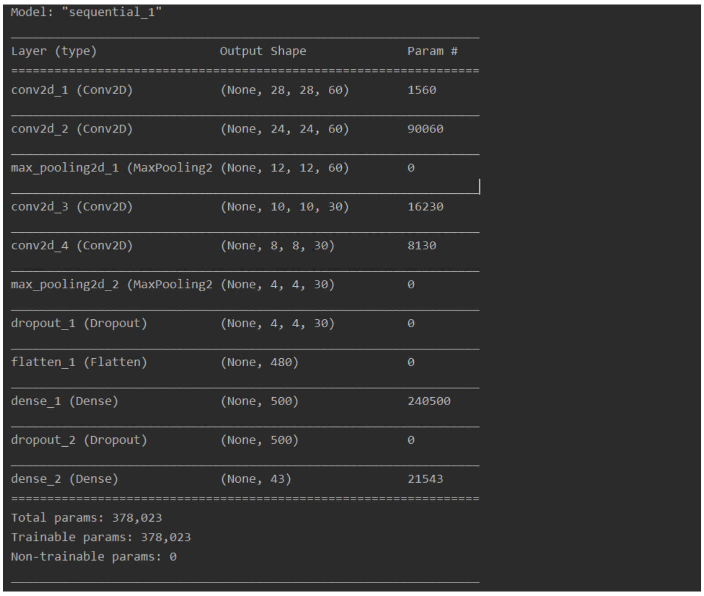
Test and validation ratios are set in the program before. These ratios determine what percentage of the data set it used to test and validate the neural network respectively. The test dataset provides the standard which is utilized to evaluate the model. It is only used once a model is completely trained. The test set is usually what is used to compare contemporary models. The test set is generally well curated. It contains care-



```
Using TensorFlow backend.
Total Classes Detected: 43
Importing Classes.....
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
Data Shapes
Train(22271, 32, 32, 3) (22271,)
Validation(5568, 32, 32, 3) (5568,)
Test(6960, 32, 32, 3) (6960,)
data shape  (43, 2) <class 'pandas.core.frame.DataFrame'>
[111, 1245, 1226, 797, 1115, 1075, 224, 807, 779, 843, 1161, 777, 1236, 1242, 442, 346, 243, 646, 685, 117, 198, 181, 209, 289, 150, 860, 342, 134, 308, 157, 263, 455, 137,
2020-01-23 18:19:53.536614: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 60)        1560
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 60)        90060
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 60)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 30)        16230
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 30)          8130
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 30)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 4, 30)          0
```

Figure 5.1: Console output during training dataset

fully sampled data that spans the various classes that the model would face. Validation set is the sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model parameters. The validation set is used to evaluate a given model, but this is for frequent evaluation. The validation set affects a model only by signifying whether its right or wrong. It does not add to the network's intelligence. Here instead of curating a different set, a ratio of the training sequences were used to test and validate the network. Both of these ratios were set to be 0.2. We may increase the ratio for more accuracy but it comes at a cost of a bigger, more complex network.



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 60)        1560
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 60)        90060
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 60)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 30)        16230
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 30)          8130
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 30)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 4, 30)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 480)               0
_________________________________________________________________
dense_1 (Dense)              (None, 500)               240500
_________________________________________________________________
dropout_2 (Dropout)          (None, 500)               0
_________________________________________________________________
dense_2 (Dense)              (None, 43)                21543
=================================================================
Total params: 378,023
Trainable params: 378,023
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.2: Summary of dataset imported

The training set images are converted to grayscale, equalized and normalized for training. A sample image from each of this class is displayed for us to ensure the processes have been working properly. This is depicted in figure 5.3.

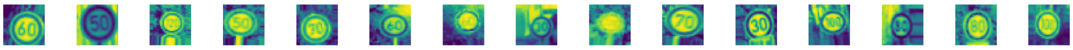


Figure 5.3: Grayscale conversion

Then the training starts. This process is also simultaneously displayed on the console port with accuracy values. This is as seen in figure 5.4.

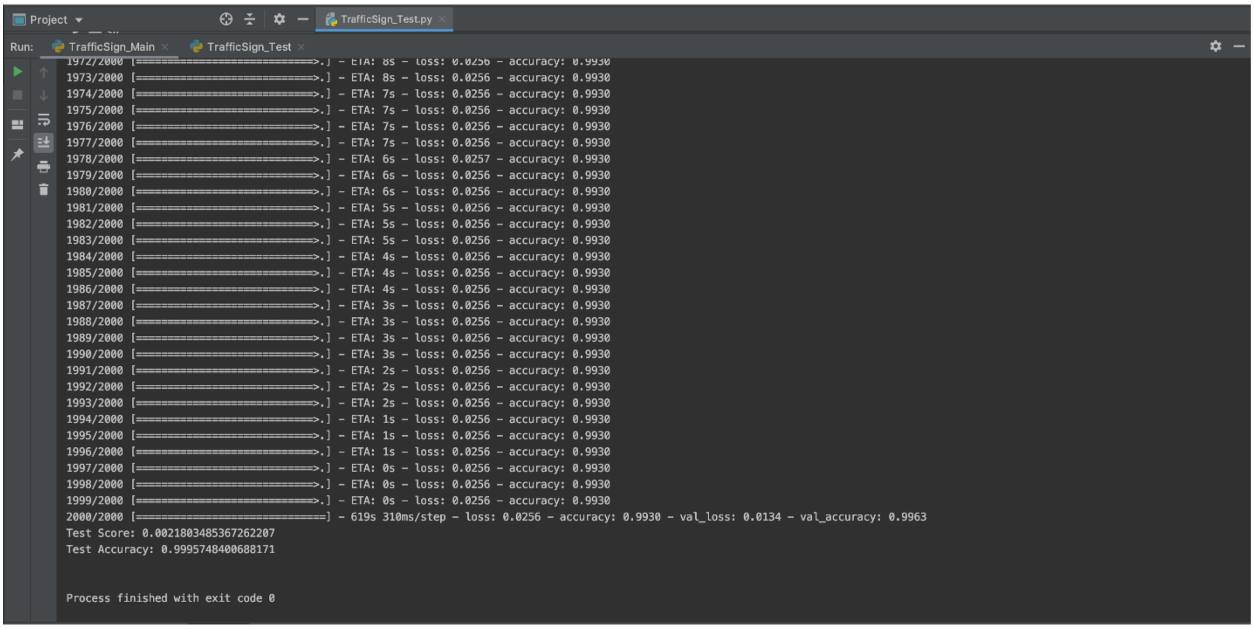


Figure 5.4: Training Results

The training takes a suitable amount of time to finish based on the value of the epoch set. Initially, a value of 30 was set for epoch to observe its performance and choose a suitable number later for better results. At the end of the training sequence, a total validation accuracy of 99.63% was achieved for the chosen validation set. This is a pretty satisfactory result although the real test should be performed with different test and validation datasets to test its real world efficiency. Once the training is complete, 2 performance plots are displayed. The depict the values of loss and accuracy against epoch value. Based on these graphs, a suitable value of epoch is chosen over which the accuracy and loss value are fairly constant i.e. the flat portion of the graph

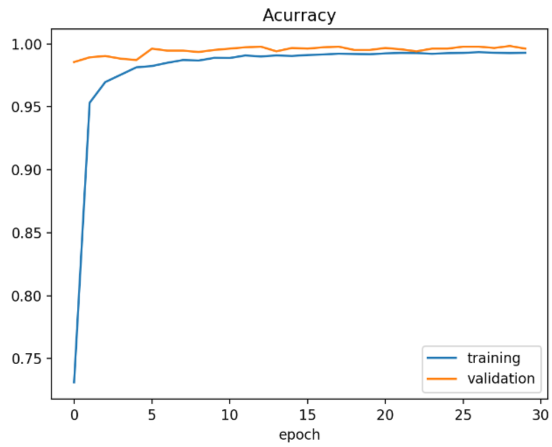


Figure 5.5: Training Accuracy plot

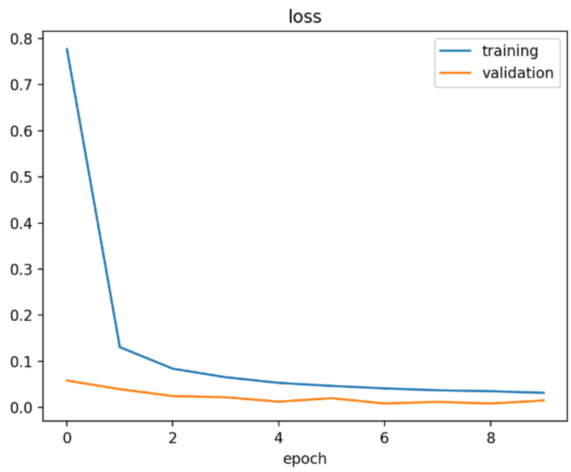


Figure 5.6: Training Loss plot

5.1.2 Image processing

Once the training was completed, a pickle file ".p" was generated by the same code. This file in encrypted and it contains the information of the network. This file is called upon by the running code whenever detection is necessary. A CNN model consists of various layers. Most of these layers are in convolution. They perform complex mathematical convolution functions on the data before they pass on to the successive layers.

Various speed limit signs were manually held in front of the camera. The program running on the raspberry pi hardware was able to detect these signs with a fair amount of accuracy. Owing to the conditions of the country at the time of this project, the authors were unable to obtain real time video of actual street signs to perform tests on. Instead sample pictures were looked up on the internet and they were used instead. Figure 5.7 shows the detection the speed limit board of 30kmph placed in front of the camera. The program can be seen detecting it with 100% accuracy. The program is only considered to be successful if the accuracy is above 90% to avoid wrongful detection.

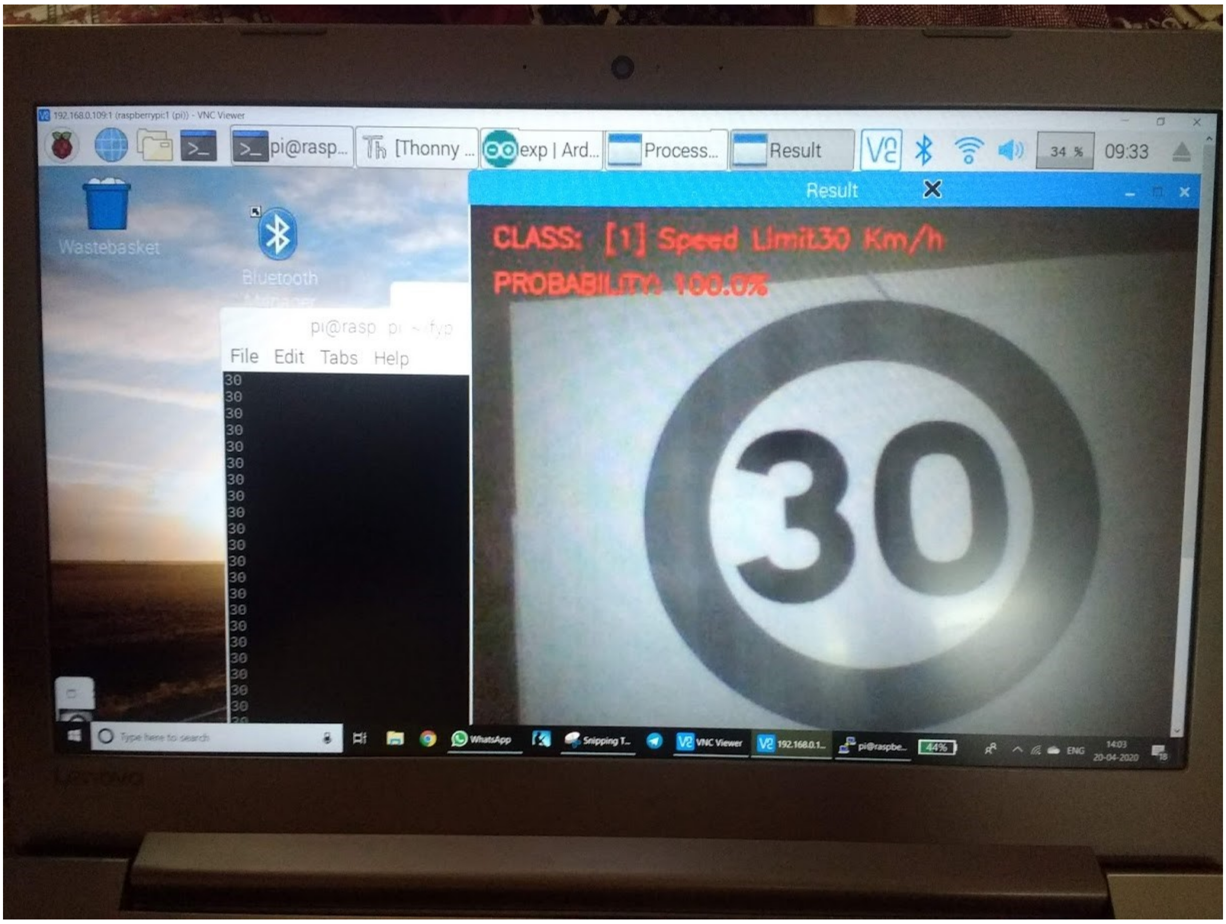


Figure 5.7: Speed limit board number detection

5.2 Hardware Results

A small robotic setup was arranged to perform demo operation. Due to the conditions at hand during the making of this report, the project could not be tested with real sign boards. To perform the demo the components used where Raspberry Pi with camera attached, 2 Arduino Uno boards and a Bluetooth module. The main automobile is represented by the Raspberry Pi and a single Arduino Uno. The second board was used to drive the motor so as to not overwhelm the Raspberry Pi. The setup was as seen in figures 5.8 and 5.9.

Figure 5.8: Bot

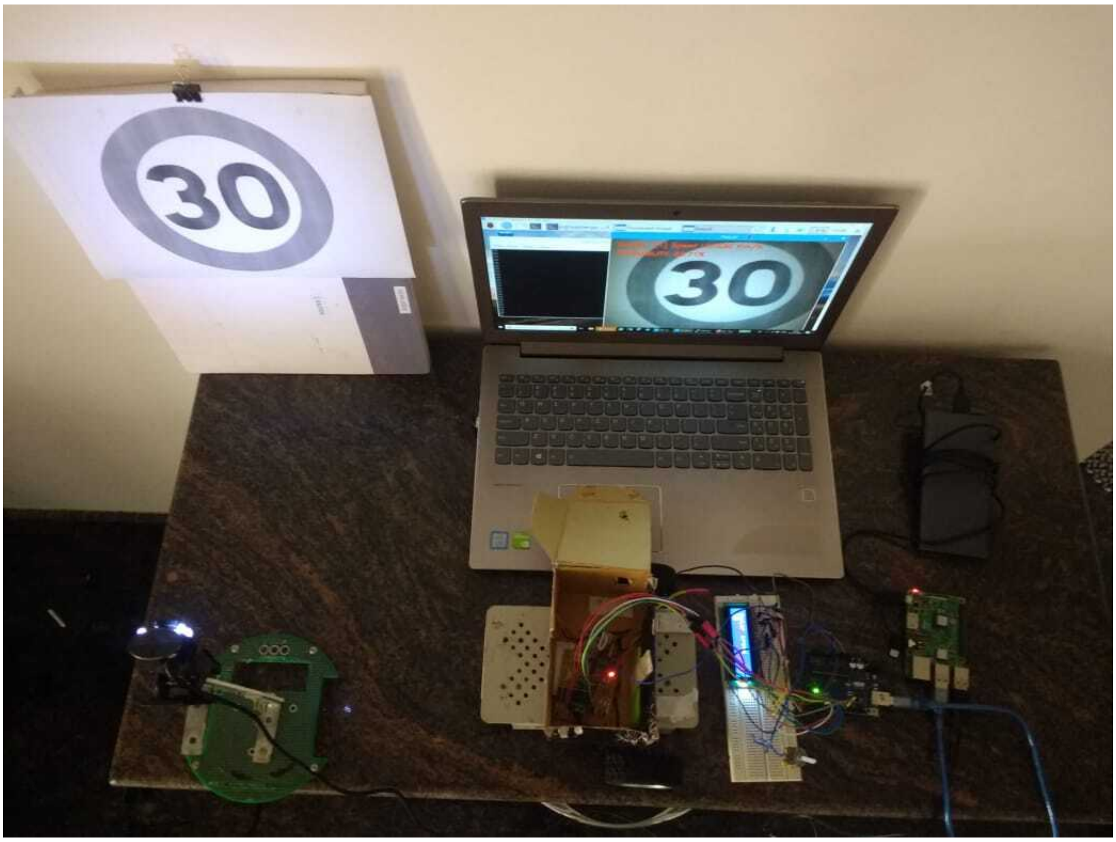


Figure 5.9: Final project circuit connection

To test the project's working various speed limit boards were shown in front of the camera. One of which is shown in the above figure. Based on the detected speed limit board, the speed of the motor was adjusted to that. This can be seen for such an instant in figure 5.10.

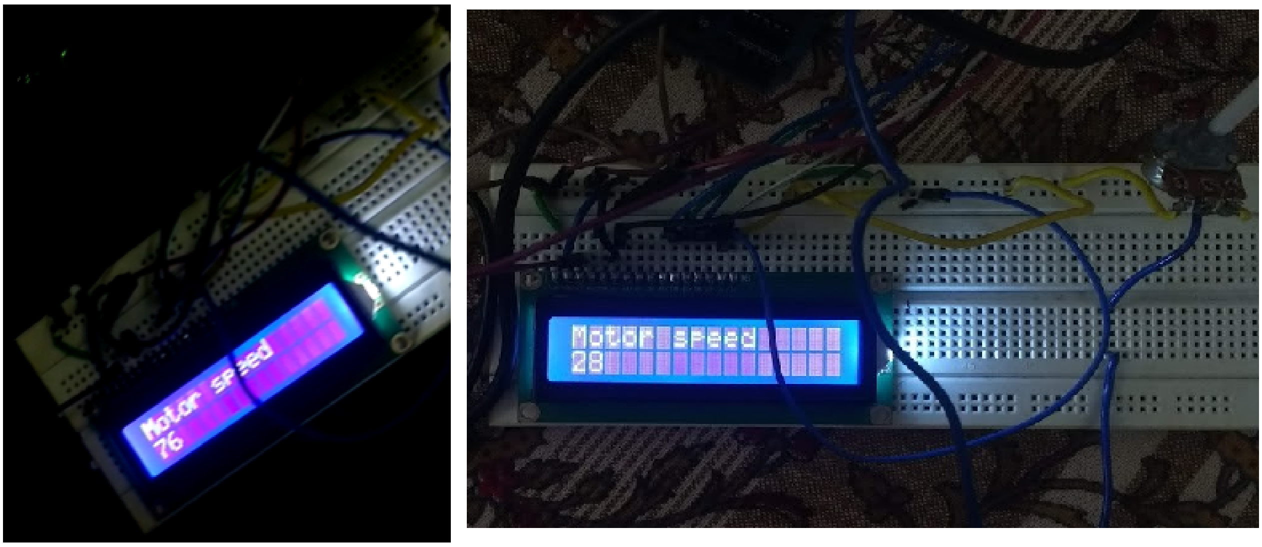


Figure 5.10: Before and After Retardation of motor speed display on LCD

In addition to this, the interpreted speed was transmitted via bluetooth to the other Arduino board connected with a bluetooth module which represents another vehicle.

5.3 Novelty and Comparison of results obtained with previous researches

This project adapts a different approach to road safety as compared to the other road safety centered project mentioned in the literature survey of this report. Most of those projects employ ultrasonic sensors or other distance measuring sensors to detect or predict collisions. After that, the driver is usually alerted which serves as a warning for him/her to bring the vehicle speed under control. This project instead detects speed limit boards placed on either sides of the road. This idea serves as a precautionary measure to make sure the drivers follow the law rather than acting in case of dangerous activities.

Hence, this project could serve as an extension to those collision avoidance systems to improve overall safety of everyone on road by a great extent. The speed control aspect of this project was inspired by all the previous works. They served at reference points for the authors to propose this project.

5.4 Conclusion

A Novel idea of automatically detecting speed limit boards and interpreting them using a camera fixed on the car and limiting the speed without driver intervention was proposed. The proposed idea also included communicating the interpretation to the surrounding similarly equipped cars saving them time from detecting the limits on their own, as well as avoid collisions due to the deceleration of the primary car.

A demo of the proposed idea, was achieved successfully. A bot was fit with a camera connected to Raspberry Pi microcontroller. This device was preloaded with neural network designed to detect the speed boards and interpret it efficiently with high accuracy. This neural network was generated by training with more than 20000 images which served as the basic training set. The interpreted value was used to regulate the speed of motor and also communicate to a neighbouring device using Bluetooth.

The authors hope this project will help inspire such safety features in automobiles. The safety of our roads is of utmost importance and such technological advancements help guide humanity to a better, safer tomorrow.

5.5 Future scope

The neural network can be further trained to recognize other traffic signs. This way the necessary voice alerts can be provided to the driver using the entertainment system in the car. Furthermore, this idea can also be implemented to mechanical cars and provide safety features to that class of users as well. The neural network becomes only more and more accurate with training. The cars set up with this system can also be used as a source for this data periodically.

Bibliography

[1] Vikas Nyamati, Tridha Chaudhuri, Kayalvizhi Jayavel, "Intelligent Collision Avoidance And Safety Warning System For Car Driving",International conference on Intelligent computing control systems,2016.

[2] Manoj Kr. Pal, Pampa Sadhukhan, Nilava Debabhuti, Prolay Sharma, "A Novel Real-Time Collision Avoidance System For On-Road Vehicles", School of Mobile Computing Communication Jadavpur University, India – 700032

[3] Anthony Ian Smith, "Exploring Vehicle-To-Vehicle Communications (V2V)", Penn State Harrisburg, Aldo Morales, Pennsylvania State University.

[4] Noof Al Abdulsalam, Raya Al Hajri, Zahra Al Abri, Zainab Al Lawati, and Mohammed M. Bait Suwailam, "Design And Implementation Of A Vehicle To Vehicle Communication System Using Li-Fi Technology", Department of Electrical and Computer Engineering, Sultan Qaboos University, Muscat, Sultanate of Oman.

[5] "Nikhil Sharma, Vibhor Saini, "Analysis Of Intelligent Transport System With Optical Vehicle-To-Vehicle Communication", Manipal Institute of Technology.

[6] Marc Bechler, Jochen Schiller, Lars Wolf, "In-Car Communication Using Wireless Technology",2018.

[7] Madhvi Verma, Satbir Singh and Baljit Kaur,. "An Overview Of Bluetooth Technology And Its Communication Application",1588— International Journal of Current Engineering and Technology,Vol.5, No.3 (June 2015).

[8] Ramesh Admane, "Automatic Roadside Traffic Sign Detection For Drivers Using Image Processing", International Journal of Science and Research (IJSR),2017.

[9] Mrs. P. Shopa, Mrs. N. Sumitha, Dr. P.S.K Patra, "Traffic Sign Detection And Recognition Using Opencv", IEEE ISBN No.978-1-4799-3834-6/14,2014.

[10] Jian-He Shi, Huei-Yung Lin," A Vision System For Traffic Sign Detection And Recognition", IEEE ISBN 978-1-5090-1412-5/17,2017.

[11] Md. Safaet Hossain,Zakir Hyder, "Traffic Road Sign Detection And Recognition For Automotive Vehicles", International Journal of Computer Applications (0975 – 8887) Volume 120 – No.24, June 2015.

[12] Biswas, R., Fleyeh, H., Mostakim, M. (2014)," Detection And Classification Of Speed Limit Traffic Signs", 2014 World Congress on Computer Applications and Information Systems (WCCAIS).

[13] Emmanuel Gbenga Dada,Abdulkadir Hammidu Alkali, Dauda Elijah Mshelia and Abdul-Azeez Adam Zarma,"Bluetooth Activated Robot Car",2017.

[14] Danda B. Rawat and Gongjun Yan, "Infractures In Vehicular Communications: Status, Challenges And Perspectives" In Advances in Vehicular Ad-Hoc Networks: Developments and Challenges, ISBN 978-1-61520-913- 2, IGI Publications, April 2010.