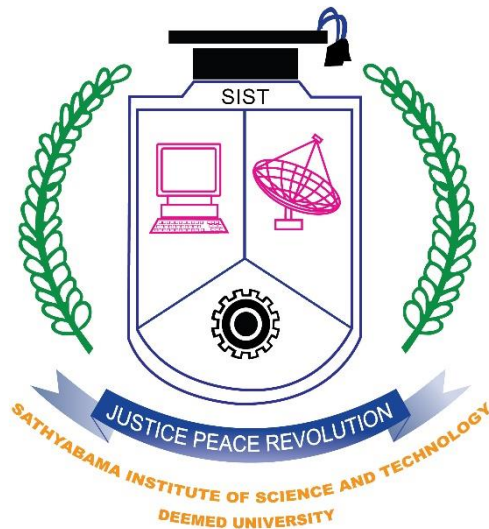


AUTOMATIC SPEED CONTROL USING IMAGE PROCESSING FOR E- VEHICLES

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in
Electrical and Electronics Engineering

by

DHANUSH G (Register No. 39140011)
ROHIT M (Register No. 39140047)



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of DHANUSH G (Reg. No. 39140011) and ROHIT M (Reg. No. 39140047) who carried out the project entitled AUTOMATIC SPEED CONTROL USING IMAGE PROCESSING FOR E-VEHICLES under our supervision from November 2022 to April 2023.

Dr. P. SIVAGAMI
Internal Guide

Dr. V. SIVACHIDAMBARANATHAN

Head of the Department

Submitted for Viva voce Examination held on 26/4/23

Name: D. Rohith

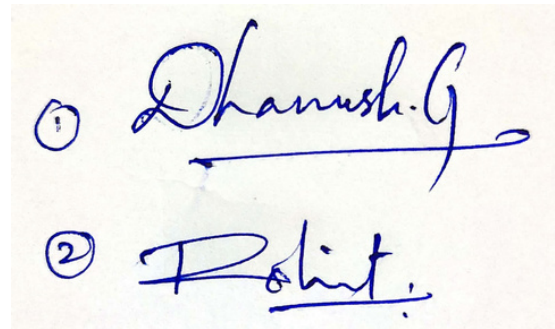
Signature:

Internal Examiner

External Examiner

DECLARATION

We DHANUSH G (Reg. No. 39140011) and ROHIT M (Reg. No. 39140047) hereby declare that the Project Report entitled “AUTOMATIC SPEED CONTROL USING IMAGE PROCESSING FOR E-VEHICLES” done by us under the guidance of Dr. P. SIVAGAMI is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electrical and Electronics Engineering.

The image shows two handwritten signatures in blue ink on a light-colored background. The first signature is labeled with a circled '1' and reads 'Dhanush.G'. The second signature is labeled with a circled '2' and reads 'Rohit.'.

DATE: 26/04/2023

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. Nandhitha., Dean, School of Electrical and Electronics Engineering** and **Dr. V. SIVACHIDAMBARANATHAN, Head of the Department, Dept. of Electrical and Electronics Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.P.SIVAGAMI** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electrical and Electronics Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

In today's world, time is extremely valuable, so everyone strives to complete the tasks as soon as possible by using different modes of transport. In this Typically, vehicles fall into one of two categories: electric or gas-powered, with electric vehicles currently dominating the market. Electric vehicles have grown more appealing not only because of depletion of fossil fuels but also it reduces carbon foot print in the atmosphere. Technology developments have also increased the effectiveness, affordability, and accessibility of electric vehicles. In this era People are considered electric vehicles contribute to the development of more sustainable future. As a result, in order to accomplish the craved responsibility in short period of time, the speed of the vehicles is increased, in the mode of transport used. According to vehicle density predictions. Accident rates are steadily rising as a result of the increase in vehicle speed. Drivers are often not adhering to the laws and regulations set out by the traffic control department in particular areas. However, most users, increase vehicle speed for various reasons in those restricted areas. Over speeding is a big issue, as the number of collisions has risen significantly, to fix these concerns, Automatic Speed Control of E-Vehicle using Image Processing is used. The sales rate of E-Vehicles are gradually Being increased because of the depletion of Fossil fuels in our contemporary scenario. It is targeted on automatic speed control of vehicles by detecting the traffic sign board By Image Processing. In this work, the Traffic Boards are identified by using a Camera. The Current Architecture was created in Flask Framework. If the Speed exceeds the limit the Buzzer alerts the driver which is built in and control the speed automatically. The Speed of the vehicle are Controlled by using CNN algorithm (Convolution Neural Network). Using this CNN (Convolution Neural Network) algorithm we can predict the Traffic Sign Boards by using Image Processing (Deep Learning).

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	x
	LIST OF TABLES	xi
1	INTRODUCTION	
	1.1 GENERAL INTRODUCTION	1
	1.2 BASIC PROGRAMMING SKILLS	3
	1.3 PROBLEM STATEMENT	3
	1.4 OBJECTIVES AND MOTIVATION	4
2	LITERATURE SURVEY	
	2.1 INTRODUCTION	5
	2.2 MAJOR FINDINGS OF LITERATURE SURVEY	5
	2.3 PROBLEM IDENTIFICATION:	9
	2.4 SCOPE OF THE PROJECT	10
	2.5 THE OBJECTIVES OF THE PROPOSED WORK	10
	2.6 METHODOLOGY	11
	2.7 APPLICATIONS	12

	2.8	IEEE STANDARDS	12
	2.9	TRADE OFF	14
	2.10	CONSTRAINTS	15
	2.11	CONCLUSION	16
3		AIM AND SCOPE OF THE PRESENT INVESTIGATION	
	3.1	INTRODUCTION	17
	3.2	EXISTING BLOCK DIAGRAM AND EXPLANATION :	17
	3.3	DISADVANTAGES OF EXISTING SYSTEM	18
	3.4	LAYOUT DIAGRAM AND EXPLANATION	18
	3.5	ADVANTAGES OF PROPOSED SYSTEM	19
	3.6	CONCLUSION	19
4		IMAGE PROCESSING MODULE	
	4.1	INTRODUCTION	20
	4.2	IMAGE PROCESSING MODULE REPRESENTATION	20
	4.2.1	COLOR THRESHOLDING OF DATA	20
	4.2.2	NORMALIZATION OF DATA	21
	4.3	SPEED CONTROL MODULE	21
	4.4	CONCLUSION	22
5		SIMULATION REQUIREMENT SPECIFICATION	

5.1	INTRODUCTION	23
5.1.1	MODEL TESTING	23
5.1.2	PREPROCESSING	25
5.1.3	PREDICTIONS	25
5.2	OUTPUT(GRAPHS)	26
5.3	CNN ALGORITHM	27
5.3.1	COMPARISON BETWEEN CNN AND VGG	28
5.3.2	OUTPUT OF CNN ALGORITHM	29
5.3.3	OUTPUT OF VGG ALGORITHM	29
5.4	SECOND SIMULATION TO DISPLAY SPEED LIMIT	31
5.5	CONCLUSION	33

6

HARDWARE DESCRIPTION

6.1	INTRODUCTION	34
6.2	COMPONENETS DESCRIPTION	34
6.2.1	WEB CAMERA	34
6.2.2	ARDUINO	34
6.2.3	MOTOR DRIVER	34
6.2.4	BLDC MOTOR	35

6.2.5	POWER SUPPLY CONTROLLER	35
6.3	TRAFFIC SIGN RECOGNITION EXPLANATION	35
6.4	DATASET	36
6.5	HARDWARE KIT(BEFORE WORKING)	37
6.6	OUTPUT	38
6.6.1	HARDWARE KIT(AFTER WORKING)	38
6.6.2	SPEED LIMIT DISPLAYED IN SCREEN	38
6.6.3	CONCLUSION	40
6.7	BENIFICATIONS	40
6.8	LIMITATIONS	41
6.9	CONCLUSION	42
6.10	FUTURE SCOPE	43
7	REFERENCES	44
	APPENDIX SOURCE CODE	47

LIST OF FIGURES

FIGURE No.	TITLE	PAGE No.
3.1	Existing Block diagram	17
3.2	Layout Proposed Block diagram	18
4.1	Image processing module representation	20
4.2	Speed control module representation	22
5.2.1	Accuracy graph	26
5.2.2	Loss graph	26
5.3.1	Convolutional neural network	27
5.4.1	Output web page	31
5.4.2	Giving input to it	32
5.4.3	Displaying output	33
6.1	Maximum speed limit	36
6.2	Minimum speed limit	36
6.3	Hardware kit(Before working)	37
6.4	Hardware kit(after working)	38
6.5	Speed limit displayed in screen	39

LIST OF TABLES

TABLE.NO	TITLE	PAGE.NO
5.1	Comparison between CNN and VGG Algorithm	28
5.2	Output of CNN	29
5.3	Output of VGG	29
5.4	Comparison accuracy of CNN and VGG	30

INTRODUCTION

1.1 GENERAL INTRODUCTION:

The majority of traffic accidents are caused by careless driving, disregard for traffic laws, and defiance of traffic sign boards by both individual drivers and members of society at large. because of an inflatedcar Numerous accidents are caused by density and excessive speeding. They used Mathematical analysis of accident reports demonstrates that there are Increased auto density has caused a significant increase in traffic on Indian roadways. Emitting the national road, quite up to the anticipated level performing while operating a vehicle, such as using a mobile device or drinking Driving while disobeying traffic laws and regulations and speeding limitations that put your own and others' safety in peril. Here is evident from the observation that each hour Around 14 persons pass away in traffic accidents. When someone disobeys a trafficsign, they endanger not only their own life but also the lives of pedestrians and other drivers. Boards with speed limits and traffic lights helps to lessen traffic on the roadways, and they are also made to amount of traffic collisions. The use of image processing technology is crucial part in capturing the speed limit sign board. This notebook includes they've devised a system that can assist the driver, greatly passenger security methods for identifying and detecting road signs have also recently adopted by numerous businesses. Previously, the road The drivers manually spotted the signs. However, the Automatic vehicle speed regulation based on signboard detection amount of traffic collisions. The speed limit sign board capture involves the use of image processing technology in a significant way.

We have devised a method that can assist the driver in this journal, greatly enhancing passenger security methods for identifying and detecting road signs have also recently adopted by numerous businesses. Previously, the road The drivers manually spotted the signs. However, theAutomatic Using an image to detect signboards to control vehicle speed Utilizing the web camera module, processing can recognise the indications with ease.

The use of a digital computer to process digital images is known as digital image processing through an algorithm, pictures. As a division or branch of the digital image processing and signal processing both have significant benefits over processing of

analogue images. It permits the application of a considerably wider variety of algorithms to the input data and can prevent issues like the accumulating noise and distortion during processing. Images are defined over because Digital image processing into two dimensions (and possibly more) can be modelled multidimensional systems, for example. development and production primarily impacted by three elements for digital image processing: first, the growth of computers and mathematics, respectively (particularly the development and advancement of the theory of discrete mathematics); third, the demand for numerous environmental applications. The advancement of medical science, industry, agriculture, and the military. Digitization of images the introduction of technologies for medical use in space Basis Space Technology In digital picture processing, alteration occurs by the use of a digital computer, of digital photos It is a field within the signals and systems, but pay close attention to pictures.

DIP concentrates on creating a computer system with image processing capabilities A digital image is used as the system's input, which is processed by the system utilising effective algorithms to produce an image. The most typical illustration is Adobe Photoshop. It is one of the commonly employed a programmed for manipulating digital photos. A camera has taken a picture and transferred it to a digital device. System to ignore all other information and concentrate just on the passing water drop enlarging it while maintaining the same level of image quality. This video teaches you about commonly used techniques and methods for analyzing digital photos for image augmentation and picture restoration and image processing, including blurring, edge detection, sharpening, and others). It also emphasizes the knowledge of how the human eye functions How does a human eye visualize many different things, but how does the brain translate those pictures? Additionally, the lesson goes through key crucial ideas in signals and systems, including sampling, quantization, convolution, and frequency domain. Evaluation, etc.).

1.2 BASIC PROGRAMMING SKILLS :

In addition to this, it necessitates some fundamental knowledge of popular programming languages like MATLAB. The need for a reliable and constant traffic surveillance system developed to manage and regulate the current situation in the rapidly expanding a time of expansion and development. It is urgently necessary to create a system for managing and controlling traffic. One can use this system to In many Indian states, the speed law is enforced.

Traffic flow An essential component of various traffic management systems is estimation. The Traffic flow provides a fixed-time snapshot of the traffic situation and aids in coping and have authority over the others. It is utilized particularly when there is traffic a traffic jam is caused by the vehicle's speed. The context provided by an intelligent system for monitoring vehicle traffic allows for the extraction of important data from traffic statistics. a precise and powerful. The brain of such a system based on vision-based tracking is a lot of time and effort has been put into vehicle detection utilising an image processing technique, and speed estimation a swift.

One of these technologies that can replace them is the detection camera system (SDCS). conventional radar-based systems and offers an effective way to detect and follow the things. The magnetic sensors gauge a vehicle's speed a vehicle using magnetic radiation as the basis. Their tracking method didn't not employ cameras and forecast its motion with reasonable certainty. the extent The high cost and limited availability of radar systems have a detrimental impact on their use precision of results. This inspired us to create a different method. This might provide a solution that is accurate and affordable.

Estimating vehicle flow is crucial for effective traffic management system. It is crucial to tracking systems and automatic video both for observation and to prevent collision This essay offers a strategy for Calculate the average speed of cars on the highway and in the city. The suggested approach can be successfully used to control the excessive.

1.3 PROBLEM STATEMENT:

The problem statement of Automatic Speed Control is to develop a computer vision system that can accurately detect and classify traffic signs from digital images or video

streams in real-time. This system is designed to assist drivers by providing real-time information about traffic signs, such as speed limits, stop signs, yield signs, and other road signs, to improve road safety and prevent accidents. The main challenges in traffic sign recognition include handling variations in lighting, weather conditions, and camera viewpoints, as well as identifying similar-looking signs that may have different meanings based on their context.

1.4 OBJECTIVES AND MOTIVATION

The objective of traffic sign recognition is to improve the safety and efficiency of driving by providing drivers with real-time information about traffic conditions, speed limits, and other important traffic regulations. This requires algorithms and systems that can automatically detect and interpret traffic signs in real-world scenes captured by cameras or other sensors mounted on vehicles or in the road side environment. Additionally, traffic sign recognition can be used to support the development of smart automatic vehicles, which rely on accurate perception of the environment to make decisions and navigate safely.

The motivation behind traffic sign recognition is to increase safety on the roads and improve the driving experience for motorists at efficient cost. Accurate and efficient recognition of traffic signs make the drivers aware of road conditions, speed limits, and other important information that can help prevent accidents and improve traffic flow. By providing drivers with real-time information about the road side traffic flow, traffic sign recognition technology can also reduce driver stress and increase confidence, making driving a more pleasant and enjoyable experience. Additionally, technical innovation towards automatic vehicles is a key motivation for traffic sign recognition, as these vehicles require reliable and accurate perception of the environment to operate safely and efficiently.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

The focus of recent image processing research has been on the recognition of traffic signs. The challenge of sign identification can be solved using a variety of techniques, according to a thorough review of the literature. The traffic sign board can be monitored and accidents can be avoided using the image processing technology.

2.2 MAJOR FINDINGS OF LITERATURE SURVEY

A. Seto, Youji, Inoue, Hideaki, Kobayashi, Yosuke (2016) An adaptive automobile speed control system includes a vehicle speed sensor that distinguishes the host vehicle's vehicle speed that is accountable to adaptive vehicle speed control, which includes the vehicle speed control and the vehicle speed control system, and an inter-vehicle distance sensor that detects the inter-vehicle distance between the host vehicle and the vehicle speed control. A man-machine interface is provided for establishing a certain vehicle speed. A desired inter-vehicle distance arithmetic processing unit estimating the desired inter-vehicle distance based on at least the speed of the host vehicle, and an associated control section inferring the desired host vehicle speed based on the speed of the host vehicle are also provided.

B. Zhigang, Fan, Wu, Wencheng (2019) A computer-implemented method for deducing the speed of a motor vehicle is disclosed in the vehicle speed detection system in this paper. The technique may implicate the reception of a plurality of images of a motor vehicle on a lane, Determining, for each of at least two of the photographs, the point of contact where the same tire of the vehicle contacts the road surface, based in part on one or more of the vehicle's identified attributes in one or more of the plurality of images; and using the points of contact and time interval separation to calculate a speed at which the vehicle is travelling on the road

C. Michalopoulos, P.G. Recent (2018) worldwide innovations are briefly discussed about the use of vehicle detection for large-scale wireless data collection and implementation of advanced traffic control and management systems such as vehicle guidance/navigation. Early work on Autoscope is examined, leading to its current advantages over other emerging devices, and recent advancements are examined, such as the development of preproduction line prototypes, field testing, and comprehensive field validation and verification plans. Two major demonstration projects recently launched in Minneapolis include the above plans. The Autoscope will be used for incident detection over a portion of Interstate 394 in the first project. The second project implicates execution of the machine vision system at a signalized intersection.

D. Vaughn (2018) David In this research paper researcher explain about GPS connectivity with vehicle which can track your vehicle speed and location this method called GPS map speed matching method. This GPS navigation device needs a database processing center, receiver, a motor computer, a video display, a speed sensor and a header sensor. This GPS navigation connect with odometer so speed of vehicle will be more accurate. For processing facility for the database maybe local or remote. Vehicle longitude, latitude, direction and speed will be easily obtained by GPS device and until device will be there is vehicle. Device also able to control vehicle in case of over speeding whenever driver speed up their vehicle over the limit that time device will automatically get all data from vehicle like odometer reading (vehicle speed), vehicle location, maximum posted speed after calculating all data device will command vehicle to slow down. That's how we can control over speeding and make drive more safe.

E. Lin, Huei-Yung, Li, Chang (2015) This is the Camera based speed detection method also called Image based method. In this method speed cameras are used for detecting speed of vehicle. This is the best and cheap way to detect vehicle speed than usual laser or radar based detection Which are more expensive comparatively speed detection camera system. The system uses cameras to capture an image of vehicle with sensing the vehicle motion for calculation of vehicle speed. This operation is done by using relative motion of the car to the camera. During the camera exposure time, the blurriness of motion in the dynamic region of image displays calculation of speed of the moving car. The results show 5% speed estimation of the real speed.

F. Rad, Arash Gholami (2015) Usually video and image processing technology (speed cameras) are used for traffic surveillance, monitoring of traffic conditions and for analysis in many cities and urban areas. In this paper a different methodology is presented for the hypothesis of vehicle speeds. The photographs and videos are taken from the roads mounted cameras which have geometrical equation based calibrations. The results shows a difference error of 7 km/hr on running with different resolutions.

G. Pornpanomchai, Chomtip, (2017) The paper is about developing a camera based speed detection system which provides reference position points and detection of speed is done by calculating the distance between the moving object and reference points.

H. Rahim, H.M (2018) In this paper, researchers have applied smart surveillance system by using PC workstation. The camera is set to 30ms to 40ms of resolution and it is limited to the camera frame rate. The factors affecting the maximum velocity are timing resolution and displacement resolution of pixels. The frame differencing technique estimates the speed of vehicle.

I. Kerekes, (2016) The radar technology is an old technology, and it is useable for traffic monitoring purposes, this technology was mainly developed for military purposes, but it can be used for civilian purposes nevertheless this technology is highly costly, and less accurate than the image processing technology.

J.Ki and Baik (2014) Proposed a method based on double loop detector for measurement of vehicle speed. However, loop based method incurs error due to spacing between double loops, scanning time, irregular trajectories of vehicles and presence of multiple vehicles. For minimizing these errors in calculation of speed, author provided error filtering algorithm.

L.Wei (2017) Presented a vehicle detection method based on magnetic effect using iterative wavelet algorithm. It uses wireless magneto-resistive sensor based vehicle detection. They estimated the speed of a vehicle using clock synchronization method that minimizes the error.

M.Mei and Li (2018) Proposed a method for measuring speed of a railway vehicle. They used simplified inverse model to extract motion features by two sensors mounted on a bogie. The crosscorrelation calculates the time delay in predicted motion between two

wheel sets. This technique replaces the conventional wheel based estimation to overcome various difficulties incurred in wheel slip/slide conditions.

N.Ibrahim (2019) Presented a method to find the speed of a vehicle from single camera. This method consists of four phases. In the first phase, adaptive background subtraction algorithm is used to detect moving objects with three frame differencing. The second phase is consists of segmentation, labeling, etc. The third phases measures the speed of a vehicle based on number of frames passed by the vehicle in a particular scene. The fourth phase captures the picture of the vehicles that violates the speed limit. However, this method does not provide comparison with any of the existing method.

O.Taghvaeeyan and Rajamani (2013) Developed a magnetic sensor based method for vehicle classification, counting and speed estimation. This uses wireless anisotropic • Subhash Chand Agrawal is currently pursuing PhD in computer science and Engineering from GLA University, Mathura, India. Email: subhash.agrawal@gla.ac.in • Rajesh Kumar Tripathi is currently pursuing PhD in computer science and Engineering from GLA University, Mathura, India. magnetic devices which are placed next to the roadway and estimates traffic in adjacent lanes. Longitudinally spaced sensors are used to measure the speed based on crosscorrelation. Vehicle classification is done through magnetic length and vertical magnetic height of the vehicle.

P.KV (2016) Proposed a method to estimate the speed without camera calibration. They segment the vehicle using frame subtraction and masking technique. Then, speed is calculated using timestamps of frames and segmented object.

Q. Yang and Lei (2013) Developed a method for vehicle classification and speed estimation using anisotropic magnetoresistive sensor. This method can be utilized in large cities where low speed congested traffic is there. Fixed threshold state machine algorithm is used to detect vehicle in a single lane. Sensor computes five features: direction of vehicle in which it is travelling, vehicle signals, average energy, energy, ratio of positive and negative energy in x axis and ratio of positive and negative energy in y axis. Thus detected vehicle is classified as motorcycle, car, bus, etc. based on classification tree module.

R.Chanawangsa (2019) Presented a surveillance system using calibrated stereo cameras. Vehicle detection and tracking is performed in 2D space, and then depth and

appearance features are incorporated into a tracking module using 2D-3D approach. 3D coordinates of a detected vehicle help to estimate the speed of a vehicle.

S.Momin (2021) Proposed a method for detection a vehicle based on Haar classifier and Adaboost. Method uses optical flow to find the changes in the intensity values across the frames. These apparent velocity components are then subjected to image processing techniques to obtain centroid of a vehicle. Afterward, the image coordinates of the centroid are mapped to world space. Using this information the velocity of the vehicle is estimated. Searching of vehicle is done through color, speed, direction and date and time.

T. Zhu and Yu (2010) Proposed a method to deal with noise disturbances for on-street parking system. For this, they utilized cross-correlation technique in wireless magnetic sensor network. The speed of a vehicle is calculated using cross correlation between signals of two sensors, placed along the road at certain spacing. They achieved accuracy of 92%.

U.Gerat (2019) Presented a method based on Gaussian mixture models, DBSCAN, Kalman filters and optical flow. The method is tested on different vehicles, driving positions etc.

From the above Literature Surveys, we come to conclusion that These projects are being implemented to Control the speed of the ordinary Vehicles using different techniques and Algorithms But this work is completely worked out for E-VEHICLES and using CNN Algorithm.

2.3 PROBLEM IDENTIFICATION:

The primary goal of traffic sign recognition is to enhance the safety of drivers, pedestrians, and other road users by providing real-time information about traffic signs such as speed limit, no entry, yield, stop, and many others. Traffic sign recognition is the process of detecting and identifying traffic signs from images or videos captured by cameras mounted on vehicles or traffic infrastructure. The issues that arise in traffic sign recognition are:

- When there are changes in lightning condition such as shadows, glare and reflection, Traffic sign recognition and traffic sign appearance becomes difficult to identify. An in-depth analysis is required to determine the sizing limits of images to acquire the accurate design.
- The design of the recognition, systems must be accurate, that they should be able to detect and recognize the traffic sign. Correctly and consistently under various conditions and scenarios light and fast-moving vehicles.

2.4 SCOPE OF THE PROJECT

The design can be used in any city around the world. Traffic sign detection and recognition plays an important role in expert systems, such as traffic assistance driving systems and automatic driving systems. It instantly assists drivers or automatic driving systems in detecting and recognizing traffic signs effectively. Scope of a project on traffic sign recognition would typically involve developing a system or algorithm that can accurately detect and classify different types of traffic signs from images or video streams captured by cameras mounted on vehicles or at roadside locations.

2.5 THE OBJECTIVES OF THE PROPOSED WORK:

The objective of traffic sign recognition is typically to develop an algorithm or system that can accurately detect and recognize traffic signs in real-time using cameras or other sensors. The primary purpose of traffic sign recognition is to improve road safety by alerting drivers to important road information, such as speed limits, stop signs, and other traffic regulations. Accurate and reliable traffic sign recognition can also support the development of autonomous vehicles, which rely on visual perception to navigate roads and make decisions. the objective of a proposed work for traffic sign recognition is likely to develop a system that can accurately and efficiently detect and recognize traffic signs, which can improve road safety and support the development of autonomous vehicles.

2.6 METHODOLOGY

In terms of automated driving assistance and security monitoring. Recognizing the direction of pedestrian movement is essential for systems. the streets roads and public gatherings where thousands of people often gather. Generally speaking, moving items that are most crucial and susceptible are pedestrians. analysis of people's movements at zebra crossings, in commercial areas, and Increasing safety at events like demonstrations is essential and enabling autonomous vehicles to operate in actual driving situations. The goal of This paper will employ deep learning methods like Convolution Neural Networks (CNN) to find people walking in a certain direction We provide a new input method thatmakes use of pedestrian data. detecting methods that generate a total ofremoved frames as an input to the hypothetical CNN. Additionally, we created a brand-new dataset for this for this reason:

There are numerous attributes in the incoming dataset in several different ways, such as accident-specific characteristics, driver-specific among other things, characteristics, situation-specific traits. This information was employed to complete the majority of the job. Data construction is often referred to as data preparation. Data will initially be cleaned of noise, missing values, and irregularities. NULL values are usedin place of omitted values The data will be selected and modified in accordance with the requirements. It could appear as a number or a symbol. The info in Although the dataset would initially be in nominal form,it can be changed to as operations are applied to numerical data. Moreover, each In order for attribute data to be used for further research,it is disproved. Accident circumstances like the way the collision occurred,the light, the weather, the condition of the road, the speed restrictions, and a drunk driver were selected as important qualities in this report. Thesecharacteristics also comprise the the values shown here.

Type of Collision: Not with a Motor vehicle, front-to-side (samedirection), front-to-side Angle-Front-to-Side (Right Angle), (Opposite Direction), (Includes) Angle-Front-to-Side, Broadside (Angle-DirectionNot Specified),

2.7 APPLICATIONS

- Automatic Speed Limit Detection Using Camera Module.
- Commercial Vehicle Operations.
- Automatic Speed Control.
- Advanced Transportation System for Safety.

2.8 IEEE STANDARDS

IEEE Std 1850-2018: "Standard for Performance Metrics for Binary Classification Problems" - This standard defines performance metrics to evaluate binary classification algorithms CNNs.

- Terminology: IEEE Std 1850 is a standard document that provides a set of definitions and terms used in the field of electronic design automation (EDA). This standard was developed by the IEEE Standards Association and is intended to facilitate clear and consistent communication among EDA professionals. By providing a common set of definitions and terms, IEEE Std 1850 Terminology helps to reduce confusion and misunderstandings that can arise when different professionals use different terminology to describe the same concepts. This standard is an important resource for anyone working in the field of EDA, whether as a designer, a verification engineer, or in any other role.

IEEE Std 1855-2016: "Standard for Learning Data Format" - This standard specifies a standard format for exchanging data between different machine learning algorithms, CNNs.

- Terminology: Std 1855 is a standard document that provides a set of definitions and terms used in the field of electronic design automation (EDA) for intellectual property (IP) cores. This standard was also developed by the IEEE Standards Association and is intended to facilitate clear and consistent communication among IP core developers, integrators, and users. By providing a common set of definitions and terms, IEEE Std 1855 Terminology helps to reduce confusion and

misunderstandings that can arise when different professionals use different terminology to describe the same concepts related to IP cores. This standard is an important resource for anyone working in the field of IP cores, whether as a developer, an integrator, or a user.

IEEE Std 1599-2008: "Standard for Visualizing Multidimensional Arrays" - This standard defines a standard format for visualizing multidimensional arrays, including those used in CNNs.

CNN Implemented in this program to train the model by having the real time image.

CNN contains the following layers:

- Input layer
- Convo layer (Convo + ReLU)
- Fully connected (FC) layer
- Soft ax/logistic layer
- Output layer
- Pooling layer

IEEE Std 1873-2015: "Standard for Network-Based Education and Training Systems" - This standard defines requirements and guidelines for network-based education and training systems that may include CNN-based algorithms.

- Terminology: IEEE Std 1873 standard document that provides a set of definitions and terms used in the field of network function virtualization (NFV). This standard was developed by the IEEE Standards Association and is intended to facilitate clear and consistent communication among NFV professionals. IEEE Std 1873 Terminology helps to reduce confusion and misunderstandings that can arise when different professionals use different terminology to describe the same concepts

related to NFV. This standard is an important resource for anyone working in the field of NFV, whether as a developer, a network operator, or in any other role.

2.9 TRADE-OFF:

Traffic sign recognition systems have several tradeoffs that can impact their effectiveness in real-world scenarios. Here are some of the key tradeoffs to consider:

- **Accuracy vs. speed:** The accuracy of a traffic sign recognition system can be improved by using more sophisticated machine learning models, but this can also increase the processing time required to recognize each sign. In some situations, such as in autonomous vehicles, speed is critical, and so the system may need to sacrifice accuracy to make faster decisions.
- **Robustness vs. flexibility:** A traffic sign recognition system that is designed to recognize a specific set of signs will generally be more accurate than one that must handle a wide range of signs. However, a more flexible system that can recognize a variety of signs may be more practical in real-world scenarios, where new signs may be added or existing signs may be modified.
- **Lighting conditions:** Traffic sign recognition systems can struggle to accurately detect signs in low-light conditions or when the sun is shining directly on the sign. This can limit their effectiveness in certain situations, such as in tunnels or during sunrise and sunset.
- **Data quality:** The accuracy of a traffic sign recognition system depends on the quality of the data used to train it. If the system is trained on poor quality or incomplete data, it may struggle to recognize signs accurately in the real world.
- **Cost:** Implementing a traffic sign recognition system can be expensive, particularly

if it requires specialized hardware or sensors. This cost may be prohibitive in certain applications, especially those with tight budgets or low profit margins.

Overall, the tradeoffs of traffic sign recognition systems will depend on the specific application and requirements of the system.

2.10 CONSTRAINTS:

There are several constraints of traffic sign prediction that need to be considered:

- **Image quality:** The accuracy of traffic sign prediction is highly dependent on the quality of the input image. Poor lighting, image blur, and low resolution can all reduce the accuracy of the prediction.
- **Variability in sign design:** Traffic signs can have a wide range of designs, even for the same type of sign. This variability can make it challenging for an algorithm to recognize all the different variations.
- **Environmental factors:** The presence of other objects in the image, such as trees, buildings, or other vehicles, can make it difficult for the algorithm to isolate the traffic sign.
- **Occlusion:** Traffic signs can be partially or fully occluded by other objects, such as other vehicles or road signs, which can make it challenging for an algorithm to recognize them.
- **Real-time processing requirements:** In some applications, such as autonomous driving, traffic sign prediction needs to be performed in real-time, which places constraints on the computational resources available for the algorithm.
- **Limited training data:** The availability of large amounts of training data can be a limiting factor in the accuracy of traffic sign prediction algorithms, particularly for rare or uncommon signs.
- **Safety criticality:** The failure of a traffic sign prediction algorithm can have serious safety implications, which means that the algorithm needs to be highly reliable and robust.

2.11 CONCLUSION:

- IEEE (Institute of Electrical and Electronics Engineers) standards are an important set of guidelines and specifications that ensure consistency, reliability, interoperability, and safety in the design, development, and deployment of various electrical, electronic, and software technologies.
- IEEE standards cover a wide range of domains and applications, including communications, computing, power systems, industrial automation, medical devices, transportation, and more. They are developed through a rigorous consensus-based process that involves input from various stakeholders, including industry experts, academics, regulators, and users.

CHAPTER – 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 INTRODUCTION:

The proposed system for traffic sign recognition is an intelligent system that uses computer vision and machine learning techniques to automatically detect and recognize traffic signs from camera images. The system aims to improve road safety by providing drivers with real-time information about road signs, such as speed limits, stop signs, and warning signs.

3.2 EXISTING BLOCK DIAGRAM AND EXPLANATION:

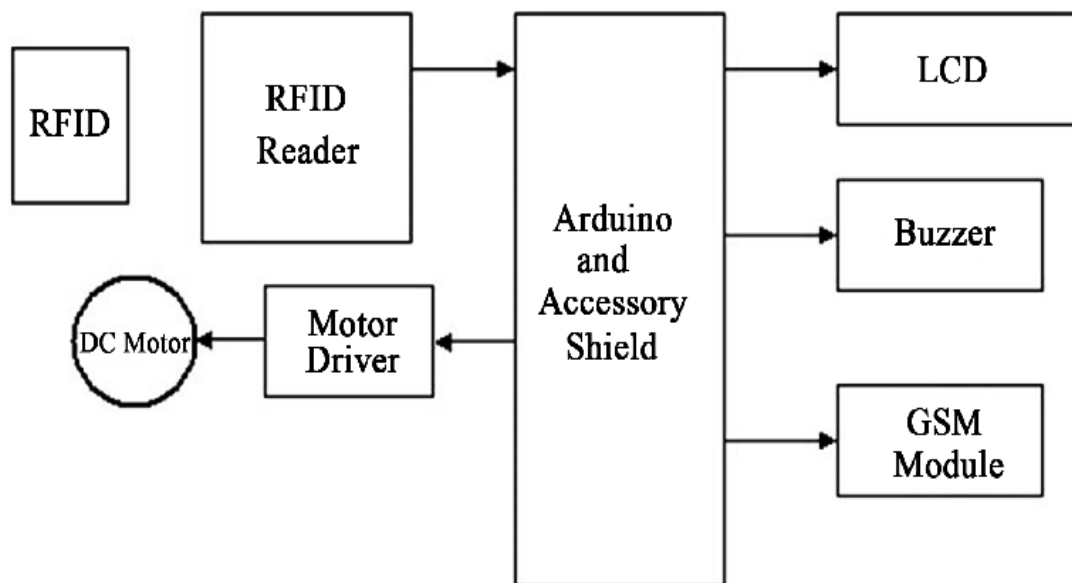


Fig 3.1 : Existing block diagram

This explains how the embedded technology and radio frequency (RF) used by the automatic speed control system for electric vehicles (E-Vehicles) work to prevent accidents. From the mode of invention until today, many accidents occur as a result of driver negligence. This mechanism aids in controlling the E-'s speed and serving as a driver alert. The current E-mail system will benefit more from this method vehicle architecture. Two different types of scenarios have been thought of for

the regulating the vehicle's speed. The first step is to warn the motorist and manage the speed upon passing another signboard that reads "Slow Speed". One is to manage the vehicle's speed when an unexpected barrier appears with the use of an ultrasonic sensor in front of the car.

3.3 DISADVANTAGES OF EXISTING SYSTEM:

- Interference.
- High Cost.
- Some materials may create Signal problem
- Overheating reading(fail to read).

3.4 LAYOUT DIAGRAM AND EXPLANATION :

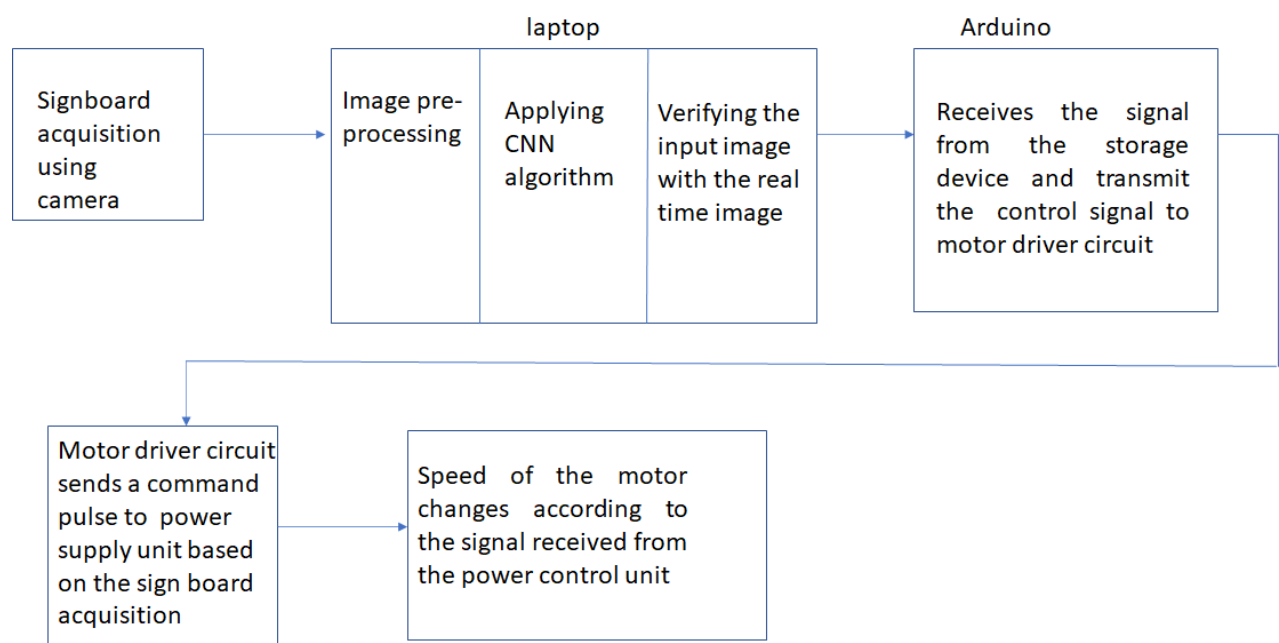


FIG:3.2 FLOWCHART REPRESENTATION OF PROPOSED BLOCK DIAGRAM

Step by step explanation of proposed block diagram:

- Signboard will be acquired by using the camera and transmit it to the laptop.
- The image will undergo pre-processing on the laptop, including noise reduction, normalization, and application of the CNN algorithm to train the model and

compare the acquired image to the real-time image and it send signals to the Arduino.

- Arduino board will transmit the control signal to the motor driver circuit after receiving the signal from the storage device.
- Based on the sign board acquisition, the motor driver circuit transmits a command pulse to the power supply unit.
- Depending on the signal from the power control unit, the motor's speed changes according to the signboard.

3.5 ADVANTAGES

- Easy to use – The deep learning method makes image detection very simple.
- Accuracy – The automation system is highly reliable, and image detection performance is very good.
- Quick response – After detecting a picture, the vehicle's speed is quickly monitored.

3.6 CONCLUSION

Based on the proposed diagram of Automatic speed control using image processing, the conclusion is that a system can be designed to recognize traffic signs by following the steps of input image acquisition, image preprocessing, feature extraction, feature selection, classification, post-processing, and output. The system uses machine learning algorithms to classify the traffic sign image into a specific category and can be used to control an automated system such as a self-driving car or a traffic control system. With proper implementation and training of the system, it has the potential to improve road safety by accurately identifying and responding to traffic signs.

CHAPTER 4

IMAGE PROCESSING MODULE

4.1 INTRODUCTION

Regardless of the weather, this image processing module aims to identify the posted speed limit on traffic signs. presents the speed limit recognition system's proposed model for sign boards. The sign board image will be captured by this technology, and it will also determine the pace. After receiving an input image, the unit converts it to grayscale before performing normalization and noise removal. The final step is to feed each preprocessed image into a CNN unit so it can perform classification.

4.2 Image Processing Module Representation

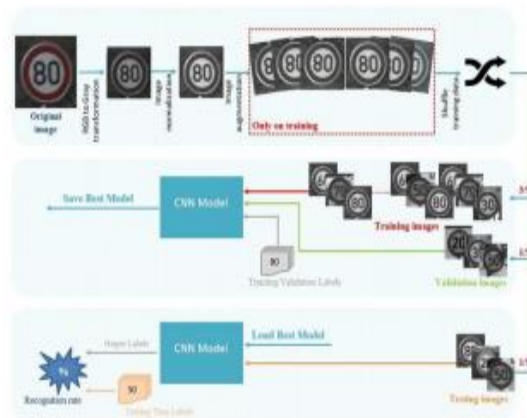


Fig. 4.1 Image Processing Module Representation

4.2.1 Color Thresholding of Data :

The RGB color space is the most easily understood. Each element is red, green, and blue in color. As a result, the expression for the color threshold is as follows:

$$g(x,y)=k1 \begin{cases} Ra \leq fr(x,y) \leq Gb \\ Ga \leq fg(x,y) \leq Gb \\ Ba \leq fb(x,y) \leq Gb \end{cases} \leq Gb$$

$g(x,y)=k2$ in any other case

where the functions that determine the red, green, and blue levels of each point in the image are, respectively, $fr(x,y)$, $fg(x,y)$ and $fb(x,y)$. More effort is needed to organise the deep neural network to be told meaningful possibilities from speed-limit images. The coaching photos are first tended to be enlarged, followed by normalisation of the enlarged images, and last, a median filter is tended to be applied.

4.2.2 Normalization of Data:

In order to reduce poor illumination differences identified in the database, we normalise the grey scale image in this stage. Let $Im(i,j)$ represent a pixel's grayscale value (i,j) Me and Std stand for the estimated means and standard deviations of Im and $Norm(i,j)$, respectively. Denotes the normalized grayscale value of pixel (i,j) . The image which is normalized is defined.

$$Norm(i,j) = (Im - Me) \div K1$$

Two constants, $K1$ and $K2$, have respective settings ranging from 50 to 100. Figure 1.5 displays the images produced by 1. Atlast, which uses a median filter to create an image of a speed restriction sign board that is more emphatic than the original image.

4.3 Speed Controller Module

The speed controller module uses the information obtained from the image processing module as a speed limit reference in order to reduce the speed of the electronic vehicle and input it into the speed controller unit. The BLDC motor system under study Only two fundamental components make up a brushless DC motor: the rotor and the mechanical component. While the mechanical device is the fixed half and features mechanical device windings, the rotor, which is the revolving portion, has rotor magnets.

In BLDC, the electromagnets are moved to the mechanical device by permanent magnets attached inside the rotor. stator The speed controller phase reduces the vehicle's speed by using the output from the image processing module as an input. 4.2

In this module, the three-phase inverter is controlled by a speed controller that uses speed

reference data from CNN classification to transmit the appropriate power supply to a BLDC motor, allowing it to classification to transmit the appropriate power supply to a BLDC motor, allowing it to operate within the confines of the road's speed restriction regulations.

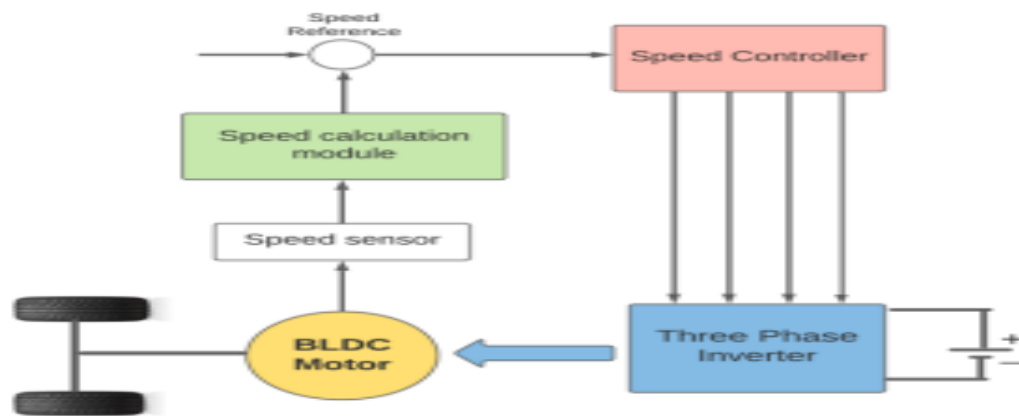


Fig 4.2 Speed control module representation

4.4 CONCLUSION:

The conclusion of image processing is that it is a powerful tool for enhancing, analyzing, and manipulating digital images to extract useful information. Image processing techniques can be applied to a wide range of fields, including medical imaging, remote sensing, robotics, and computer vision. With advances in technology and algorithms, image processing has become increasingly important and is widely used in various industries to improve efficiency, accuracy, and automation.

CHAPTER 5

5. SIMULATION REQUIREMENT SPECIFICATION

5.1 Introduction

Traffic sign recognition simulation is a process of testing and evaluating the performance of a traffic sign recognition system in a virtual environment. Simulation allows researchers and developers to assess the accuracy, robustness, and efficiency of the system in various scenarios without the need for physical experiments on real roads. The simulation typically involves creating a virtual scene that includes different traffic signs and testing the system's ability to detect and recognize them accurately. The virtual scene can be designed to simulate various driving conditions, lighting conditions, weather conditions, and other environmental factors that affect the recognition process.

5.1.1 MODEL TESTING

- ▶ Before moving on to detection or classification, the most important part is the availability of a generalized dataset. A prediction model is trained using this dataset and predictions are done for test dataset.
- ▶ It consists of large number of images
- ▶ The traffic signs are of different variety, background, and color variation which in turn will help the model to perform accurately.

- ▶ # Classes of traffic signs

classes = { 0:'Speed limit (20km/h)', #

- ▶ Classes of traffic signs

Classes of traffic signs

classes = { 0:'Speed limit (20km/h)

1:'Speed limit (30km/h)',
2:'Speed limit (50km/h)',
3:'Speed limit (60km/h)',
4:'Speed limit (70km/h)',
5:'Speed limit (80km/h)',
6:'End of speed limit

7:'Speed limit (100km/h)',
8:'Speed limit (120km/h)',
9:'No passing',
10:'No passing veh over
11:'Right-of-way at
12:'Priority road',
13:'Yield',
14:'Stop
15:'No vehicles',
16:'Vehicle > 3.5 tons
17:'No entry',
18:'General caution',
19:'Dangerous curve left',
20:'Dangerous curve right',
21:'Double curve',
22:'Bumpy road',
23:'Slippery road',
24:'Road narrows on the
25:'Road work',
26:'Traffic signals',
27:'Pedestrians',
28:'Children crossing',
29:'Bicycles crossing',
30:'Beware of ice/snow',
31:'Wild animals
32:'End speed + passing
33:'Turn right ahead',
34:'Turn left ahead',
35:'Ahead only',
36:'Go straight or right',
37:'Go straight or left',
38:'Keep right',
39:'Keep left',
40:'Roundabout,
41:'End of no passing',
42:'End no passing

- **These are the above traffic signs that we used as the dataset to train the model**

5.1.2 PREPROCESSING:

- ▶ *The dataset we have is fairly unbalanced which might make model more bias towards particular class plus deep learning models may have million of parameters which require vast amount data to tune it. `image = cv2.imread(path + '/' + img)`
`#reads the images as arrays`*
- ▶ *`image_fromarray = Image.fromarray(image, 'RGB')` #transforms the arrays to images*
- ▶ *`resize_image = image_fromarray.resize((32, 32))` #resizes the images to 32x32*
- ▶ *`data.append(np.array(resize_image))` #adds the arrays to our 'data' list*
- ▶ *`labels.append(i)` #adds the labels to our 'labels' list*

Our dataset includes a folder called "test" that contains the main working file "test.csv." It consists of the picture paths and the corresponding class labels for each one. To extract the image path and accompanying labels, we can use the pandas Python package. Our photos must then be resized to 3030 pixels in order to forecast the model and provide a numpy array with image data. We need to import accuracy score from the sklearn.metrics in order to comprehend how the model predicts the actual labels. In order to maintain our trained model, we are finally using the Keras model.save() method.

5.1.3 PREDICTIONS

Lastly, using the trained model to make predictions on the test data, the output displays the name of the traffic sign together with the class ID. The system can predict the correct sign when any input image is presented.

5.2 OUTPUT (GRAPHS) :

THE SIMULATION SOURCE CODE IS GIVEN BELOW THE REFERENCE AND THE OUTPUT IS:

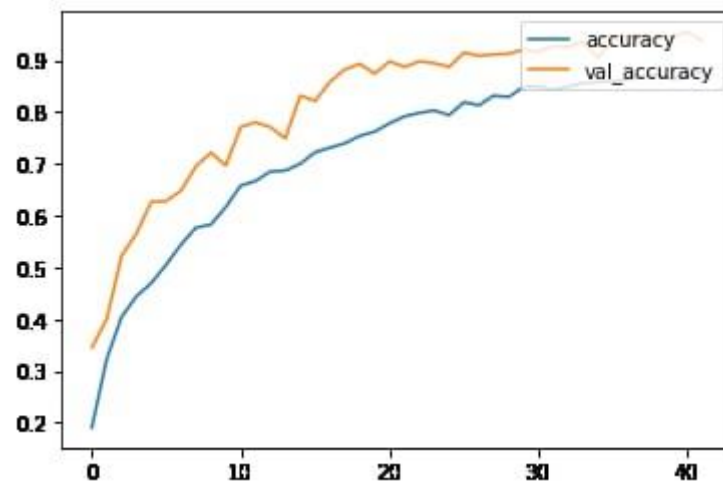


Fig: 5.2.1 ACCURACY

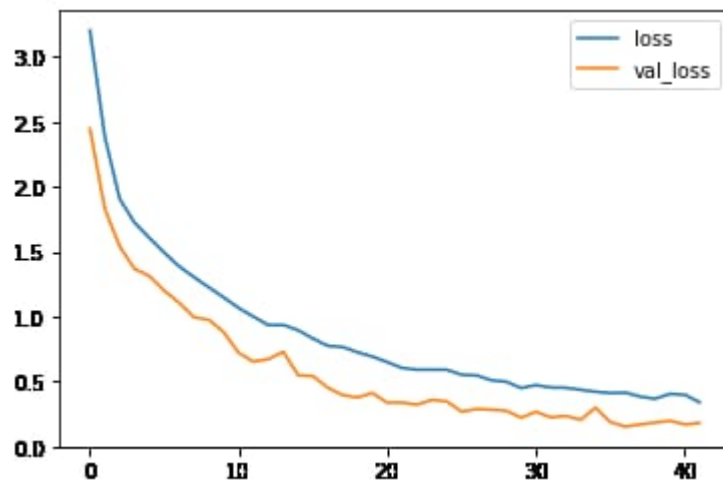


Fig: 5.2.2 LOSS

5.3 CNN ALOGORITHM

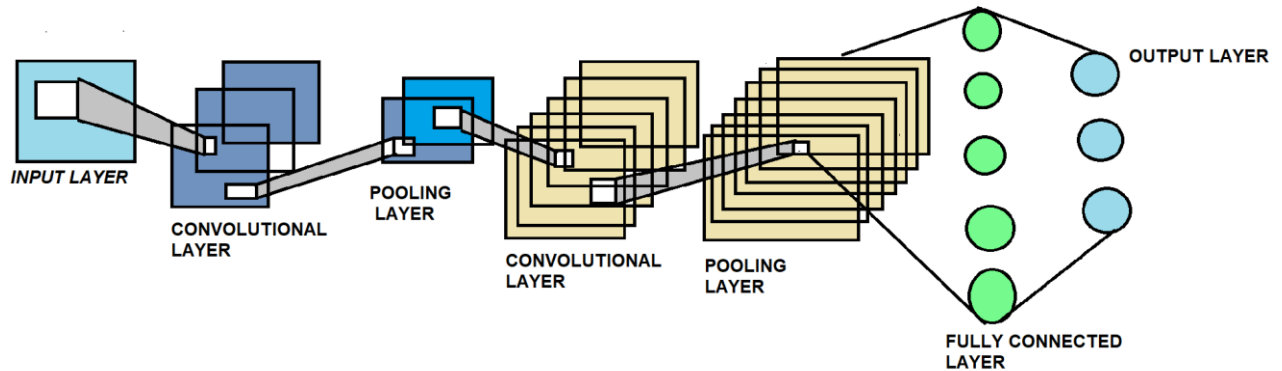


Fig.5.1:Convolutional neural network

The accuracy and loss graph is designed by using CNN Algorithm:

- .
- The basic idea of using a CNN for traffic sign recognition is to train the network on a large dataset of traffic signs with their corresponding labels. The network learns to identify patterns and features in the images that are indicative of different traffic signs.
- The CNN architecture typically consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers.
- The convolutional layers perform feature extraction by applying a set of learned filters to the input image.
- The pooling layers reduce the spatial size of the feature maps, while maintaining important features. Finally, the fully connected layers perform the classification task, by mapping the extracted features to the corresponding traffic sign labels.

- The training process involves adjusting the weights of the network to minimize the difference between the predicted output and the actual label. This process is typically done using backpropagation and gradient descent.
- Once the network is trained, it can be used for traffic sign recognition by feeding the input image into the network, and obtaining the predicted label from the output of the fully connected layer.

Overall, CNNs have shown to be highly effective for traffic sign recognition, achieving state-of-the-art performance on various benchmark datasets.

5.3.1 Comparison between CNN and VGG Algorithm

TABLE.NO 5.1

CNN	VGG
• CNN (Convolutional Neural Network) is a deep learning algorithms used for image classification tasks. • CNN may have only 5-10 layers. • Compared to VGG, CNN is less complex and it gives accurate output.	• Visual Geometry Group is also an deep learning algorithms used for image classification tasks. • VGG networks have 16-19 layers • VGG is a more complex than CNN architecture with a deeper network and it takes more time for output.

5.3.2 Output of CNN

TABLE.NO 5.2

CNN Algorithm	Loss	Val Loss	Accuracy	Val Accuracy
EPOCH 1	0.4924	0.3050	0.8716	0.9157
EPOCH 2	0.0592	0.0293	0.9852	0.9925
EPOCH 3	0.0299	0.0206	0.9922	0.9949
EPOCH 4	0.0251	0.0147	0.9930	0.9959
EPOCH 5	0.0168	0.0135	0.9952	0.9967

5.3.3 Output of VGG

TABLE.NO 5.3

VGG Algorithm	Loss	Val Loss	Accuracy	Val Accuracy
EPOCH 1	0.7315	0.2751	0.7850	0.9242
EPOCH 2	0.2675	0.3169	0.9332	0.9317
EPOCH 3	0.1798	0.1836	0.9554	0.9509
EPOCH 4	0.1659	0.1346	0.9606	0.9609
EPOCH 5	0.1493	0.0135	0.9952	0.9967

5.3.4 Comparing the accuracy value of CNN and VGG Algorithms

TABLE.NO 5.4

Accuracy	
CNN Algorithm	VGG Algorithm
0.96744	0.92588

From the above table it concludes that compared between CNN Algorithm and VGG algorithm CNN will have the more accuracy that we have implemented in our project. Although automatic systems are advantageous regarding the simplicity of the task, the reliability won't be like that from human work. At such, our system has different limitations whose prime cause being the ones we discussed in problem Faced. Resolving these issues will help to enhance our system to a level where following objective could be met:

- Accuracy can be increased using Neural Networks. Advanced image processing algorithms and libraries could be used so that the system can be used efficiently even during unfavorable lighting conditions and during the night time as well.

Here by we conclude that this work is very easy to implement on current system, low cost and durable, ensures maximum safety to passengers and public, the driver gets enough information about road safety. The system is enhanced by automatic speed control when the vehicles get any hazard signal from outside environment. These collision avoidance systems are designed to minimize or avoid collisions, rather than simply minimise harm to passengers during collisions.

5.4 Second Simulation to display the speed limit:

The simulation is built by using flask framework and executed in the PyCharm IDE. In the below program we have given 42 trained data set, whenever the camera identifies the signboard It will display in the digital board according to the condition. These are the following steps were the program runs and gives the output.

The source code of the program is given below the references:

Displaying the output in the web page:

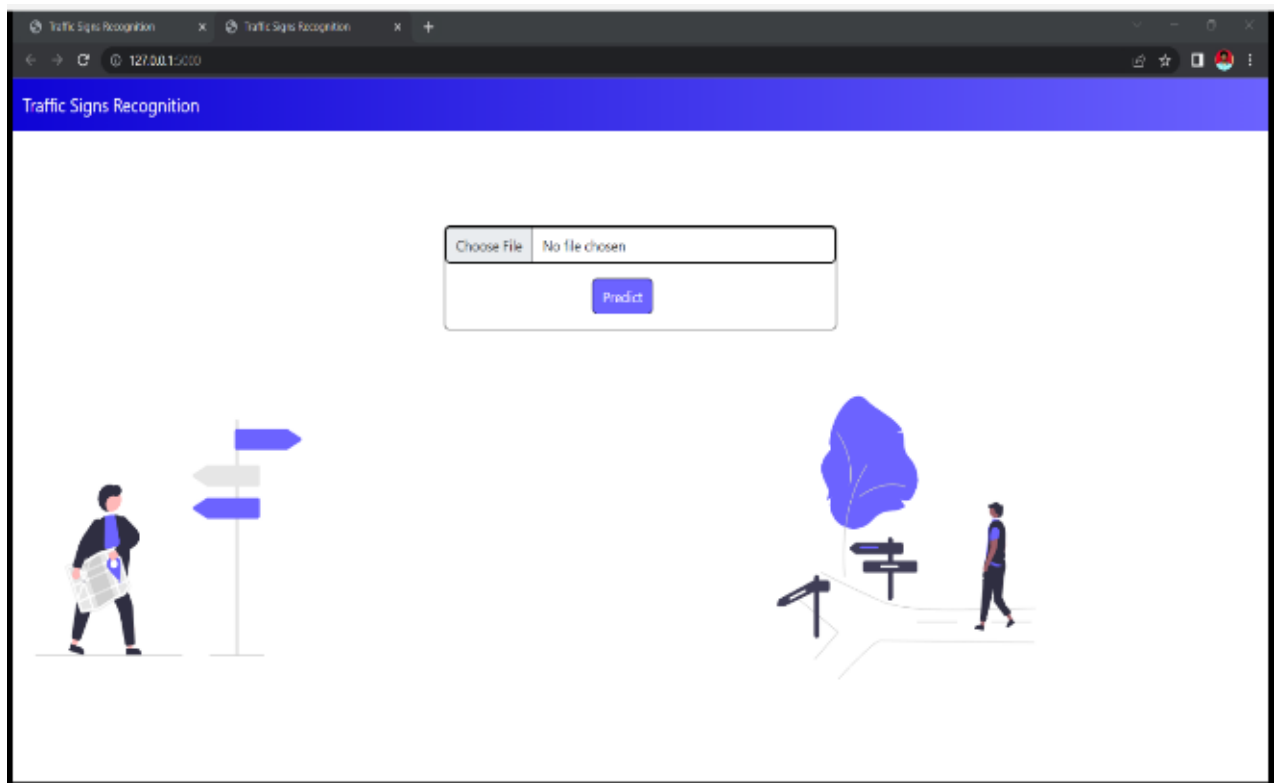


Fig: 5.4.1 Displaying the output by opening the web page by using flask framework and it ask to choose the signboard.

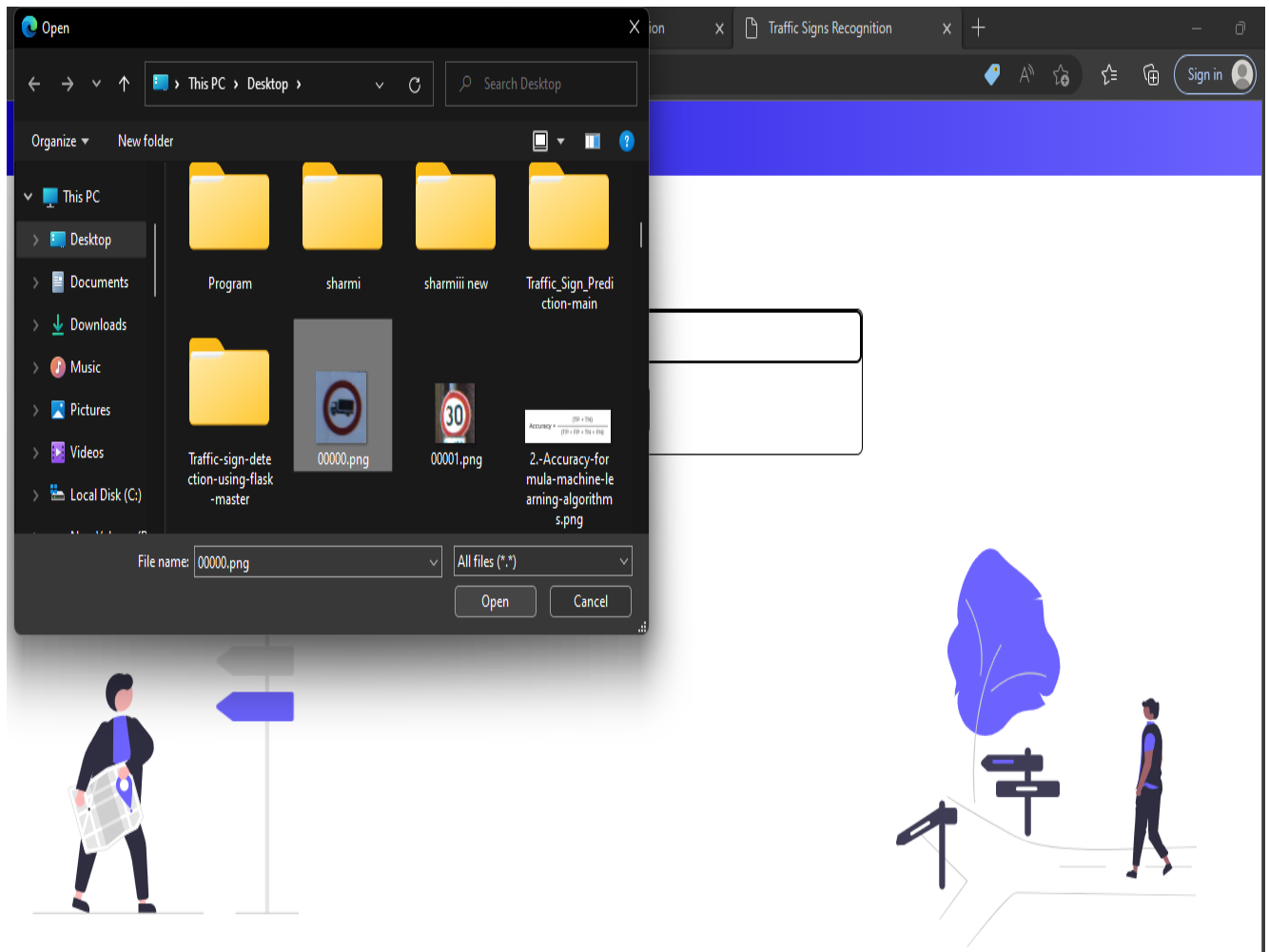


Fig: 5.4.2 *Displaying the output and giving the signboard as the input file to the webpage.*

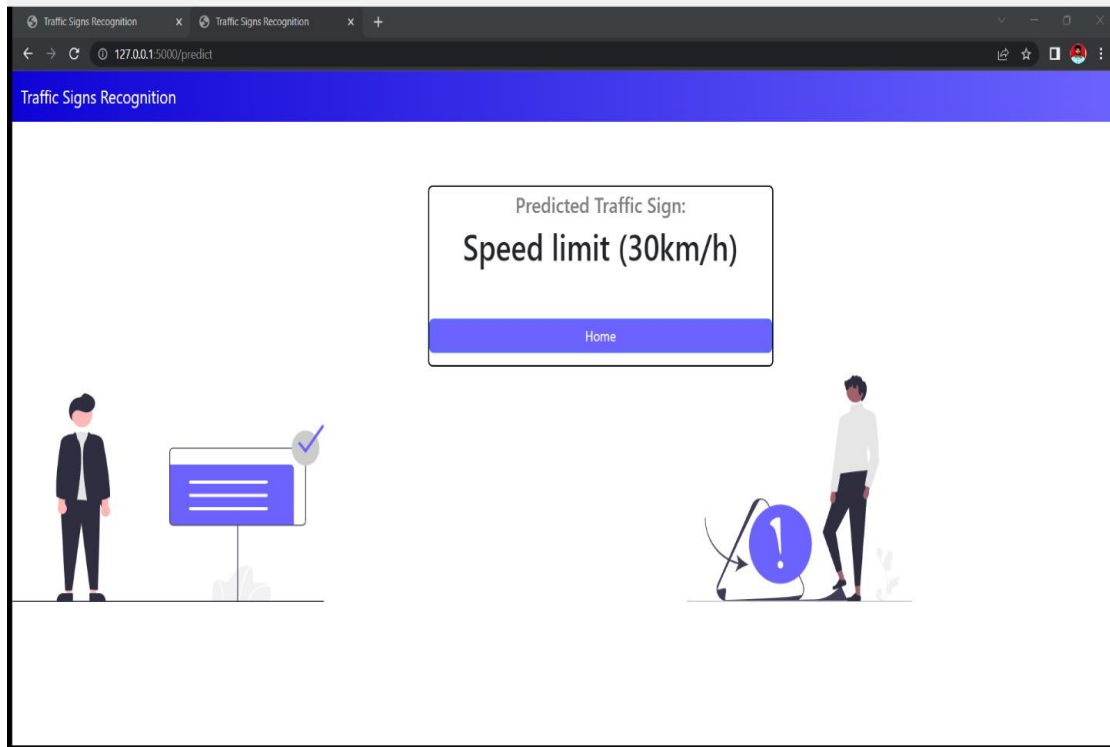


Fig: 5.4.3 Displaying the output that camera recognizes the signboard which has the speed limit of 30Km/h.

CONCLUSION:

The conclusion of traffic sign recognition simulation is that it is a valuable tool for evaluating and improving the performance of traffic sign recognition systems in a controlled and efficient manner. As a result of the results above, we can infer that CNN Algorithm will be more accurate at identifying signboards than VGG Algorithm.

Then the second simulation provides an HTML web page with a flask frame to display the speed limit. Simulations can provide a realistic and challenging environment to test the system's accuracy, robustness, and efficiency under various scenarios and conditions, which can be difficult or impossible to reproduce in physical experiments on real roads. Additionally, simulation can help reduce the cost and time required for testing and evaluation of traffic sign recognition systems, ultimately leading to more efficient and safer transportation systems.

CHAPTER 6

6.1 INTRODUCTION:

Automatic speed control using image processing hardware is designed to enable the detection and recognition of traffic signs in real-time, typically using image processing algorithms and machine learning techniques. The hardware components required for Automatic speed control using image processing include a camera or multiple cameras for capturing images of the road, a processor for image processing and analysis, and a display unit to provide feedback to the driver or control an automated driving system.

6.2 COMPONENTS DISCRIPTION:

6.2.1 WEBCAMERA

- A web camera is a digital camera that is primarily used for capturing video and transmitting it over the internet.
- Web cameras typically connect to a computer via USB or wirelessly, and they can capture both still images and video.
- Web camera is used to capture the signboard and transmit to the system.

6.2.2 ARDUINO

- Arduino is an open-source electronics platform based on easy-to-use hardware and software.
- Arduino can be used as a microcontroller to process the data from a camera or image sensor and recognize traffic signs using image processing algorithms.
- Arduino digital pin is connected the motor driver and input and ground pin is connected to the power supply controller.

6.2.3 MOTOR DRIVER

- A motor driver is an electronic circuit or device that controls the operation of an electric motor.
- It takes input signals from a power supply controller or other devices and uses

them to regulate the speed, direction, and torque of the motor.

- Motor drivers can be used with various types of motors, including DC motors, stepper motors, and BLDC motors. They are typically designed to handle high current and voltage levels to drive the motor effectively.

6.2.4 BLDC MOTOR

- BLDC stands for Brushless DC motor, which is a type of electric motor that operates using DC (direct current) power but does not use brushes to transfer power to the rotor.
- BLDC motors are typically controlled using a motor driver circuit that adjusts the voltage and current supplied to the motor's windings.

6.2.5 POWER SUPPLY CONTROLLER

- A power supply controller is an electronic circuit or device that regulates and controls the output voltage and current of a power supply.
- The power supply controller typically receives feedback signals from Arduino board through input and ground port and gives output as voltage and current to the motor driver, and uses this information to adjust the power supply's input voltage to the motor driver circuit.

6.3 TRAFFIC SIGN RECOGNITION PROGRAM

A traffic sign recognition program is a computer software that can identify and interpret traffic signs from images or video feeds, using machine learning algorithms and computer vision techniques. Such a program can be used in advanced driver assistance systems (ADAS), autonomous vehicles, and smart city applications to help drivers, pedestrians, and traffic management systems navigate safely and efficiently.

Here are the basic steps involved in building a traffic sign recognition program:

- Data collection: Gather a large dataset of traffic sign images, including different types of signs, angles, lighting conditions, and weather conditions.
- Data pre-processing: Clean and preprocess the data to remove noise, standardize image size, and augment the dataset by rotating, flipping, or adding

noise to the images.

- Feature extraction: Extract relevant features from the images, such as color, shape, and texture, using computer vision techniques.
- Training a model: Train a machine learning model, such as a convolutional neural network (CNN), using the extracted features and a labeled dataset.
- Evaluation and tuning: Evaluate the model's performance on a validation set, tune the hyperparameters, and fine-tune the model.
- Deployment: Integrate the trained model into a real-time system, such as an ADAS or an autonomous vehicle, and test it in various scenarios.

There are many open-source libraries and frameworks available for building traffic sign recognition programs, such as TensorFlow, PyTorch, OpenCV, and scikit-learn.

6.4 DATASET



FIG 6.1 MAX SPEED LIMIT

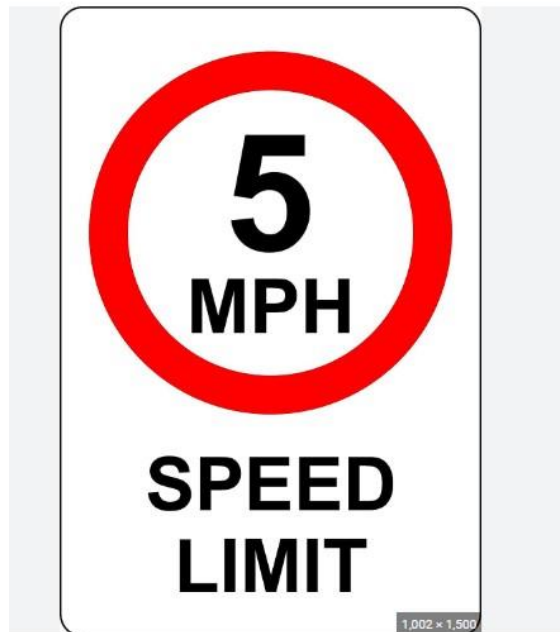


FIG 6.2 MIN SPEED LIMIT

The signboard that is shown in front of the camera is used to change the motor speed automatically based on the signboard if the camera recognizes it. Right here, We have 2 signboards. The motor will operate at its maximum speed whenever the camera detects a speed limit of 50 km/h, and at its lowest speed when the speed limit is 5 km/h.

6.5 HARDWARE KIT (BEFORE WORKING):

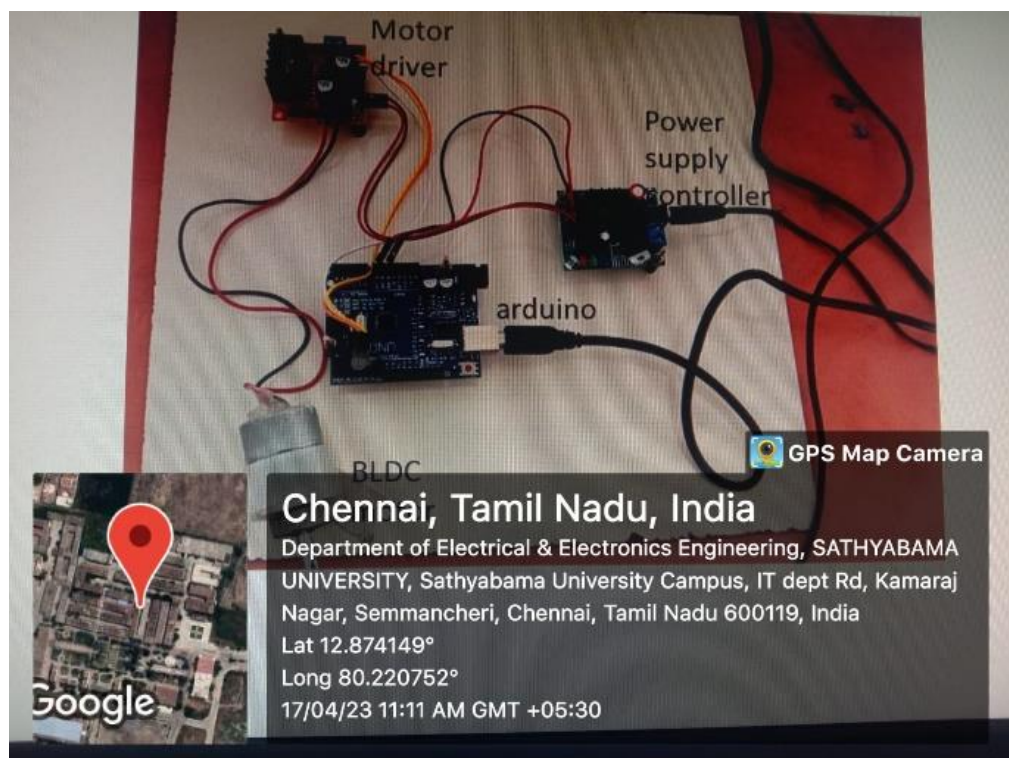


Figure 6.3. hardware kit (Before running the program)

- Whenever the hardware code program gets executed, the webcam will turn on (assuming it is properly connected and configured). The output of the program will depend on what the program is designed to do.
- Webcam is used to recognize signboards in a sign traffic detection program. This is known as computer vision or image processing, and it involves using CNN algorithms to analyze the images captured by the webcam to detect and recognize specific patterns or objects.
- In the case of a sign traffic detection program, the webcam would capture images of the road signs as vehicles pass by, and the program would analyze these images (Sign, etc.).
- This information can then be used to provide alerts or notifications to drivers, or to assist with traffic management and planning.
- However, implementing such a program can be complex, and typically requires a combination of hardware (such as a high-quality webcam), Arduino board, power supply controller, motor driver, BLDC motor, and software (such as computer vision

libraries and machine learning algorithms). It's also important to ensure that any such system complies with relevant privacy and security regulations.

- If the camera identifies a signboard that indicates a reduction in the speed limit, the Arduino can send a signal to the power supply controller, which then sends a signal to the motor driver to reduce the speed of the motor. Then the motor speed will get adjusted according to the signboard.
- According to the signboard the speed limit will be displayed in the screen.

6.6 OUTPUT:

6.6.1 HARDWARE KIT(AFTER WORKING):

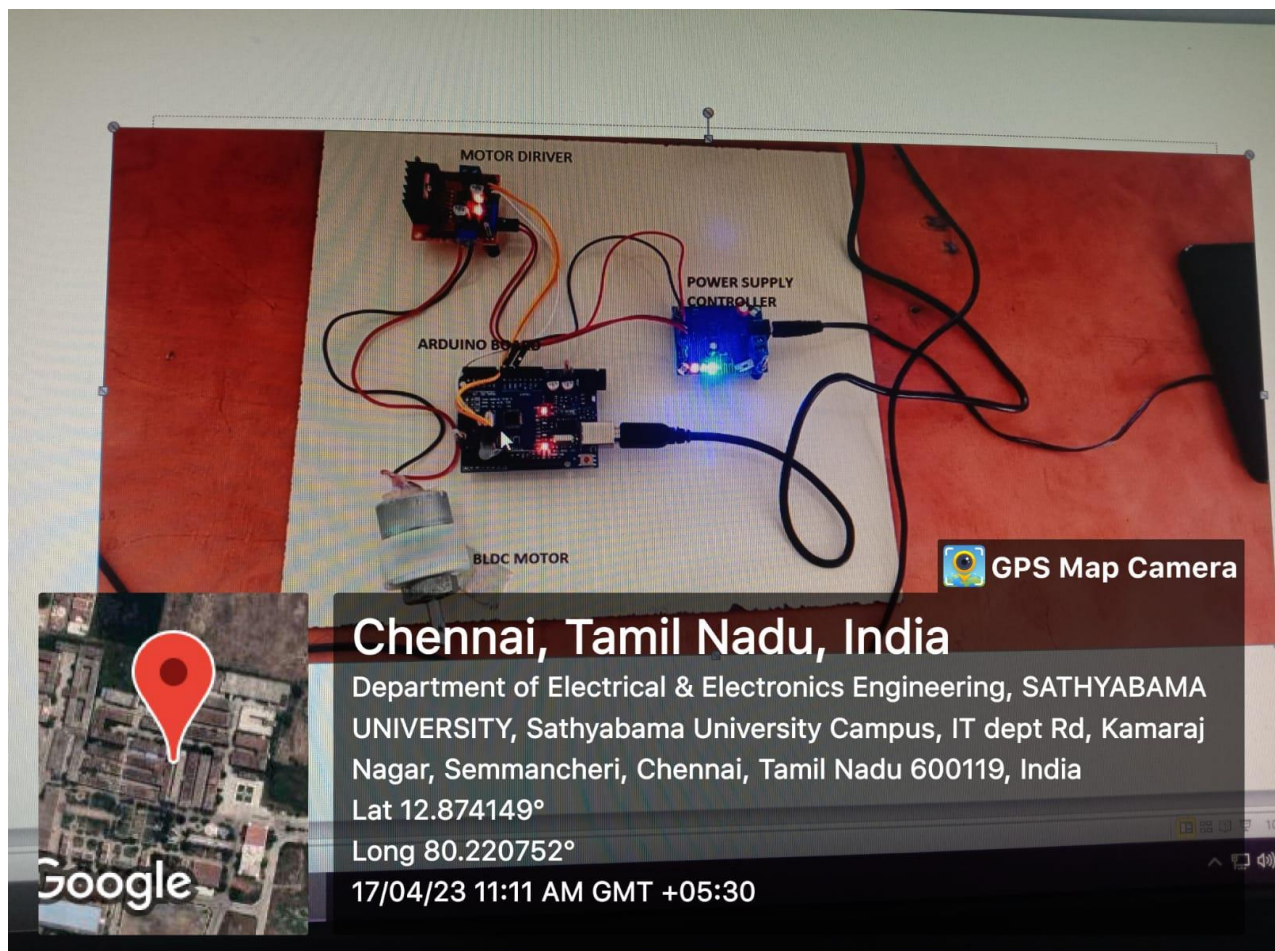
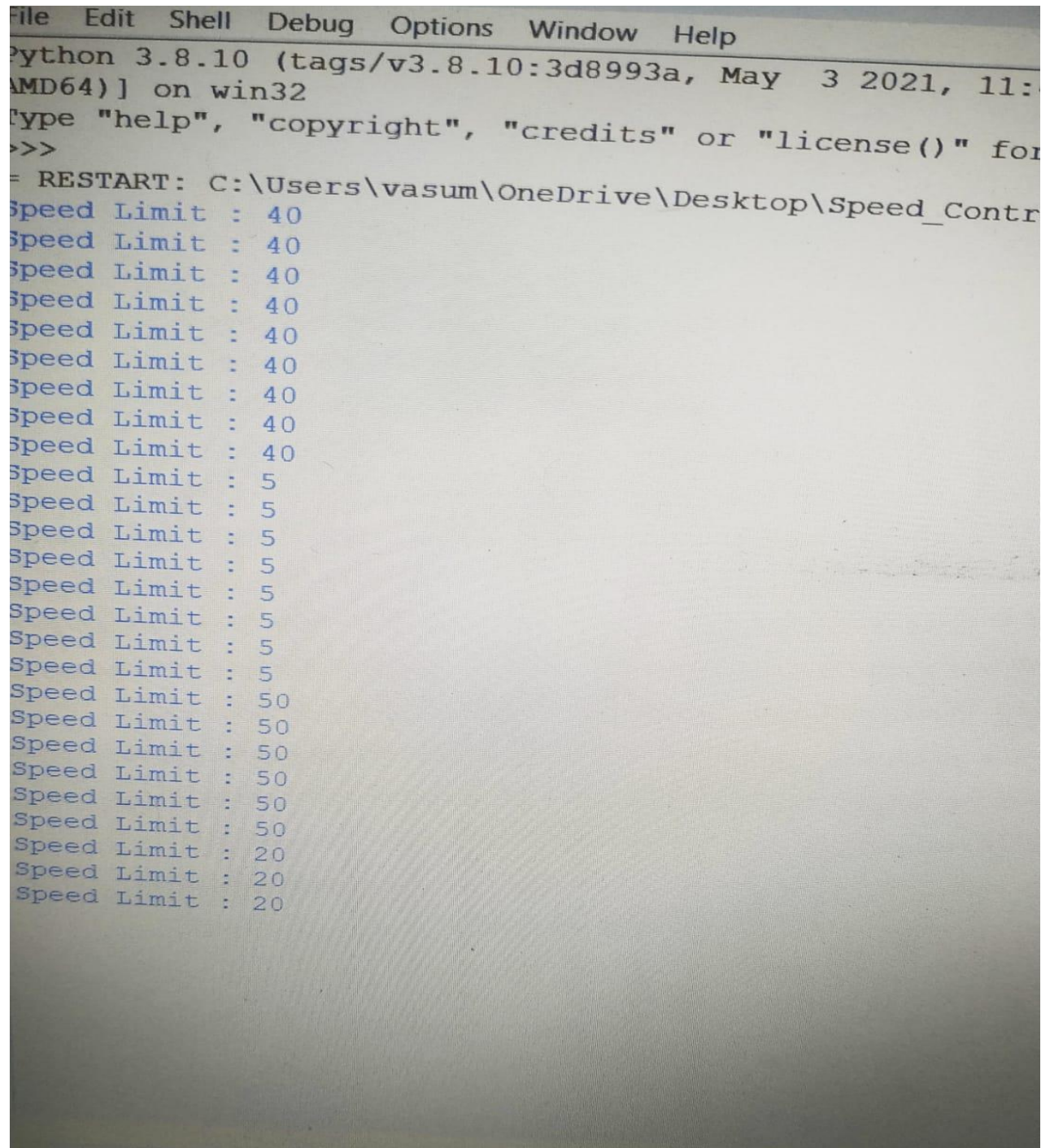


Fig:6.4 The speed of the motor will get adjusted According to the signboard and the speed limit will be displayed in the screen.

6.6.2 SPEED LIMIT DISPLAYED IN SCREEN



The screenshot shows a Python 3.8.10 terminal window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The terminal text shows the Python version and build information, followed by a prompt to type 'help', 'copyright', 'credits', or 'license()'. The user has entered a command that results in a restart of a script located at C:\Users\vasum\OneDrive\Desktop\Speed_Contr. The output of the script is a list of speed limits: 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 50, 50, 50, 50, 50, 50, 50, 50, 20, 20, and 20.

```
File Edit Shell Debug Options Window Help
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
>>>
= RESTART: C:\Users\vasum\OneDrive\Desktop\Speed_Contr
Speed Limit : 40
Speed Limit : 40
Speed Limit : 40
Speed Limit : 40
Speed Limit : 40
Speed Limit : 40
Speed Limit : 40
Speed Limit : 40
Speed Limit : 40
Speed Limit : 40
Speed Limit : 5
Speed Limit : 5
Speed Limit : 5
Speed Limit : 5
Speed Limit : 5
Speed Limit : 5
Speed Limit : 5
Speed Limit : 5
Speed Limit : 5
Speed Limit : 50
Speed Limit : 50
Speed Limit : 50
Speed Limit : 50
Speed Limit : 50
Speed Limit : 50
Speed Limit : 50
Speed Limit : 20
Speed Limit : 20
Speed Limit : 20
```

Fig:6.5.2 The speed limit will be displayed in the screen

6.6.3 Conclusion:

Overall, the hardware requirements for automatic speed control using image processing are significant, but they are achievable with current technology. As image processing and machine learning algorithms continue to improve, we can expect to see more advanced and efficient hardware designs in the future.

6.7 BENEFICATIONS:

The project of traffic sign recognition has several potential benefits, including:

- **Improved road safety:** The system can help drivers recognize traffic signs accurately and quickly, reducing the risk of accidents caused by missed or misinterpreted signs.
- **Enhanced driving experience:** With the system in place, drivers can have a more relaxed and enjoyable driving experience, as they don't have to worry about missing or misinterpreting traffic signs.
- **Increased efficiency:** By accurately recognizing traffic signs, the system can help reduce traffic congestion, improve traffic flow, and reduce travel time.
- **Reduced traffic violations:** The system can help reduce traffic violations, such as speeding or running red lights, by alerting drivers to the presence of traffic signs and reminding them to follow the rules of the road.
- **Improved accessibility:** The system can help make driving more accessible for people with disabilities or who may have difficulty reading traditional traffic signs.
- **Environmental benefits:** With fewer accidents and more efficient traffic flow, the system can help reduce fuel consumption and greenhouse gas emissions.

Overall, the project of traffic sign recognition has the potential to improve road

safety, enhance the driving experience, increase efficiency, and reduce traffic violations, all while benefiting the environment.

6.8 LIMITATIONS:

While traffic sign recognition technology has many potential benefits, it also has some limitations, including:

- Limited accuracy: The system's accuracy may be affected by weather conditions, lighting, and other environmental factors, which can reduce its effectiveness in some situations.
- Limited range: The system may not be able to recognize signs that are too far away or obstructed by other objects, such as trees or buildings.
- Limited coverage: The system may not be able to recognize signs that are specific to certain regions or countries, which can limit its usefulness in some areas.
- Dependence on camera quality: The quality of the camera used for the system can greatly affect its accuracy, and cameras that are dirty, damaged, or improperly calibrated can reduce its effectiveness.
- Limitations in detecting temporary signs: The system may not be able to detect temporary signs, such as signs for construction or events, which can create potential safety hazards.
- High development costs: The development of the system can be expensive, requiring significant investment in research, development, and testing.
- Need for regular updates: The system requires regular updates to ensure that it can recognize new traffic signs and reflect changes in traffic regulations, which can

increase maintenance costs.

Overall, traffic sign recognition technology has limitations that may affect its effectiveness in certain situations and require ongoing maintenance and updates.

6.9 CONCLUSION

However, the weather, the blur from moving vehicles, and the lighting conditions make it difficult to automatically recognise those signs. To overcome these obstacles, Researchers recommended utilising machine learning and image processing acquiring skills. Traffic sign identification and classification are two key components of automatic speed control for automobiles utilising image processing. There are several distinctive traffic signs. choices such as colour, shape, and symbol. Throughout the detection phase, Preprocessing is done to make the input images more metameric in step with their colour or using only math.

The system's primary goal is to decrease accidents caused by disregarding traffic laws and speed limit sign boards. The system's performance is solely dependent on the input image and accuracy of image processing in detecting speed limit sign boards algorithm. In the future, we can enhance CNN to resolve the problem using a classification technique and an image processing approach to quicker and more properly recognise the sign board. Utilizing this proposed device in every car can cut accidents by 30% occurs as a result of speeding and disregarding traffic speed limit signs boards.

As a result of the speed control system, there will be fewer accidents in School zones are kept to a minimum and stillness is enforced there close to the hospital areas This technique is inexpensive, durable, and low- gives optimum performance and power. This investigation has shown that utilising a vehicle speed control system reduces the number of collisions caused by drivers that don't observe traffic laws roadside speed signs in designated areas. It highlights the significance of decreasing vehicle speed and its effects on pedestrian safety automatically and user safety on roads

These excellent vehicle speed controls make a major improvement in safety by assuring the security of both road users and pedestrians. It permits the car to move separately to slow the car down as it gets closer and faster speed, which actively and effectively

lowers collisions brought on by driver error.

6.10 Future scope of Automatic speed control using image processing:

Automatic speed control using image processing is an important area of research in the field of computer vision and autonomous driving. As technology continues to advance, the future scope of traffic sign recognition is likely to expand and become increasingly important. Some potential areas of development include:

- **Autonomous vehicles:** With the growing popularity of self-driving cars, traffic sign recognition will play a crucial role in ensuring the safety of these vehicles. Accurate and reliable recognition of traffic signs can help autonomous vehicles make better decisions and improve their ability to navigate complex roadways.
- **Improved road safety:** Accurate recognition of traffic signs can help reduce the number of accidents on the road by alerting drivers to important information such as speed limits, road conditions, and upcoming hazards.
- **Smart cities:** As cities become more connected and intelligent, traffic sign recognition can be integrated with other smart city technologies to create more efficient transportation systems. For example, traffic sign recognition can be used to control traffic lights and optimize traffic flow.
- **Enhanced driver assistance:** Traffic sign recognition can be integrated with existing driver assistance systems to provide more comprehensive information to drivers. This can help improve driver awareness and reduce the risk of accidents.

Overall, the future scope of traffic sign recognition is promising, with a wide range of potential applications that could benefit both drivers and society.

REFERENCE

- Karthikeyan D, Enitha C, Bharathi S, Durkadevi K, 2020, Traffic Sign Detection and Recognition using Image Processing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCICT – 2020 (Volume 8 – Issue 08).
- Aman Sharma , Rohit Chib , Deeptanshu Samotra , Rohit Bharti,Shefali, 2021, Advanced Traffic Signal Monitoring using Programmable Logic Controller (PLC), INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 12 (December 2021).
- G.Revathi, Dr.G.Balakrishnan,2016 , Indian Sign Board Recognition Using Image Processing Techniques ,InternationalJournal of Advanced Research in Biology Engineering Science andTechnology (IJARBEST) Vol. 2, Special Issue 15, March 2016
- Yuji Goad, Lifang Zhang, Seiichi Sericea, “Proposal a Vehicle Speed Measuring System Using Image Processing” International Symposium on Computer, Consumer and Control (2014).
- Aditya Pratap Singh, Udit Narayan, Akash Verma, “Speed Control of DC Motor using Pid Controller Based on MATLAB” InternationalConference on Recent Trends in Applied Sciences with Engineering Applications(2013).
- Anju Manjooran, Anphy Varghese, Annmariya Seby and Krishnadas J, “Traffic Sign Board Detection and Voice Alert SystemAlong with Speed Control” Asian Journal of Applied Science and Technology (AJAST) Volume 2, Issue 1, Pages 28 1-286, 2018Z. Sun, G. Bebis, and R. Miller, “On-road vehicle detection usingoptical sensors a review,” IEEE International Conference on Intelligent Transportation Systems, Washington, DC, October 3-6,585-590. 2004.
- FrankLindner, Ulrich Kressel , and Stephan Kaelberer," Real-timeVision For Intelligent Vehicles” IEEE Instrumentation & Measurement Magazine June 2001.

- Shi, Y.Q. and Yang, X.G. (2013) The Public Transportation System of High Quality in Taiwan. 2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS), Mexico City, 6-8 March 2013, 1-6.
<https://doi.org/10.1109/ISADS.2013.6513426>
- Tarnoff, P.J., Bullock, D.M., Young, S.E., et al. (2009) Continuing Evolution of Travel Time Data Information Collection and Processing. Transportation Research Board Annual Meeting, Washington, 11-15 January 2009, Paper #09-2030.
- Prem Kumar, V., Barath, V. and Prashanth, K. (2013) Object Counting and Density Calculation Using MATLAB.
- Lillywhite, K., Lee, D.J., Tippetts, B. and Archibald, J. (2013) A Feature Construction Method for General Object Recognition.
Pattern Recognition, 46, 3300.
<https://doi.org/10.1016/j.patcog.2013.06.002>
- Gamer, J.E., Lee, C.E. and Huang, L.R. (1990) Center for Transportation Research, the University of Texas at Austin. Infrared Detectors for Counting, Classifying, and Weighing Vehicles, 3- 1088/0-1162.
- Niksaz, P. (2012) Automatic Traffic Estimation Using Image Processing. International Conference on Image, Vision and Computing, Iran, V50.11.
DOI: 10.4236/jsea.2017.109042776
- Pandit, V., Doshi, J., Mehta, D., Mhatre, A. and Janardhan, A. (2014) Smart Traffic Control System Using Image Processing. International Journal of Emerging Trends & Technology in Computer Science, 3, 5902-5906.
- Akoum, A., Daya, B. and Chauvet, P. (2010) Two Neural Networks for License Number Plates Recognition. Journal of Theoretical and Applied Information Technology, 12, 25-32.

- Pancharatnam, M. and Sonnadara, D.U.J. (2010) Vehicle Counting and Classification from a Traffic Scene. The 26th National ITConference, Colombo, Sri Lanka, 72- 78.
- Linganagouda, R., Raju, P. and Patil, A. (2016) Automatic Intelligent Traffic Control System. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 5, 5902-5906.
- Fleet, D.J. and Weiss, Y. (2005) Toronto, Fleet Research OpticalFlow Estimation, Flow Chapter 05.

APPENDIX SOURCE CODE:

CODE USED TO PREDICT ACCURAY AND LOSS GRAPH:

```
import numpy as np # linear algebra
import pandas as pd
#import packages
import matplotlib.pyplot as plt
#import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
import os
import tensorflow as tf
from tensorflow import keras#image processing,extract
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, InputLayer, Reshape,
MaxPooling2D, Flatten
from tensorflow.keras import layers
from keras.layers import Dropout#dropping the 2d layers
from keras.layers import BatchNormalization#recentering and re scaling
from keras.preprocessing.image import ImageDataGenerator

import cv2#resize,rgb,image processing computer vision library
from PIL import Image#python image library
from sklearn.model_selection import train_test_split

# assign our data, test and train path's and show how many categories there are

data = 'dataset/'
categories = len(os.listdir(data))
print('Categories:', categories)

#Get all images and their labels and add them to our lists
```

```

data = []
labels = []
total_classes = 43
height = 43
width = 43
channels = 3

for i in range(categories):
    path = 'dataset/' + str(i) #goes through all 43 subfolders of the dataset folder
    images = os.listdir(path) #gets a list of entries in our path, so our images

    for img in images: #loops through all images in our path
        try:
            image = cv2.imread(path + '/' + img) #reads the images into a numpy array
using open cv
            image_fromarray = Image.fromarray(image, 'RGB') #transforms the arrays to
images
            resize_image = image_fromarray.resize((32, 32)) #resizes the images to
32x32
            data.append(np.array(resize_image)) #adds the arrays to our 'data' list
            labels.append(i) #adds the labels to our 'labels' list
        except:
            print("Error in loading image")

data = np.array(data)#converting the data as array
labels = np.array(labels)#converting the label as the array

data.shape, labels.shape

x_train, x_val, y_train, y_val = train_test_split(data, labels, test_size=0.2,
random_state=42, shuffle=True)#splitted as x train shape and y traain shape
x_train.shape, x_val.shape, y_train.shape, y_val.shape

x_train = x_train/255#normalise the traij data

```

```
x_val = x_val/255
```

```
y_train = keras.utils.to_categorical(y_train, categories)
```

```
y_val = keras.utils.to_categorical(y_val, categories)
```

```
y_train.shape, y_val.shape
```

```
# Define Model
```

```
model=Sequential()
```

```
model.add(InputLayer(input_shape=(32,32,3)))
```

```
model.add(Reshape(target_shape=(32,32, 3)))
```

```
model.add(Conv2D(filters=32,kernel_size=(3,3),activation='relu')) # increased filter  
from 12 to 32
```

```
model.add(BatchNormalization()) #added Batch Normalization
```

```
model.add(Conv2D(filters=32,kernel_size=(3,3),activation='relu'))
```

```
model.add(BatchNormalization())
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Dropout(0.2))
```

```
model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu')) # increased filter  
from 32 to 64
```

```
model.add(BatchNormalization())
```

```
model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu'))
```

```
model.add(BatchNormalization()) #added Batch Normalization
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Dropout(0.4)) #increased the dropout percentage from 0.2 to 0.4
```

```
model.add(Flatten())
```

```
model.add(Dense(256, activation='relu'))
```

```
model.add(BatchNormalization())
```

```
model.add(Dense(43, activation='softmax')) #we have 43 categories
```

```
# Compile model 'categorical_crossentropy'-yes or no,binary_crossentropy-binary  
classification
```

```
model.compile(loss='categorical_crossentropy', optimizer='adam',
```

```

metrics=['accuracy'])#loss percent
#model.compile(loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),optimizer="adam",metrics="accuracy")
# dataset model
history = model.fit(x_train, y_train, batch_size=64, epochs=5, validation_data=(x_val, y_val))
racy=model.evaluate(x_val,y_val)#training the data

print('Test accuracy', accuracy)

model.save("my_model.h5")

#plotting graphs for accuracy
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.show()

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()
plt.show()

```

CODE USED TO CREATE WEB PAGE THAT DISPLAYS SIGNBOARD:

```
import os
import numpy as np
from flask import*
import tensorflow
from keras.models import load_model
from werkzeug.utils import secure_filename
app = Flask(__name__)
# Classes of traffic signs
classes = { 0:'Speed limit (20km/h)',
1:'Speed limit (30km/h)',
2:'Speed limit (50km/h)',
3:'Speed limit (60km/h)',
4:'Speed limit (70km/h)',
5:'Speed limit (80km/h)',
6:'End of speed limit (80km/h)',
7:'Speed limit (100km/h)',
8:'Speed limit (120km/h)',
9:'No passing',
10:'No passing veh over 3.5 tons',
11:'Right-of-way at intersection',
12:'Priority road',
13:'Yield',
14:'Stop',
15:'No vehicles',
16:'Vehicle > 3.5 tons prohibited',
17:'No entry',
18:'General caution',
19:'Dangerous curve left',
20:'Dangerous curve right',
21:'Double curve',
22:'Bumpy road',
23:'Slippery road',
24:'Road narrows on the right',
25:'Road work',
26:'Traffic signals',
27:'Pedestrians',
28:'Children crossing',
29:'Bicycles crossing',
30:'Beware of ice/snow',
31:'Wild animals crossing',
32:'End speed + passing limits',
33:'Turn right ahead',
34:'Turn left ahead',
35:'Ahead only',
36:'Go straight or right',
37:'Go straight or left',
38:'Keep right',
39:'Keep left',
40:'Roundabout mandatory',
```

```

41:'End of no passing',
42:'End no passing vehicle > 3.5 tons' }
def image_processing(img):
model = load_model('./model/TSR.h5')
data = []
img = tensorflow.keras.utils.load_img(img,target_size = (30,30))
img = tensorflow.keras.utils.img_to_array(img)
data.append(img)
X_test = np.array(data)
Y_pred = model.predict(X_test)
Y_pred = model.predict(X_test)
a = np.argmax(Y_pred,axis=1)
return a[0]
@app.route('/')
def index():
return render_template('/index.html')
@app.route('/more')
def more():
return render_template('more.html')
@app.route('/predict',methods=['GET','POST'])
def upload():
if request.method == 'POST':
f = request.files['file']
file_path = secure_filename(f.filename)
f.save(file_path)
result = image_processing(file_path)
result = classes[result]
os.remove(file_path)
return render_template('/result.html', result = result)
return None
if __name__ == '__main__':
app.run(debug=True

```

CODE USED FOR HARDWARE:

PROGRAM:

```
import cv2
import numpy as np
from subprocess import call
import time
import smtplib
import serial

arduino = serial.Serial(port='COM8', baudrate=9600, timeout=0.1)

mode_status = 0

match_=0
MIN_MATCH_COUNT=35

#detector = cv2.SIFT()
detector = cv2.SIFT_create()
#detector = cv2.xfeatures2d.SIFT_create()
FLANN_INDEX_KDTREE=0
flannParam=dict(algorithm=FLANN_INDEX_KDTREE,tree=5)
flann=cv2.FlannBasedMatcher(flannParam,{})

nt=cv2.imread('dataset/nt.jpg',0)
ntKP,ntDecs=detector.detectAndCompute(nt,None)

ar=cv2.imread('dataset/ar.jpg',0)
arKP,arDecs=detector.detectAndCompute(ar,None)

er=cv2.imread('dataset/er.jpg',0)
erKP,erDecs=detector.detectAndCompute(er,None)
```

```

pr=cv2.imread('dataset/pr.jpg',0)
prKP,prDecs=detector.detectAndCompute(pr,None)

cam=cv2.VideoCapture(0)
cam.set(3,240)
cam.set(4,360)
while True:
    for x in range(0,4):
        if(x==0):
            MIN_MATCH_COUNT=80
            match_=0
            ret,QueryImgBGR=cam.read()
            QueryImg=cv2.cvtColor(QueryImgBGR,cv2.COLOR_BGR2GRAY)
            queryKP,queryDesc=detector.detectAndCompute(QueryImg,None)
            matches=flann.knnMatch(queryDesc,ntDecs,k=2)
            goodMatch=[]
            for m,n in matches:
                if(m.distance<0.75*n.distance):
                    goodMatch.append(m)
            if(len(goodMatch)>MIN_MATCH_COUNT):
                cv2.waitKey(10)
                tp=[]
                qp=[]
                for m in goodMatch:
                    tp.append(ntKP[m.trainIdx].pt)
                    qp.append(queryKP[m.queryIdx].pt)
                tp,qp=np.float32((tp,qp))
                H,status=cv2.findHomography(tp,qp,cv2.RANSAC,3,0)
                h,w=nt.shape
                traingBorder=np.float32([[[0,0],[0,h-1],[w-1,h-1],[w-1,0]]])
                queryBorder=cv2.perspectiveTransform(traingBorder,H)
                cv2.polylines(QueryImgBGR,[np.int32(queryBorder)],True,(0,255,0),5)
                print ("Speed Limit : 5")
                arduino.write(b'1')

```

```

else:
    match_=len(goodMatch)
    #print "nt enough matches -
%d/%d"%(len(goodMatch),MIN_MATCH_COUNT)
    cv2.imshow('result',QueryImgBGR)
    cv2.waitKey(10)
    cv2.imwrite('Captured.jpg',QueryImgBGR)
elif(x==1):
    MIN_MATCH_COUNT=80
    match_=0
    ret,QueryImgBGR=cam.read()
    QueryImg=cv2.cvtColor(QueryImgBGR,cv2.COLOR_BGR2GRAY)
    queryKP,queryDesc=detector.detectAndCompute(QueryImg,None)
    matches=flann.knnMatch(queryDesc,arDecs,k=2)
    goodMatch=[]
    for m,n in matches:
        if(m.distance<0.75*n.distance):
            goodMatch.append(m)
    if(len(goodMatch)>MIN_MATCH_COUNT):
        cv2.waitKey(10)
        tp=[]
        qp=[]
        for m in goodMatch:
            tp.append(arKP[m.trainIdx].pt)
            qp.append(queryKP[m.queryIdx].pt)
        tp,qp=np.float32((tp,qp))
        H,status=cv2.findHomography(tp,qp,cv2.RANSAC,3,0)
        h,w=ar.shape
        traingBorder=np.float32([[[0,0],[0,h-1],[w-1,h-1],[w-1,0]]])
        queryBorder=cv2.perspectiveTransform(traingBorder,H)
        cv2.polylines(QueryImgBGR,[np.int32(queryBorder)],True,(0,255,0),5)
        print("Speed Limit : 50")
        arduino.write(b'2')

```

```

else:
    match_=len(goodMatch)
    #print "ar enough matches -
%d/%d"%(len(goodMatch),MIN_MATCH_COUNT)
    cv2.imshow('result',QueryImgBGR)
    cv2.waitKey(10)
    cv2.imwrite('Captured.jpg',QueryImgBGR)
elif(x==2):
    MIN_MATCH_COUNT=80
    match_=0
    ret,QueryImgBGR=cam.read()
    QueryImg=cv2.cvtColor(QueryImgBGR,cv2.COLOR_BGR2GRAY)
    queryKP,queryDesc=detector.detectAndCompute(QueryImg,None)
    matches=flann.knnMatch(queryDesc,erDecs,k=2)
    goodMatch=[]
    for m,n in matches:
        if(m.distance<0.75*n.distance):
            goodMatch.append(m)
    if(len(goodMatch)>MIN_MATCH_COUNT):
        cv2.waitKey(10)
        tp=[]
        qp=[]
        for m in goodMatch:
            tp.append(erKP[m.trainIdx].pt)
            qp.append(queryKP[m.queryIdx].pt)
        tp,qp=np.float32((tp,qp))
        H,status=cv2.findHomography(tp,qp,cv2.RANSAC,3,0)
        h,w=er.shape
        traingBorder=np.float32([[[0,0],[0,h-1],[w-1,h-1],[w-1,0]]])
        queryBorder=cv2.perspectiveTransform(traingBorder,H)
        cv2.polylines(QueryImgBGR,[np.int32(queryBorder)],True,(0,255,0),5)
        print ("Speed Limit : 40")
        arduino.write(b'3')

```

```

else:
    match_=len(goodMatch)
    #print "ti enough matches -
%d/%d"%(len(goodMatch),MIN_MATCH_COUNT)
    cv2.imshow('result',QueryImgBGR)
    cv2.waitKey(10)
    cv2.imwrite('Captured.jpg',QueryImgBGR)
elif(x==3):
    MIN_MATCH_COUNT=80
    match_=0
    ret,QueryImgBGR=cam.read()
    QueryImg=cv2.cvtColor(QueryImgBGR,cv2.COLOR_BGR2GRAY)
    queryKP,queryDesc=detector.detectAndCompute(QueryImg,None)
    matches=flann.knnMatch(queryDesc,prDecs,k=2)
    goodMatch=[]
    for m,n in matches:
        if(m.distance<0.75*n.distance):
            goodMatch.append(m)
    if(len(goodMatch)>MIN_MATCH_COUNT):
        cv2.waitKey(10)
        tp=[]
        qp=[]
        for m in goodMatch:
            tp.append(prKP[m.trainIdx].pt)
            qp.append(queryKP[m.queryIdx].pt)
        tp,qp=np.float32((tp,qp))
        H,status=cv2.findHomography(tp,qp,cv2.RANSAC,3,0)
        h,w=pr.shape
        traingBorder=np.float32([[[0,0],[0,h-1],[w-1,h-1],[w-1,0]]])
        queryBorder=cv2.perspectiveTransform(traingBorder,H)
        cv2.polylines(QueryImgBGR,[np.int32(queryBorder)],True,(0,255,0),5)
        print("Speed Limit : 20")
        arduino.write(b'4')

```

```

else:
    match_ = len(goodMatch)
    #print "pr enough matches -
%d/%d"%(len(goodMatch),MIN_MATCH_COUNT)
    cv2.imshow('result',QueryImgBGR)
    cv2.waitKey(10)
    cv2.imwrite('Captured.jpg',QueryImgBGR)

hsv = cv2.cvtColor(QueryImgBGR,cv2.COLOR_BGR2HSV)
gray = cv2.cvtColor(QueryImgBGR,cv2.COLOR_BGR2GRAY)
    # Create a binary image with where white will be skin colors and rest is black
mask2 = cv2.inRange(hsv, np.array([2,0,0]), np.array([20,255,255]))
    # Kernel for morphological transformation
kernel = np.ones((5,5))
    # Apply morphological transformations to filter out the background noise
dilation = cv2.dilate(mask2, kernel, iterations = 1)
erosion = cv2.erode(dilation, kernel, iterations = 1)
    # Apply Gaussian Blur and Threshold
filtered = cv2.GaussianBlur(erosion, (3,3), 0)
ret,thresh = cv2.threshold(filtered, 127, 255, 0)
    # Show threshold image
#cv2.imshow("Thresholded", thresh)
#cv2.imshow('GRAY',gray)
#cv2.imshow('HSV',hsv)
arduino.flushInput()
arduino.flushOutput()
time.sleep(0.1)

cap.release()
cv2.destroyAllWindows()

```