

DRIVER ASSISTANCE & ALERTNESS SYSTEM

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering degree in

Electronics and Communication Engineering

By

PATURU DHEERAJ REDDY (38130157)

MADIREDDY SAINATHA REDDY (38130122)



SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

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

March - 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119
www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of

PATURU DHEERAJ REDDY (38130157)

MADIREDDY SAINATHA REDDY (38130122)

who carried out the project **driver assistance & alertness system**
under our supervision from December 2021 to March 2022

Internal guide

Dr.V.Balamurugan, M.Tech., Ph.D.

Head of the Department

Dr.T.Ravi, M.E, Ph.D.

Submitted for Viva voce Examination held on 07/05/2022

Internal Examiner

Dr.V.Balamurugan

External Examiner

Dr.Nirmal Raj

Declaration

We declare that the report titled “**Driver Assistance &Alertness**” submitted by us is an original work done by us under the guidance **of Dr V Balamurugan School of Electrical and Electronics Engineering, SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** during the final semester , in the School of Electrical and Electronics Engineering. The work is original and wherever I/We have used materials from other sources, I/We have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

DATE:11-03-2022

PLACE: Chennai

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. N.M. Nandhitha, Dean, School of Electrical and Electronics Engineering and Dr. T. Ravi, Head of the Department, of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. Balamurugan for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

Abstract

Driver assisting & alerting system which includes automatic detection and recognition of traffic signs and drowsiness detection are challenging now-a-days. Driver negligence is one of the reasons for the road accidents which became an alarming issue in day-to-day life. When driving long, driver may fall asleep or fail to recognize the traffic signs exactly which increases the chance of accident. There are lot many factors which need to be considered to reduce the road accidents. But, currently we have focused on traffic signs and drowsiness detection in our project. The developed system helps the vehicle driver to stay awake with the help of the alarm and also displays the traffic signs which appear on road by classifying it accordingly.

In this project, we have two modules Automatic Detection and Recognition of the Traffic Signs and Drowsiness detection. There are traffic signs such as turn right or left, speed limit, double curve etc. The process of recognizing the respective class of the traffic sign is known as traffic signs classification. Detection of traffic signs is more towards the automation using Convolution Neural Network (CNN) algorithm which is a deep learning algorithm. An open database is utilized for training and testing the system. Once the model has been trained, we can use that particular model to recognize the traffic signs. Drowsiness detection is developed to locate the eyes of the driver by capturing the video stream through Open CV library available in python. Once eyes are located, the distance between the eyelids are calculated and accordingly if found drowsy the alarm is played.

The created system is utilized to diminish the quantity of accidents by assisting and alerting the driver in real time by improvising it to the real time specifications and integrating with hardware.

For advanced driver assistance, automatic traffic sign recognition and drowsiness alert has been designed in this system. Driver and pedestrian inattention is one of the main causes of road traffic accidents. Among the key tasks for both active driver assistance systems and control systems for autonomous vehicles, the analysis of marking and traffic signs occupies an important place. Cab drivers, transport drivers, truck drivers and individuals voyaging significant distance experience the ill effects of absence of rest. Because of which it turns out to be risky to drive when feeling lethargic. Most of

accidents occur because of the drowsiness of the driver. To prevent these accidents we will build a system using Python, OpenCV, and Keras which will alert the driver when he feels sleepy and a deep neural network model that can classify traffic signs present in the image into different categories through GUI.

Here we have different models for automatic traffic sign recognition and driver drowsiness alert operated through a single GUI. In automatic traffic sign recognition system we use a dataset that has 43 different classes of several images. Then we build a CNN and we train the model. For driver drowsiness alert we find the euclidean distance between eye lids. Then we calculate the eye aspect ratio which we use it as a reference to detect the drowsiness. An alarm sound will alert the driver as soon as it detects the drowsiness.

GUI is designed such that you have the option to upload the image which is a traffic sign classification and drowsiness detection where laptop cam is opened to start the video stream.

Specific Contribution

- Have developed the drowsiness alert system using shape detector with image processing techniques and have integrated this model with traffic sign recognition through GUI

Specific Learning

- Have learnt significance of shape detector and implementation of image processing techniques in real time applications as well as the libraries we utilized in the project

Table of Contents

TITLE	PAGE
Bona-fide Certificate	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
List of Figures	viii
Abbreviations	ix
1. Introduction	
1.1. Introduction to Driver Assistance & Alertness System	
1.1.1. Automatic Detection and Recognition of Traffic Signs	1
1.1.2. Drowsiness Detection	2
1.2. Motivation	3
1.3. Objective	3
1.4. Scope of the project	3
2. Literature Survey	
2.1. Automatic Detection and Recognition of Traffic Signs	4
2.2. Drowsiness Detection	6
3. Methodology	
3.1. Driver Assistance & Alertness System	8
3.2. Automatic Detection and Recognition of Traffic Signs	
3.2.1. CNN Algorithm	8
3.2.2. Steps involved in developing the model	11
3.3. Drowsiness Detection	15
3.4. Tools and Libraries used	18
4. Results and Discussion	19
5. Conclusions and Further Work	24
6. References	26

List of Figures

Figure No	Title	Page No
1.1	Venn diagram of components of Artificial Intelligence(AI)	2
3.1	Block diagram of Driver Assistance & Alertness System	8
3.2	Layers of CNN	9
3.3	Flattening and Fully Connected Layer	11
3.4	Images of Traffic signs	12
3.5	Testing Dataset	12
3.6	Flow chart of the CNN model	14
3.7	Visualizing the 68 facial landmark coordinates	16
3.8	The six landmarks of face associated with the eye	17
4.1	Accuracy and Loss values of the model while training	19
4.2	Graph representing the training and validation accuracy	20
4.3	Graph representing the training and validation loss	20
4.4	GUI of Driver Assistance & Alertness System	21
4.5	GUI of Traffic Signs 1	22
4.6	GUI of Traffic Signs 2	22
4.7	Drowsiness Detection	23

Abbreviations

The following technical abbreviations are used in the report

AI	Artificial Intelligence
CNN	Convolutional Neural Network
EAR	Eye Aspect Ratio
GUI	Graphical User Interface
OpenCV	Open Source Computer Vision Library
ROI	Region of Interest
RELu	Rectified Linear Unit

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO DRIVER ASSISTANCE & ALERTNESS SYSTEM

Many people travelling on highways day and night experiences drowsiness and inattention to traffic signs which leads to lot accidents costing loss of lives. The amount of accidents is increasing every year in traffic accidents due to human errors. 17.8% of accidents on passenger car vehicle type between 2008 and 2017 are due to failure of driver's right of way. As per 2013 figures, 800 people were died due to fatigue, asleep, sickness of drivers [1]. In the event that the driver neglected to focus on driving for some time, it diminishes driver response time and impedes directing conduct.

One approach to recognize driver sleepiness is to watch the driver while driving. The monitoring is done through a webcam. If the driver is not concentrating on driving, this system needs to alert the driver with alarm sound. Coming to the cause when a driver misses traffic signs, this system need to detect traffic sign and display the correct instruction. So this developed system that identify traffic signs can be a valuable assistance to the greater part of the drivers. The final goal is not only to detect the extreme and visible cases of drowsiness but to detect even softer signals of drowsiness.

In this project we have modules, namely Automatic detection and recognition of the traffic signs and Drowsiness Detection.

To prevent these accidents we will construct a system using OpenCV library which will alarm the driver when he feels drowsy and a Deep Neural Network model that can recognize traffic signs present in the image into various classifications through GUI. The entire project has been developed using python programming language and jupyter Integrated Development Environment (IDE).

Jupyter is a scientific python development environment with plugins jupyter Notebook, terminal, unit test and reports. These plugins and components available in jupyter helps to build a dynamic system according to our specifications and with ease one can find bugs in the project through debugging if any.

1.1.1. AUTOMATIC DETECTION AND RECOGNITION OF TRAFFIC SIGNS

Deep learning is a piece of Machine realizing which is fundamentally used to group the information through features. It utilizes numerous layers to separate the features of elevated level from the raw input. Most of the deep learning algorithms are based on the artificial neural networks which has similarity with the human brains. Human brains are dynamic whereas these algorithms are static. Convolution Neural Networks are the most widely used algorithm of deep learning. It has many layers involved in to extract the features, namely Convolution (as the name implies) layer, Max pooling layer, Flattening layer and Fully connected layer. The raw data is sent into the layers to extract the high level features for further classification.

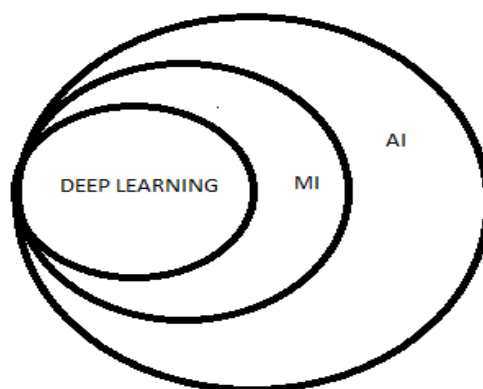


Fig.1.1. Venn diagram of components of Artificial Intelligence (AI)

In automatic traffic sign recognition system we have used a dataset that has 43 different classes of several images [2]. Then we have built a CNN model. In the wake of building the model, engineering we prepared the model with train information which is 80% of our dataset. Further we approved our model with test information which is of 20% of our dataset. The accuracy on validation is around 95%. And finally we can test our model manually with dataset through GUI which displays the image label.

1.1.2 DROWSINESS DETECTION

OpenCV (Open Source Computer Vision Library) is a python library which is for the most part utilized for the PC vision applications. It is mainly used to do operations on images. It is also used to analyze the data from the cameras, i.e., read and write images, detection of faces and its features and for all other image processing applications. Dlib library an open source library is utilized to find the spots of the face from which we can extract the eye landmarks. These libraries need to be imported and deployed into the project.

For driver drowsiness alert we find the Euclidean distance between eyelids. Then we calculate the eye aspect ratio which we use it as a reference to detect the closing of eyes. If the eye aspect ratio is decreasing rapidly, it is a blink. If it didn't increase further, then we count a defined seconds to detect it as drowsiness. As soon as it is been confirmed as drowsiness, an alarm sound will alert the driver. And then back to the normal position once he/she is awake by making the count value zero.

1.2 MOTIVATION

ROAD SAFETY

Road accidents have become an alarming issue now-a-days. According to the global statistics 1.3 million people died to the road accidents on an average of 3,700 deaths per day in a year [3]. To reduce these, factors causing accidents are important. The one of causes was driver inattention. So, we have decided to take our work on this aspect considering drowsiness and detection of traffic signs to help the driver by alerting and assisting. The display system displays the traffic sign detected early on which reduces the failure of seeing an important traffic sign. An alarm is played when the driver feels drowsy; this can be identified by monitoring the driver through a camera. This could reduce the cost of lives due to the road accidents.

NAVIGATION

Another motivation is to navigate when driving in dense urban roads where GPS availability is limited. An exact recognition of the traffic signs prompts the protected goal.

1.3 OBJECTIVE

The main objective of ours is to develop a system which helps the driver by alerting when drowsy and assisting him/her with traffic signs for a safe drive.

1.4 SCOPE OF THE PROJECT

Building a real time system to reduce number of accidents by assisting and alerting the driver since the number of accidents on records are due to the driver inattention and errors.

CHAPTER 2

LITERATURE SURVEY

2.1 AUTOMATIC DETECTION AND RECOGNITION OF TRAFFIC SIGNS

Traffic signs are identified depending on the color information importantly blue and red. Then the shapes are classified as triangular, square, and circular. In light of the color and shape data traffic signs are identified. One of the methods is to detect and classify the image or video taken from the camera placed in the vehicle. Three stages suggested are detection, classification and recognition [4]. In detection stage we concentrate on specific areas known as Region of Interest i.e., we search for the region of traffic sign. The classification stage tests every ROI to characterize it into one of the class. Finally, in recognition the traffic sign is identified. When dealing with videos we additionally have tracking stage for quicker identification of ROI and better sign detection and classification by investigating the data.

For recognition stage fundamental significance is given to shading to discover the areas of traffic signs colour segmentation. Infact, the colour of paints is defined priori. Be that as it may, the color appearance may change contingent upon the time. RGB pictures are generally changed over to another color space for examination. Colour is explored for sign detection. Decoupling the colour and intensity information is allowed, so every input image is changed to HSV color space. Red and blue hues are recognized in the information pictures by breaking down the Hue (H) part. The saturation(S) segment is likewise utilized.

Accepting signs are not all that much pivoted or harmed and realizing that permitted shapes are triangles, squares, octagons or circles, the width of a sign is relied upon to be fundamentally the same as its height. Finally, any area contained inside another locale is disposed of as a plausible mixed up (incomplete) detection. For circle distinguishing proof, every ROI is checked utilizing the fast radial symmetry detection method (FRS) [9]. In the event that a roundabout shape is available, the FRS yield will contain high qualities on the circle's focal zone. In perfect conditions, just the middle pixel of the yield picture would should be tried, yet for genuine pictures all pixel esteems inside a square district, around the yield community, are analyzed. Within this squared focal locale, all pixel esteems are arrived at the midpoint of (avg). That normal is partitioned by the greatest (max) yield esteem, bringing about a circle likelihood. Triangular and squared shapes are recognized by finding the sides of every ROI, utilizing the Harris

corner identification calculation [10]. The presence of corners is then tried in six diverse control areas of the ROI

As the traffic signs have defined shapes in the classification stage, the ROI are detected based on geometric shape properties. For shape recognition edge or corner location techniques are regularly utilized. Cross correlation based template matching with road sign template, genetic algorithm, Haar wavelets have also been used for this purpose [4]. Finally a model which utilizes template matching or trained back propagation neural network is used to analyse the Region of Interest (ROI) permitting the traffic signs to be recognized.. Results dependent on the making of a search window around the past sign transient position are thought of for tracking using Kalman filters. Also, RFID is used to detect the traffic signs [6].

NEURAL NETWORK APPROACH:

This strategy proposes three stages: image preprocessing, localization and classification. During image preprocessing, HSV shading space is utilized to extract blue and red pixels from a picture. The strategy for localization, which is an adjustment of summed up Hough transform, has been created considering the imperatives on schedule for preprocessing a single frame. Classification with artificial neural system is a well known methodology for recognition issues [7-9]. Regularly, neurons are sorted out in layers, and the associations are built up between neurons just from adjoining layers. The yield layer neurons sum is equivalent to the quantity of grouping pictures. The training procedure is to limit the cost work with minimization techniques dependent on gradient descent. The cost function utilized is cross entropy. Training systems with enormous number of layers additionally called deep networks, with RELu function is utilized as activation function.

A convolutional neural system takes a 2D picture and significantly processes it with convolutional layers. Each convolutional layer comprises of a lot of trainable filters and performs dot product between these filters and input layer contributing an activation map. We can utilize the profound learning Tensor stream library. The Gabor wavelet is utilized as the underlying convolutional part between the input layer and the first convolutional layer. In the real street scenes, the difference in light, the harm of traffic signs, and the hindrance of snags will affect the nature of the preparation picture. Regardless, Gabor wavelet can take care of such issues estimably. The Gabor wavelet

can face challenges toward changes in light; thusly, it has great versatility to light. Moreover, it has unrivaled scale and bearing determination attributes that are touchy to the edges of the preparation picture.

After each pooling layer, the BN is included for information standardization. In the deep learning system model, as the quantity of preparing builds, the hidden layer slope close to the yield layer grows and the boundary refreshing quickens. In the mean time, the hidden layer inclination close to the input layer shows the inverse; that is, introducing a condition of arbitrary circulation called gradient dispersion.

The Adam method is chosen as the optimizer algorithm. This method is an extended first-order optimization algorithm based on the stochastic gradient descent method, which can dynamically adjust the learning rate of related parameters by using the moment estimation of the gradient. Training and testing were actualized utilizing dataset. The network architecture developed comprises of a few convolutional layers and softmax layer.

Tensorflow comprises of set of tools to imagine model at various derivation levels. To train and assess the model, the underlying dataset was partitioned into train and test datasets with proportion 4:1 correspondently. After effective training the precision was registered utilizing all pictures from test dataset. This technique gives great acknowledgment results even with traffic sign pictures which are difficult to perceive for a human.

2.2 DROWSINESS DETECTION

IMAGE PROCESSING APPROACH:

The proposed way to deal distinguish driver's sleepiness is at two levels. The application is introduced on driver's gadget running Android operating system (OS). The procedure begins with capturing of live pictures from camera and is in this manner sent at neighborhood server. At the server's side, Dlib library is utilized to identify facial spots and a limit esteem is utilized to recognize whether driver is sleepy or not. These facial spots are then used to figure the EAR (Eye Aspect Ratio) and are returned back to the driver. In our unique circumstance, the EAR esteem got at the application's end would be contrasted and the limit esteem taken as 0.25. If that the EAR esteem is not exactly the limit esteem, at that point this would show a condition of weakness. If there should arise an occurrence of Drowsiness, the driver and the travelers would be cautioned by an alert. The resulting segment subtleties the working of each module.

For utilizing the application, the driver plays out an enlistment if utilizing the application just because. Subsequent to playing out a sign-up, the driver includes a ride by entering the source and goal of the ride. Similarly, an interface for the travelers is likewise furnished where the travelers can associate with the ride, included by the driver. The driver at that point begins the ride. The proposed application at that point catches the continuous pictures of the driver. Pictures are caught each time the application gets a reaction from the server. The procedure goes on until the driver stops the ride. The logs of the outcomes that were caught by the application were gathered and dissected with the assistance of AI classifiers.

To separate the facial tourist spots of drivers, Dlib library was imported and sent in our application. The library utilizes a pre-prepared face identifier, which depends on an adjustment to the histogram of arranged inclinations and utilizations straight SVM (support vector machine) strategy for object identification. Genuine facial milestone indicator was then introduced and facial tourist spots caught by the application were utilized to figure separation between focuses. These separations were utilized to register

EAR esteem. EAR is characterized as the proportion of height and width of the eye. In our sleepiness case, the Eye Aspect Ratio is observed to check if the worth falls beneath limit esteem and furthermore it doesn't increments again over the edge an incentive in the following casing. The above condition infers that the individual has shut his/her eyes and is in a tired state. Despite what might be expected, if the EAR esteem increments once more, it suggests that the individual has recently squinted the eye and there is no instance of drowsiness.

After catching the facial milestone focuses, EAR esteem registered by the server is presently gotten at the android gadget of the driver and contrasted and the edge esteem which was before set to be 0.25. On the off chance that the worth is not exactly the edge, at that point the counter worth is increased, else the counter worth is interfered with to zero. If that the counter worth ranges to three, an alert is activated in the android gadget.

Another variable (Sleep Counter) is kept up which checks the occasions the EAR esteem is not as much as limit esteem. Variable (Sleep Counter) stores the all reactions from the server side and is utilized to compute the ECR (Eye Closure Ratio). It is characterized as the proportion of Sleep Counter and Total Counter. The estimation of ECR was determined for each 15 sequential casings (caught from camera). When the casing number compasses to 16, the estimation of absolute counter gets one and rest counter gets zero. At whatever point the ECR surpasses the limit which is set to 0.5, at that point a notice is produced in the android application to show tired condition of the driver.

CHAPTER 3

METHODOLOGY

3.1 DRIVER ASSISTANCE & ALERTNESS SYSTEM

The project consists of two modules namely,

- Detection and Recognition of Traffic Signs
- Drowsiness Detection

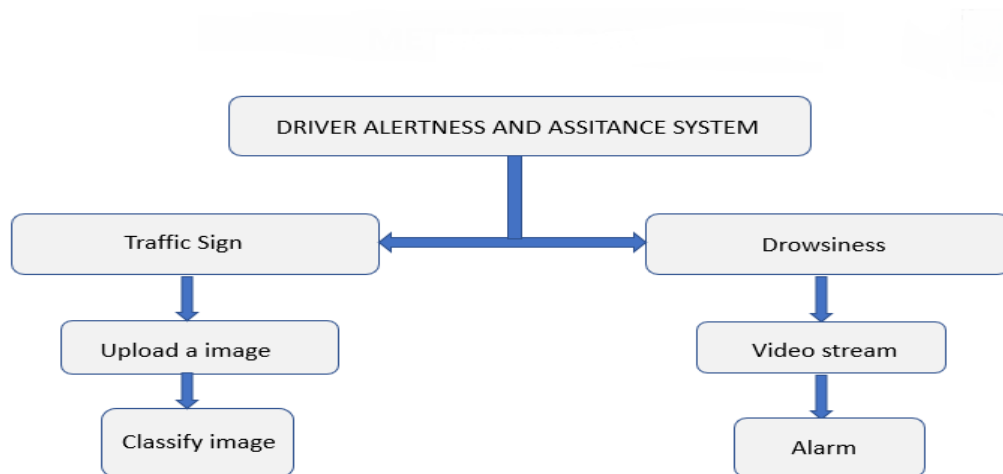


Fig. 3.1. Block diagram of Driver Assistance & Alertness System

The integration of the both modules has been done through a Graphical User Interface using python as shown in fig.3.1. Tkinter library has been used to develop the GUI. Two options traffic signs and drowsiness detection are been made available to the user upon chosen directed to the particular modules for functioning. Spyder IDE is used for the entire project which is been developed using python programming language.

When traffic signs option is opted then the functioning will be directed to the traffic signs module developed to distinguish the traffic sign and perceive the class it has a place with.

When drowsiness detection option is chosen the program is directed to the drowsiness detection module which uses a web cam to capture the streaming video and plays an alarm if the driver is found drowsy.

3.2 DETECTION AND RECOGNITION OF TRAFFIC SIGNS

The detection and recognition of traffic signs uses CNN algorithm to train the model and a brief note about the CNN algorithm is as follows.

3.2.1 CNN ALGORITHM:

Convolutional Neural Network (CNN) algorithm is a profound learning network which is utilized to break down the visual images. It comprises of input and yield layer alongside different hidden layers as shown in fig.3.2. The middle layers of CNN consists of convolutional layers that convolves with dot product. RELU layer is an activation function and is followed by pooling layers, fully connected layers and normalization layers. These three layers are known as hidden layers in view of concealing of their data sources and yields by the activation function and last convolution.

Layers of the CNN algorithm:

- Convolution
- Max Pooling
- Flattening
- Fully Connected Layer

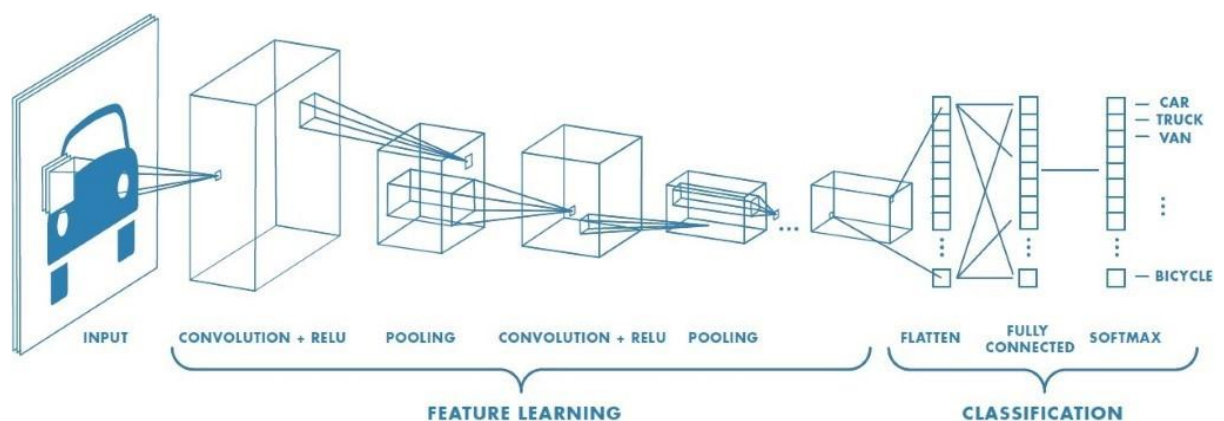


Fig. 3.2.Layers of CNN [17]

(i) CONVOLUTION LAYER:

Convolution layer is the absolute first layer in the algorithm to detect the features. It keeps the connection between utilizing little squares of information pixels by learning image features. It takes two information sources such as image matrix and a filter or kernel and does a basic mathematical operation.

For example an image matrix of 5×5 is been convolved with kernel matrix of 3×3 to get an output of 3×3 matrix and the output is known as the Feature Map. Convolution of image matrix with different filter matrices brings about different filters such as edge detection, blur and sharpen.

Non-Linearity(Relu):

RELU is an activation function stands for Rectified Linear Unit for a non-linear operation. It is to give the non-linearity in our Convolutional Network providing non-negative values. The output of the function $f = \max(0, x)$.

(ii) POOLING LAYER:

At the point when the pictures are huge this layer lessens the parameters. Spatial pooling diminishes the dimensionality of each map by holding the significant data. Spatial pooling as called as Down-Sampling can be of different types:

- Max Pooling
- Average Pooling
- Sum Pooling

Max pooling takes the largest component depending on the filter size from the rectified feature map and places in the output respectively.

According to the filter size, feature map average is taken in average pooling.

Whole of all components in the element map brings about sum pooling

In this project we are using the max pooling layer.

(iii) FLATTENING LAYER:

The target of this layer is to take the yield of max pooling layer as information and convert them into a single vector of values. This 1-dimensional array is fed into the

fully connected layer for the further process. In Fig.3.3 flattening of the 2-dimension vector into 1-dimension is shown. Each value speaks to a likelihood that a specific component has a place with a class separately.

(iii) FULLY CONNECTED LAYER:

In fully connected layer the output from the flattening layer is fed into a neural network. In the below diagram, the single dimensioned array e.g., $(x_1, x_2, x_3, x_4, x_5, \dots)$ is converted to another 1-dimension array of $(y_1, y_2, y_3, y_4, y_5, \dots)$. The fully connected layer is same as the Artificial Neural Network (ANN) and performs calculations such as $g(wx+b)$. Here g is activation function, w is the weight matrix, x is the input matrix which is of 1-dimension and b is the bias matrix. The features are combined in the fully connected layer to create a model to classify the outputs as turn left/right, speed limit etc.,

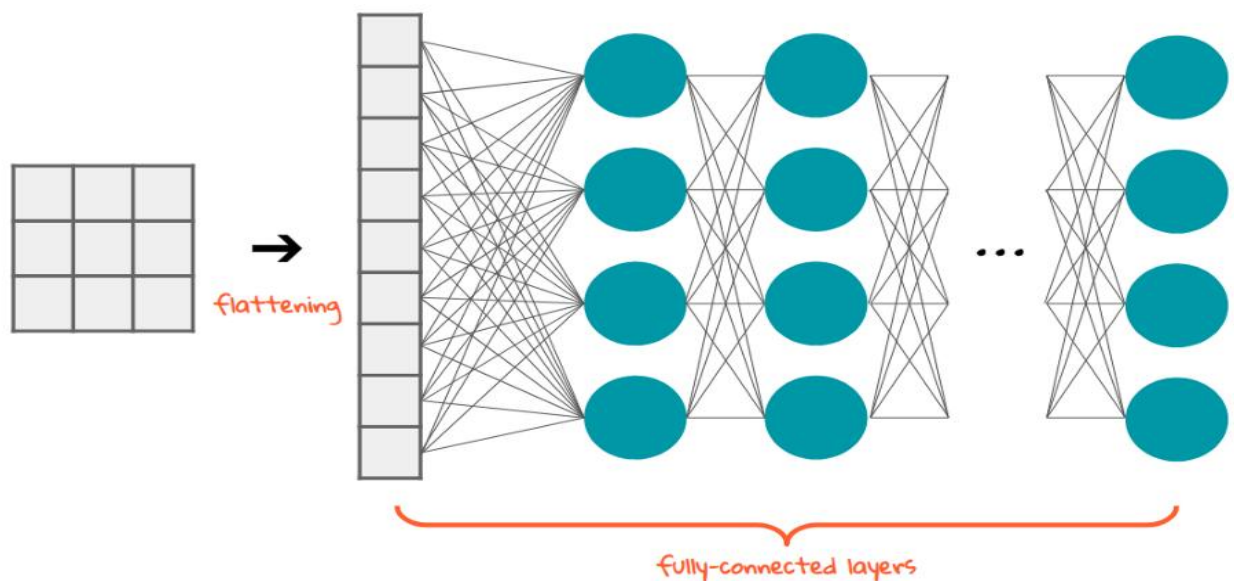


Fig. 3.3.Flattening and Fully Connected Layer [18]

3.2.2 STEPS INVOLVED IN DEVELOPING THE MODEL:

In the process to build a model involves four steps. They are

- Exploring the data set
- Building an CNN model
- Training the model
- Validation of the model
- Testing the model

(i) EXPLORING THE DATASET:

In this project, the dataset used is an open source public dataset which consists of approximately 39,000 images of different traffic signs [3]. The traffic signs are classified into 43 different classes as shown in fig. 3.4. Along with the images in the dataset labels of the images are saved in .csv file for the validation purpose. This particular dataset is varying; few classes have many images whereas few have some images. The dataset has two folders test and train. Test folder has 43 sub folders representing each class with images inside and train folder has images for testing our model.



Fig. 3.4.Images of Traffic Signs

CNN classifier is used to classify an image into appropriate categories accordingly. An info picture is viewed as matrix of pixels relying upon goals by PC resolution. The image is seen as height x width x dimension. For example an image of 6 x 6 x 3 array of matrix where 3 refers to RGB values depending on the image resolution.

The train folder consists of 43 folders of different class. Since, there are 43 different classes the images are stored in folders from 0 to 42. Using OS package available in python, we can emphasize into all the classes and add pictures and their particular labels in the information and labels list.

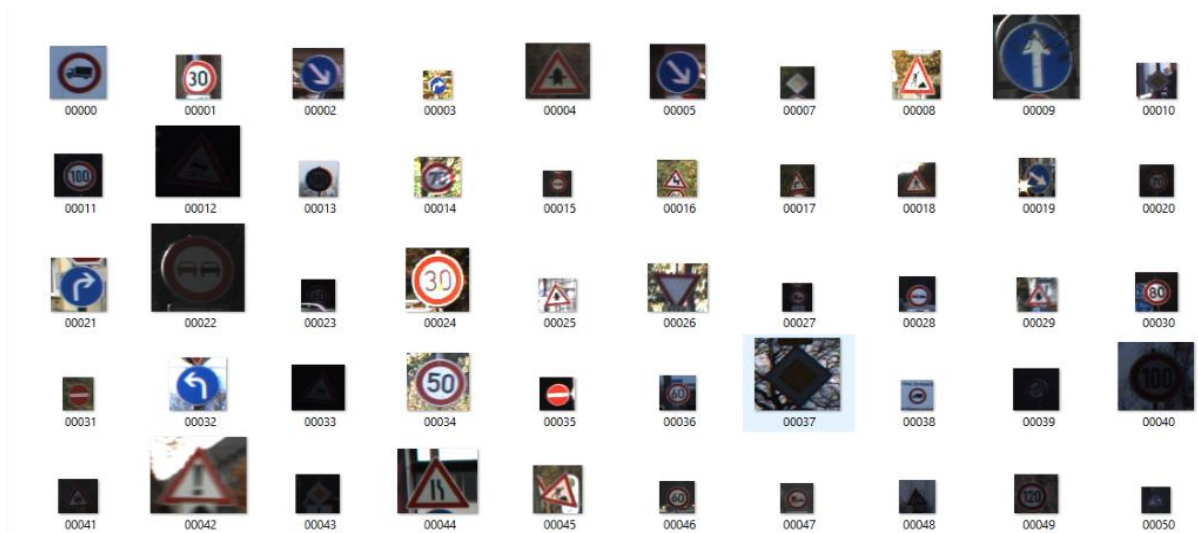


Fig. 3.5. Testing Dataset

To open an image from the train folder into an array PIL (Python Image Library) which is an open source library is used. The required pre-processing in this algorithm is less. The images need to be resized for developing the CNN algorithm. In present undertaking the pictures are resized to 30*30 pixels. The resized images are stored into the lists. Using sklearn package, train_test_split() method is utilized to part training and testing information. Finally the lists are converted into numpy arrays by importing numpy package to feed into the model.

(ii) BUILD A CNN MODEL:

CNN model is built to classify the images to the respective class. For image classification it is the best one. Keras package is used to build the neural network. Conv2D, Maxpool2D, Flatten and Dropout are imported from keras.layers package.

The architecture of the model is

- Conv2D layer
- MaxPool2D layer
- Conv2D layer
- MaxPool2D layer
- Flatten layer
- Fully connected layer

In convolution layer, the features are extracted from the image using kernel matrix. The activation function is applied on the feature matrix to make it non-linear. In Max Pooling layer the maximum values depending on kernel is taken. In flatten layer the 2D matrix is converted into 1D vector matrix. Finally in the fully connected layer, the 1D matrix is used to classify the image into respective class. The model is compiled using adam optimiser.

(iii) TRAINING THE MODEL:

Training an algorithm means making an algorithm to predict an image to which class it belongs to. Once the architecture of the model is built, the model is trained using fit() function. The training dataset consists of 43 folders which contain images of traffic signs respectively in each folder. The model is trained using the 80% of the dataset. With 64 as batch size the algorithm performed better. The trained model has got 95% of accuracy. The trained model has to be saved using save function.

FLOW CHART OF CNN MODEL:

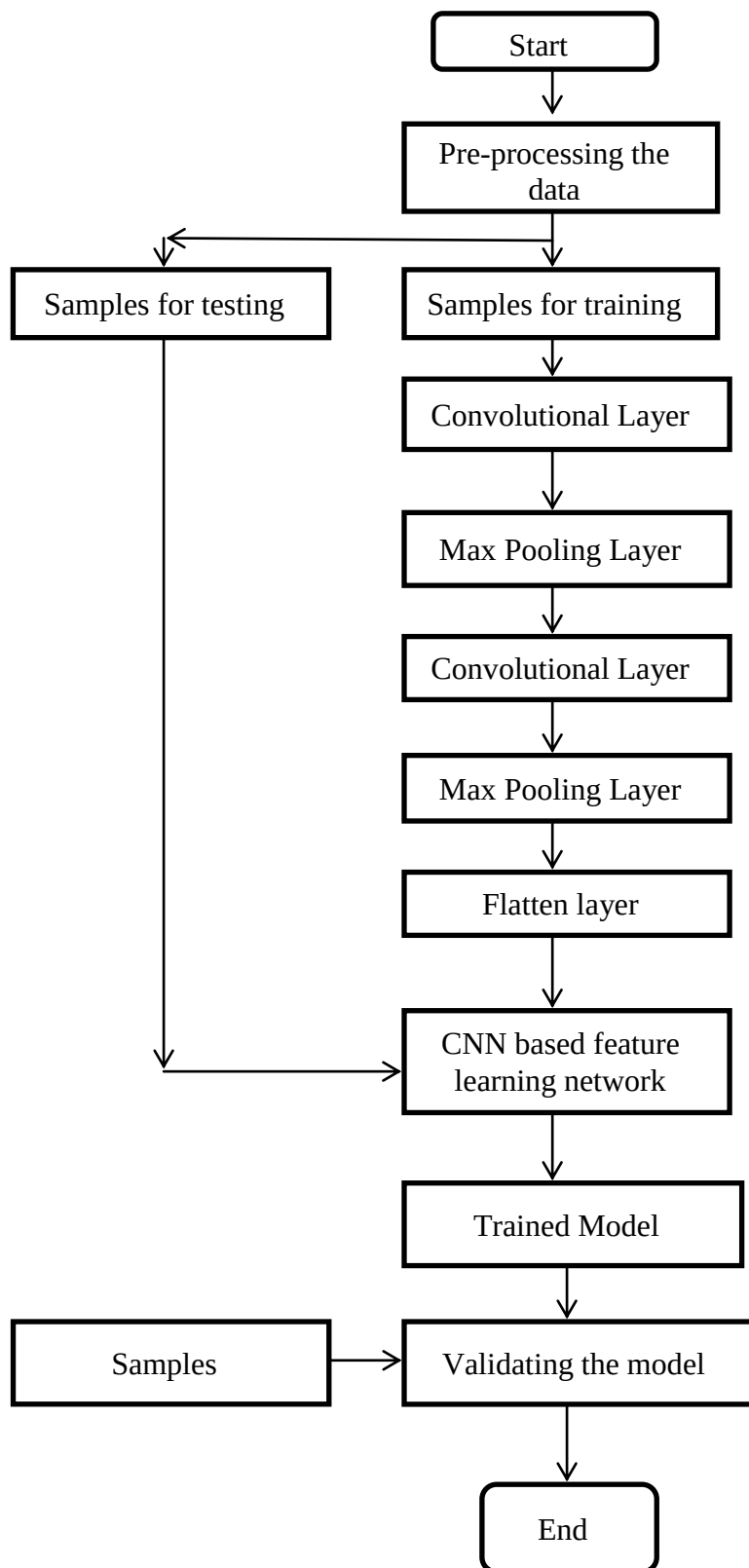


Fig. 3.6.Flow chart of the CNN model

(iv) VALIDATING THE MODEL:

Validation of the model has been done using the 20% of the dataset. The entire 20% of the images are given to the evaluate function which can be invoked by model along with the respective image class names for the validation. This function gives the percentage of accuracy of our model by verifying the class of the image obtained by the model to the values provided by us. For the particular model developed, we have obtained the accuracy of 95%.

(v) TESTING THE MODEL:

To test the model, a graphical user interface has been developed with upload button and classify button using tkinter package. A testing image is given as input and need to be resized into the same dimension as in training i.e., 30 x 30 pixels. The images available in test folder are shown in fig. 3.5. Using predict_classes(image) function the class to which it belong is found. The function returns a number between 1 and 43 since there are 43 different traffic signs and accordingly the name of the class is found from the label list.

3.3 DROWSINESS DETECTION:

Drowsiness Detection is a technology which is developed to decrease the quantity of accidents because of sleepiness of drivers when driving long. This system is developed using libraries such as imutils, dlib and OpenCV in python.

Algorithm:

- To monitor the stream of faces a setup of webcam/camera is needed
- Eye regions are extracted when a face is found by applying facial landmark detection
- Eye aspect ratio is calculated to check if the eyes are closed once we have the eye regions

- If found that eyes have been closed more than threshold amount of time, to wake up the driver we will sound an alarm.

Facial landmarks of an image are detected using OpenCV and dlib libraries. The salient regions of the face are represented and localized using facial landmarks such as eyes, nose and mouth etc.,

OpenCV library is open source software which is used to identify the face from an image or video stream. It has its wide applications in image processing domain and thus we have used this library for the detection of drowsiness.

To appraise the area of 68 coordinates that guide to facial spots on the face, a pre-trained landmark is used which is inside dlib library. So, for our current project a dlib library have been imported and deployed.

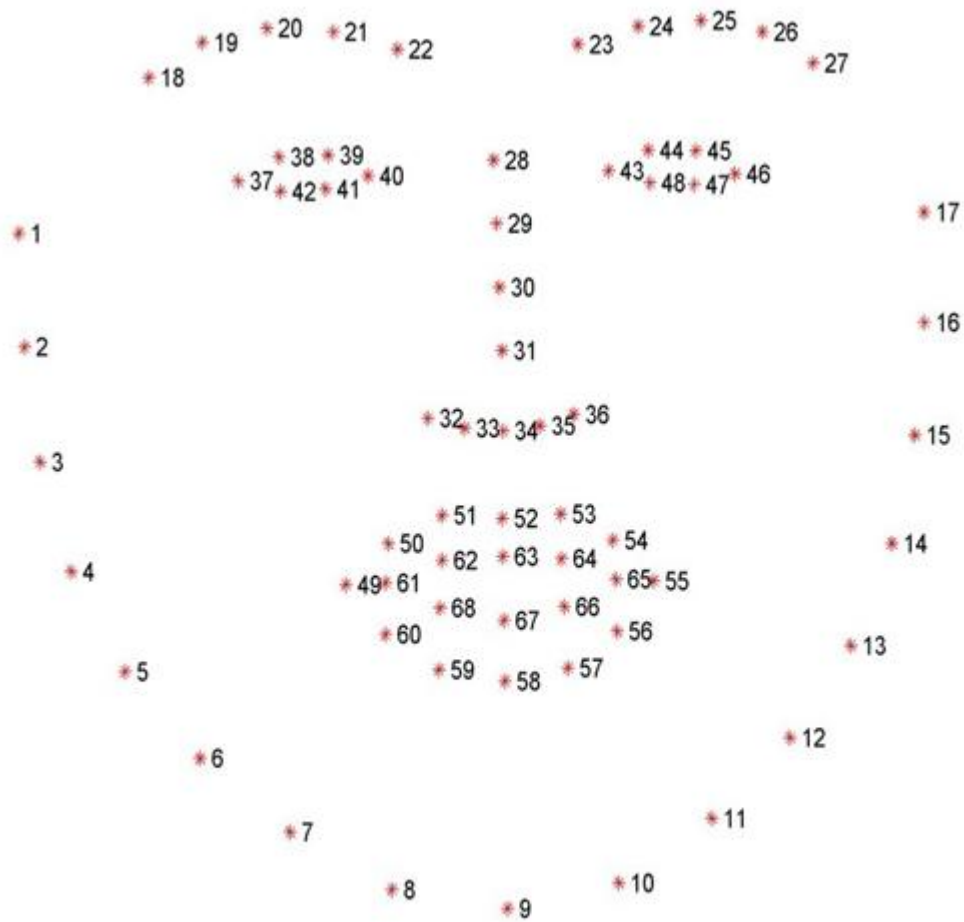


Fig. 3.7. Visualizing the 68 facial landmark coordinates [19]

From the above diagram we can infer that the coordinates (37-40) represents the left eye whereas (43-48) represents the right eye.

Each eye is spoken to by six coordinates which begins at the left-corner of eye and proceeds in clock astute course. Eye perspective proportion (EAR) is consistent when the eye is open and tumbles to zero when the blink is taking place. Using the coordinates of eyes the eye aspect ratio (EAR) for eyes is calculated using Euclidean distance.

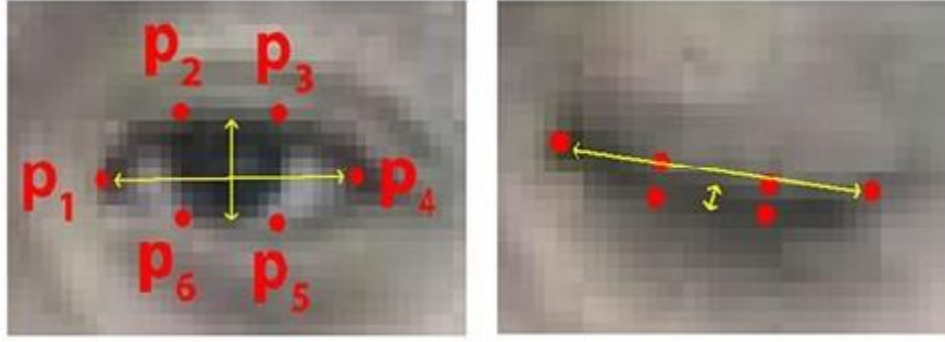


Fig. 3.8.The 6 landmarks of face associated with the eye [20]

The equation which relates the width and height of the coordinates are to calculate the eye aspect ratio (EAR) is given as follows.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

where $p_1, p_2, \dots, p_6$ are landmarks of face which are associated with the eye. The denominator of the condition figures the separation between horizontal eye landmarks while numerator of the condition ascertains the separation between the vertical eye spots.

On the off chance that the eye aspect ratio (EAR) determined is beneath the limit,, the number of frames the person has closed their eyes are counted. If the number of frames the person has closed their eyes exceeds a certain value, the alarm will sound to wake up the person. To play an alarm playsound function is used from playsound library which accepts alarm's path as parameter. When the eyes are open, we will reset counter value and make sure that alarm is off.

3.2 TOOLS AND LIBRARIES USED

➤ **Spyder**

Spyder is an Integrated Development Environment (IDE) with lot number of plugins and components. The presence of the components in the GUI helps the developer by making the environment flexible. The easy way to run projects developed in python are through spyder which is installed as a part of Anaconda distribution.

➤ **Numpy**

This is a python library which is used to work with arrays, basically to store values of same data type in arrays.

➤ **Keras**

Keras is one of the most important packages which is developed upon tensorflow also it is a Neural Network Library. This package consists of the key functions used to build CNN like Conv2D, MaxPool2D, Flatten, and Dropout.

➤ **Sklearn**

It contains most of the tools used in Machine Learning and is one of the most commonly used libraries. In this project this package is used to import `train_test_split`, which is used to part the dataset into two sections, training and testing according to the ratio given.

➤ **Matplotlib**

Matplotlib is a python library which is an extension of the numerically mathematics of Numpy library. This library is used to plot the graphs between validation and training for accuracy and loss.

➤ **OS**

The os module in python is used to interact with the operating system (OS). In this project to train the data the train folder present in our machine directory need to be accessed. So, using the function `getcwd()` of os current working path directory can be obtained. Also to recursively enter into the folders os module is helpful.

➤ **Tkinter**

Tkinter package is used to develop a GUI. The subpackage of this package, `filedialog` is used to open the machine directory to get the input image for the prediction of the image label.

CHAPTER 4

RESULTS AND DISCUSSIONS

RESULTS OF AUTOMATIC DETECTION AND RECOGNITION OF TRAFFIC SIGNS MODULE DURING TRAINING AND VALIDATION

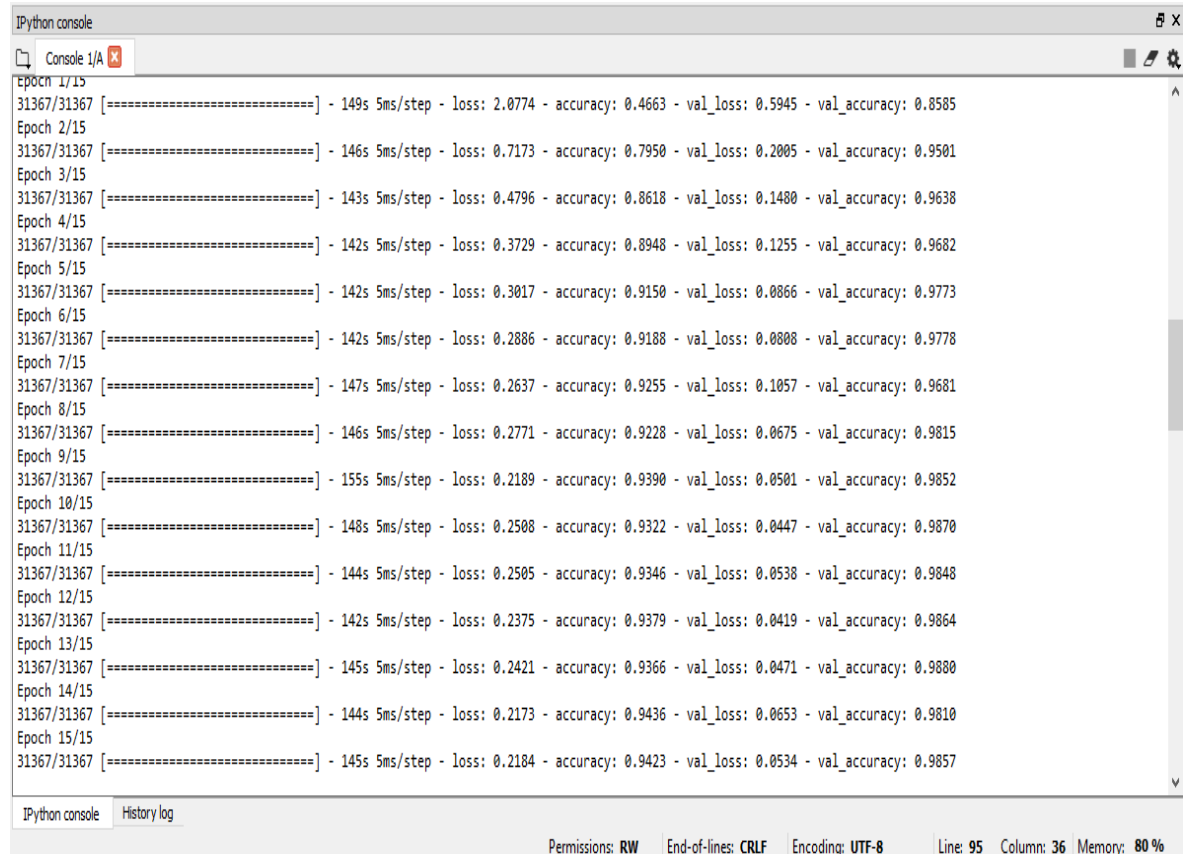


Fig. 4.1. Accuracy and Loss values of the model while training

The model after being passed through the layers of CNN, it is compiled using fit function available in the keras package. The value of epoch we have given is 15 with batch size 64. The number of epochs is the number of times we go through the training dataset. Each time a batch is processed the model is updated which implies that the model is updated multiple number of times during one epoch. The loss and accuracy values of training as well validation is shown in fig. 4.1.

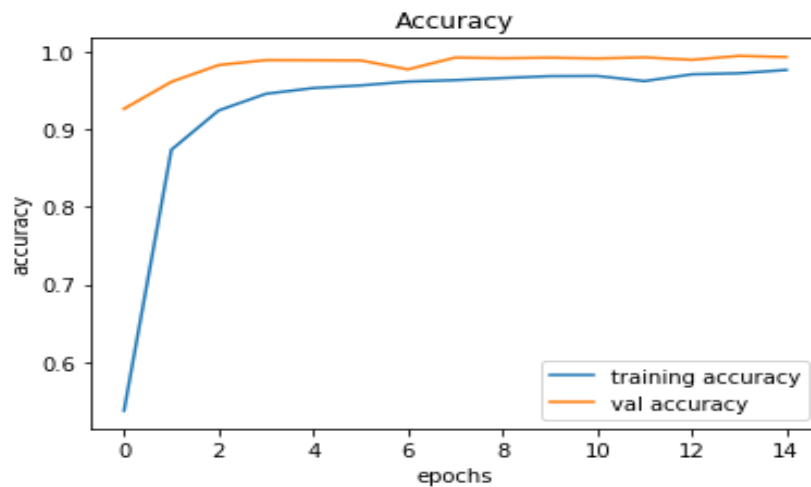


Fig. 4.2. Graph representing the training and validation accuracy

The graph has been plotted between the training accuracy and the validation accuracy by referring the values in fig.4.1. The validation accuracy varies from 0.95 to 1.0 in the fig. 4.2 which represents the model works as expected. Also can observe that the accuracy is stable after 7 epoches. The testing accuracy on the test dataset is 0.96.

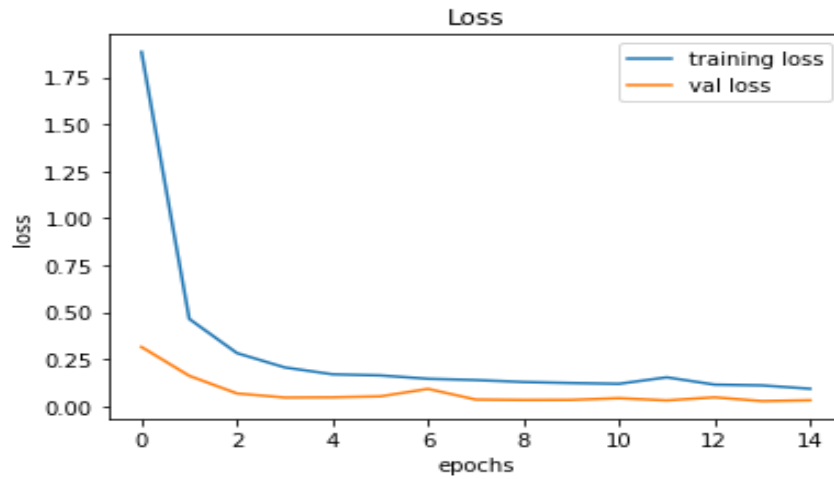


Fig. 4.3. Graph representing the training and validation loss

The graph has been plotted between the training loss and the validation loss. The validation loss can be inferred from fig. 4.3 which is less than 0.07. Also can observe that the accuracy is stable after 7 epoches. The validation loss is 0.041.

RESULTS OF DRIVER ASSISTANCE & ALERTNESS SYSTEM

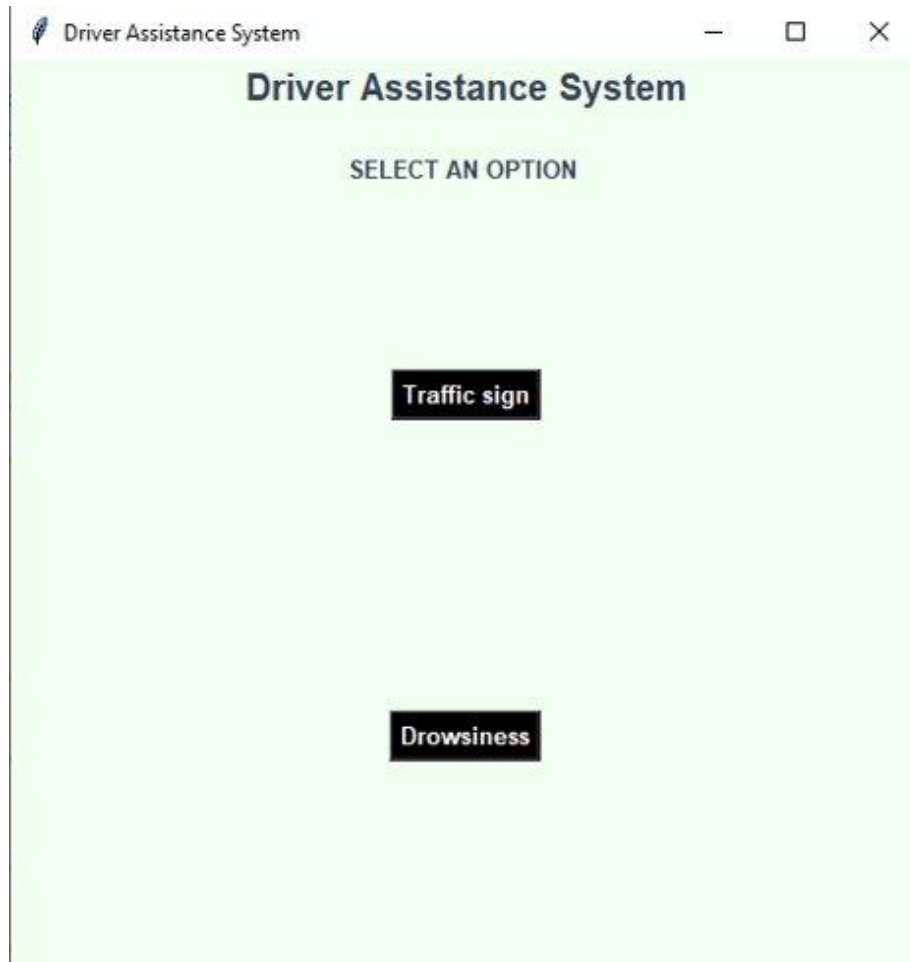


Fig. 4.4.GUI of the Traffic Assistance and Alertness System

This is the output of the system we have developed; upon clicking the run option in spyder IDE right away the page opens as shown in fig.4.4. The entire appearing thing is developed using tkinter package in python. *Traffic signs* and *Drowsiness* are buttons developed with the command specified to the recommended modules. Upon clicking the buttons, it we be redirected to the module it should to be to.

When Traffic signs option is been opted from the main page, the Traffic Sign Detection page opens with *Upload an Image* option, clicking on it takes to the user machine to select an image. After selecting an image, *Classify Image* button appears and upon selecting it displays the class of the respective image.

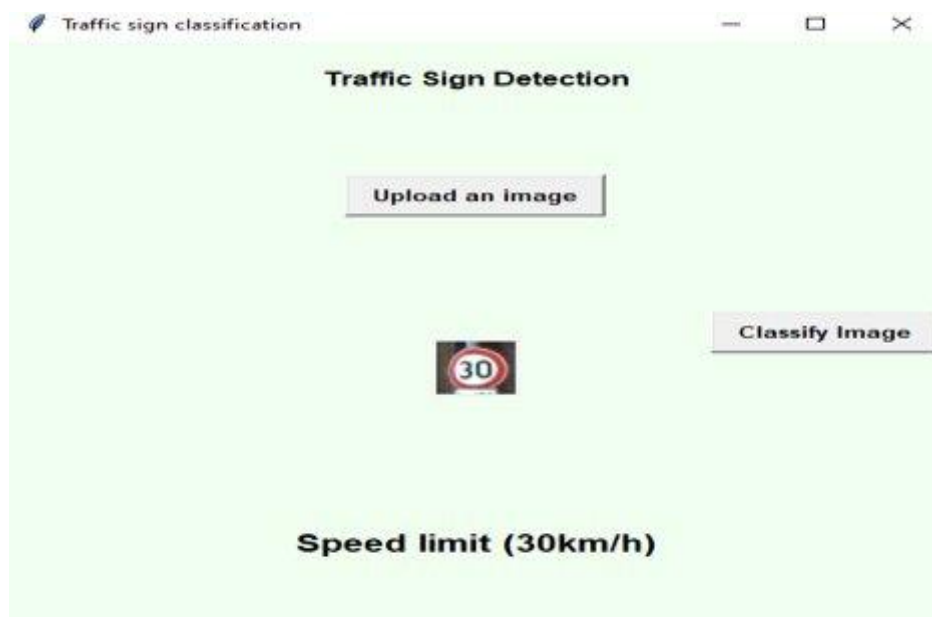


Fig. 4.5.GUI of the Traffic Signs 1



Fig. 4.6.GUI of the Traffic Signs 2

After uploading an input when classify button is clicked, the output i.e., the label of the image is displayed at the bottom of the interface as shown in fig. 4.5 and fig. 4.6.

When Drowsiness option is opted a frame opens to capture the streaming video of the driver to plot the eye landmarks and to detect the drowsiness to play an alarm. The developed model even works for the person wearing glasses.

When the driver closes his/her eyes for more than a threshold value then a drowsiness alert is given i.e., through playing an alarm.

CHAPTER 5

CONCLUSION AND FURTHER WORK

In this project, a system which is used to assist and alert the driver is been developed in python. Firstly, CNN a deep learning algorithm is used to develop the module automatic detection and recognition of traffic signs. The public dataset is been used for training and testing the model. For training the model 80% of the data is taken and remaining for validation of the model. The accuracy of the model is approximately 95% and to be precise it is 96.57% on validation because of the high level extraction of the features of data than the image processing techniques. The manual testing of the model can be done by using GUI developed by giving an input as an image.

Secondly, the Drowsiness detection module has been developed using OpenCV and dlib libraries available in the python. These libraries are used to detect the face from a video stream and to extract the eye landmarks. Form which the Eye Aspect Ratio (EAR) is been calculated using Euclidean distance. If EAR is less than reference value for particular number of frames an alarm is played to help the driver from drowsiness.

When Driver Assistance & Alertness system is implemented in real time it would be a great benefit to drivers from drowsiness and misinterpreting the traffic signs. The entire project has been developed as software, to make it useful in real time it need to improvised according to the real time specifications by integrating with the hardware like microcontrollers. Cameras need to installed to capture the video stream of driver as well the images of the traffic signs.

CONCLUSION AND FURTHER WORK

With this project we will be able to detect the traffic signs that are uploaded in the GUI of automatic traffic sign detection and alert the driver with an inbuilt alarm sound whenever he feels sleepy or drowsiness. When Driver Assistance & Alertness system is implemented in real time it would be a great benefit to drivers from drowsiness and misinterpreting the traffic signs. Here the deep learning methods like CNN is used for classification of traffic signs. The shape predictor is utilized for detecting the landmarks on face to calculate the eye aspect ratio.

For real time implementation of a hardware setup consisting of 360 degree camera and a microcontroller have to be used in the vehicle. The camera is required to take the images of road traffic signs and video capturing of driver during journey. The microcontroller is used to have the model stored in it and to raise an alarm when it detects the sleepy or drowsy driver behavior.

REFERENCES

1. Drowsy Driving: Asleep at the wheel, CDC 24/7 saving lives, Protecting People. Retrieved from <https://www.cdc.gov/features/dsdrowsydriving/index.html>
2. GTSRB – German Traffic Sign Recognition Benchmark, Multi-class, single-image classification challenge. Available at www.kaggle.com
3. Road Traffic Injuries and Deaths—A Global Problem, Centers for Disease Control and Prevention, CDC 24/7 saving lives, Protecting People, Retrieved from <https://www.cdc.gov/injury/features/global-road-safety/index.html>
4. M.Sajjad Hossain; M.Mahumudal Hasan; M.Ameer Ali; Md. Humayun Kabir .Automatic detection and recognition of traffics signs 2010 IEEE Conference on Robotics, Automation and Mechatronics
5. Luo, Y.; Gao, Y.; You, Z.D. Overview research of influence of in-vehicle intelligent terminals on drivers' distraction and driving safety. In Proceedings of the 17th COTA International Conference of Transportation Professionals: Transportation Reform and Change-Equity, Inclusiveness, Sharing, and Innovation (CICTP), Shanghai, China, 7–9 July 2017; pp. 4197–4205.
6. Jingwei Cao 1,2 , Chuanxue Song 1,2, Silun Peng 1,2, Feng Xiao 1,2 and Shixin Song 3,Improved Traffic Sign Detection and Recognition Algorithm for Intelligent Vehicles, Sensors (Basel). 2019 Sep; 19(18): 4021, Published online 2019 Sep 18. doi: 10.3390/s19184021
7. D V Filatov; K V Ignatev; A V Deviatkin and E V Serykh, Application of machine learning methods for traffic signs recognition, published 1 February 2018 under licence by IOP Publishing Ltd , Published 1 February 2018 , Published under licence by IOP Publishing Ltd
8. Alexander Shustanov and Pavel Yakimov, A Method for Traffic Sign Recognition with CNN using GPU, Samara National Research University, Moskovskoye shosse 34, Samara, Russia, In Proceedings of the 14th International Joint Conference on e-

9. Business and Telecommunications (ICETE 2017) - Volume 5: SIGMAP, pages 42-47, ISBN: 978-989-758-260-8

10. Md. Abdul Alim Sheikh ; Alok Kole; Tanmoy Maity, Traffic sign detection and classification using colour feature and neural network, Publisher: IEEE, Published in: 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)

11. Belal Alshaqaqi; Abdullah Salem Baquhaizel; Mohamed El Amine Ouis; Meriem Boumehed; Abdelaziz Ouam, Driver drowsiness detection system, <https://ieeexplore.ieee.org/document/6602353>

12. Alioua N., Amine A., Rziza M., Aboutajdine D. (2011) Driver's Fatigue and Drowsiness Detection to Reduce Traffic Accidents on Road. In: Real P., Diaz- Pernil D., Molina-Abril H., Berciano A., Kropatsch W. (eds) Computer Analysis of Images and Patterns. CAIP 2011.

13. Sukrit Mehta, Sharad Dadhich, Sahil Gumber, Arpita Jadhav Bhatt, Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio, Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur - India, February 26-28, 2019

14. Andreev, S.; Petrov, V.; Huang, K.; Lema, M.A.; Dohler, M. Dense moving fog for intelligent IoT: Key challenges and opportunities. IEEE Commun. Mag.2019, 57, 34–41.

15. I. García, S. Bronte, L. M. Bergasa, J. Almazán, and J. Yebes, “Vision-based drowsiness detector for real driving conditions,” IEEE Intelligent Vehicles Symposium, Proceedings, pp. 618–623, 2012.

16. S. S. Nagargoje and D. S. Shilvant, “Drowsiness Detection System for Car Assisted Driver Using Image Processing,” International Journal of Electrical and Electronics Research ISSN, vol. 3, no. 4, pp. 175–179, 2015.

17. Hrishikesh, B., Mahajan, S., Bhagwat, A., Badiger, T., Bhutkar, D., Dhabe, S., Manikroa, L.: Design of drodeasys (drowsy detection and alarming system), Proceedings of the World Congress on Engineering and Computer Science 2007 WCECS 2007, October 24-26, 2007, San Francisco, USA

18. Sumit Saha, A Comprehensive Guide to Convolutional Networks – the ELI5 way, Dec 15, 2018. Retrieved from <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
19. Jiwon Jeong, The Most Intuitive and Easiest Guide for Convolutional Neural Network, Jan 24, 2019. Retrieved from <https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480>
20. Adrian Rosebrock, Facial landmarks with dlib, OpenCV and Python, April 3 2017. Retrieved from <https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>
21. Eye blink detection with OpenCV, Python, and dlib, Adrian Rosebrock, April 24, 2017. Retrieved from <https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/>