AN EFFICIENT DROWSINESS AND DISTRACTION DETECTOR -MYDRIVE

A Project Report Submitted To

The APJ Abdulkalam Technological University

In partial Fulfilment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
BY

Mr. MOHAMMED FARIS	**VVT18CS007**
Ms. SWATHI P	**VVT18CS014**
Ms.ATHULYA S	**VVT18CS004**
Mr. AKSHAY K	**VVT17CS002**

Under the Guidance of

Dr.KAVITHA S MURUGESAN

(HEAD OF CSE DEPARTMENT)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VEDAVYASA INSTITUTE OF TECHNOLOGY

KARADPARAMBA, MALAPPURAM

MAY 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VEDAVYASA INSTITUTE OF TECHNOLOGY,

KARADPARAMBA

CERTIFICATE

This is to certify that the report entitled '***AN EFFICIENT DROWSINESS AND DISTRACTION DETECTOR -MYDRIVE'*** submitted by **Mr. MOHAMMED FARIS (Reg:VVT18CS007), Ms. SWATHI P (Reg:VVT18CS014), Ms. ATHULYA S (Reg: VVT18CS004), Mr. AKSHAY K (Reg:VVT17CS002)** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer science and engineering is a bonafide record of the project work carried out by her under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal supervisor External supervisor

Head of the Department

DECLARATION

I undersigned hereby declare that the seminar report "***An Efficient Drowsiness And Distraction Detector -Mydrive***", submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Dr.Kavitha S Murugesan.** This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place : Kozhikode

Date :

Student Name :

Signature :

Student Name:

Signature :

Student Name:

Signature :

Student Name:

Signature :

ACKNOWLEDGEMENT

The satisfaction of having achieved the successful completion of a project would simply be incomplete if the words of gratitude to the great minds that helped in its attainment are not expressed. So, we take this opportunity to thank all of them.

We express our heart-felt gratitude to Honorable Principal ***Dr. Sangheethaa Sukumaran*** *and Director* ***Dr. Arun Korath*** *for allowing us to have the maximum use of college facilities to complete this project. We also thank our most respected head of the department* ***Dr. S Kavitha Murugesan*** *for having provided us with the good atmosphere to present the project and giving necessary guidelines. Our Project guide* ***Dr.Kavitha S Murugesan****, Head Of Department of Computer Science And Engineering had been very helpful and we are grateful to them for their kind of support and guidance through every stage of our work and timely advice.*

We thank all of the teachers in the department of Computer Science And Engineering and our dear classmates for their valuable support and encouragement. We would also like to thank our dear parents and friends, whose encouragement and advice helped us a lot during this project. Last but not least is also seeking this opportunity to remember each and every person who helped and encouraged us during this project.

Student Name:

Student Name:

Student Name:

Student Name

TABLE OF CONTENTS

ABSTRACT

Plastic pollution is the widespread of plastic objects and particles for example plastic bottles, bags etc in the Earth's environment that adversely affects wildlife, wildlife habitat, and humans. Although there are municipal authorities working on it, the method used by them is unscientific and old. In order to find a solution for this An Android-based application is being build that can be used to measure information about the plastic waste collection and removal data. This method promote users to separate organic and non-organic waste so that municipal authorities can collect the non-organic waste from houses. Application is expected to collect data on how much plastic waste is generated by houses in a given area. With this data the user understands how much waste is produced by an individual house and thus awareness can be brought, leading the country to a plastic free country. Application consist of simple UI that can be used by any house members to separate and inform municipal authorities about the waste produced in their house letting them know when to collect the waste produced in that area. This Application database can be accessed by municipal authorities with no skill required.

Keywords: Simple UI, Android Based, Plastic free country

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Drowsy driving is a contributing factor in majority of the car accidents occurring across the world. As a result, the most important approach for preventing these accidents is driver sleepiness detection, which can help us to reduce many road accidents. Sleep Related Vehicle Accidents has the biggest part in traffic accidents. 30% of all injuries and fatalities have been caused by drowsiness globally. Annually 20% of all accidents are caused by fatigue and distractions.

1.1. TREND IN PEOPLE'S THINKING

Humans have always invented machines and devised techniques to ease and protect their lives, for mundane activities like traveling to work, or for more interesting purposes like aircraft travel. With the advancement in technology, modes of transportation kept on advancing and our dependency on it started increasing exponentially. It has greatly affected our lives as we know it. Now, we can travel to places at a pace that even our grandparents wouldn't have thought possible. However, there are some rules and codes of conduct for those who drive irrespective of their social status. One of them is staying alert and active while driving. Neglecting our duties towards safer travel has enabled hundreds of thousands of tragedies to get associated with this wonderful invention every year. It may seem like a trivial thing to most folks but following rules and regulations on the road is of utmost importance.

1.2 FACTS

Our current statistics reveal that just in 2015 in India alone, 148,707 people died due to car related accidents. Of these, at least 21 percent were caused due to fatigue causing drivers to make mistakes. This can be a relatively smaller number still, as among the multiple causes that can lead to an accident, the involvement of fatigue as a cause is generally grossly underestimated. The number of accidents because of distracted driver has been increasing since few years. National Highway

Traffic Safety Administrator of United States (NHTSA) reports deaths of 3477 people and injuries to 391000 people in motor vehicle Fatigue, in general, is very difficult to measure or observe unlike alcohol and drugs and also the distractions, we express our feelings like happy ,sad, to our nearest person on our vehicles .these also have key indicators and tests that are available easily. Probably, the best solutions to this problem are awareness about these type of accidents and promoting drivers to detect the distraction and drowsiness when needed. The former is hard and much more expensive to achieve, and the latter is not possible without the former as driving for long hours is very lucrative. Some countries have imposed restrictions on the number of hours a driver can drive at a stretch, but it is still not enough to solve this problem as its implementation is very difficult and costly.

In order to monitor and prevent a destructive outcome from such negligence, many researchers have written research papers on driver drowsiness and distraction detection systems. But at times, some of the points and observations made by the system are not accurate enough. Hence, to provide data and another perspective on the problem at hand, in order to improve their implementations and to further optimize the solution, this study has been done.

1.3. NEED FOR STUDY

The American National Highway Traffic Safety Administration (https://www.nhtsa.gov (accessed on 4 August 2021)) has an estimated 100,000 accidents reported each year mainly due to drowsy driving. This results in more than 1550 deaths, 71,000 injuries, and 12.5 billion dollars of property damage. According to the National Safety Council (https://www.nsc.org (accessed on 4 August 2021)), 13% of drivers admitted to falling asleep behind the wheel at least once a month and 4% of them resulted in accidents. Mergenthaler et al. announced that drowsiness is one of the main causes of traffic accidents in their study [1]. It is estimated that about 10–15% of car accidents are related to lack of sleep. The sleep questionnaire obtained from professional drivers [2] showed that more than 10.8% of drivers are drowsy while driving at least once a month, 7% had caused a traffic accident, and 18% had near-miss accidents due to drowsiness. These alarming statistics point to the need for capable systems for monitoring drowsy drivers to prevent unfortunate traffic accidents that may occur.

In recent years, building intelligent systems for drowsy driver detection has become a necessity to prevent road accidents. Therefore, it requires a lot of research to design robust alert methods to recognize the level of sleepiness while driving. Many studies focused on constructing the smart

alert techniques for intelligent vehicles that can automatically avoid traffic accidents caused by falling asleep, as illustrated in Figure 1. Rateb et al. [3] introduced real-time driver drowsiness detection for an android application using deep neural networks. A minimal network structure was proposed based on facial landmarks to identify drowsy drivers. The method presented a lightweight model and achieved an accuracy of more than 80%. This study focused only on eye facial landmarks without detecting the yawning of the drivers. Moreover, the method was based on a multilayer perceptron classifier with three hidden layers, which is a limitation that leads to low accuracy. Fatigue detection using Raspberry Pi 3 was provided by Akalya et al. [4] by processing driver's faces and eyes images. A Haar cascade classifier was applied to detect the blink duration of the driver, and the eye aspect ratio (EAR) was computed by the Euclidean distance between the eyes. Abouelnaga et al. [5] created a new data set similar to State Farm's data set for distracted driver detection. Authors prepossessed the images by applying skin, face and hand segmentation and proposed the solution using weighted ensemble of five different Convolutional Neural Networks.

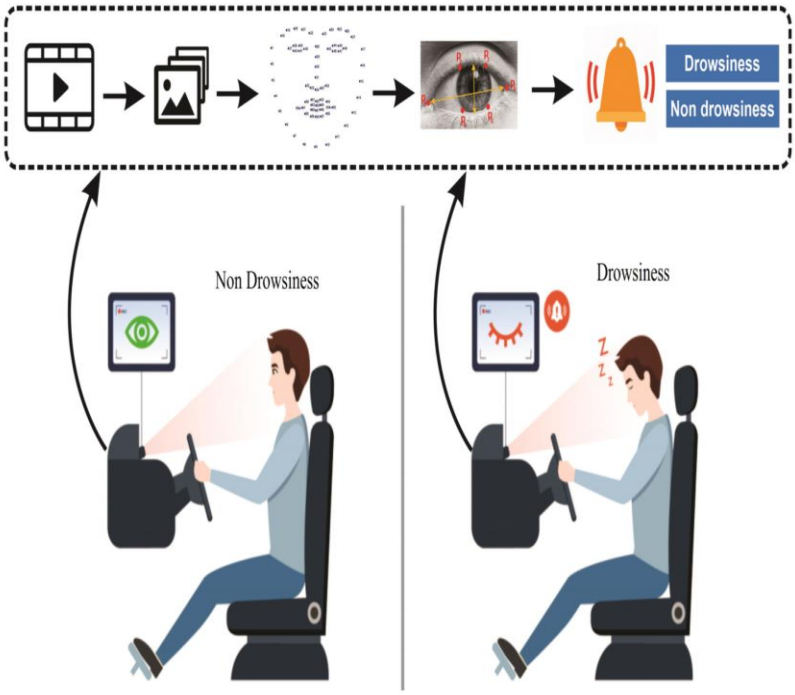

Fig 1.1 General model of the drowsy detection system

1.4. BACKGROUND

1.4.1. DROWSINESS AND DISTRACTION

Sleep is the natural cyclical rest state of the body and mind. In this state, people often close their eyes and lose consciousness partially or completely, thereby reducing their response to external stimuli. Sleep is not an option. It is necessary and inevitable to help the body rest and restore energy. The term “microsleep” or “drowsiness” is defined as brief and involuntary intrusions of sleep that can occur at any time due to fatigue or a prolonged conscious effort. Microsleep can last for a few seconds, and during this time, the brain falls into a rapid and uncontrolled sleep, which can be extremely dangerous, especially in the case of driving or in situations demanding focused attention. There are some signs that show that drivers are not awake: yawning, blinking repeatedly and difficulty opening eyes, the inability to concentrate, the inability to keep the head straight, a distracted mind, feelings of tiredness, and blurred vision.

Distraction involves a diversion of attention from driving, because the driver is temporarily focusing on an object, person, task or event not related to driving which reduces the drivers awareness, decision making ability, and/or performance, leading to an increased risk of corrective actions, near crashes .Distraction could completely cause the driver to lose sight of the road or lose focus of the controls which again could lead to catastrophic failure. some of the examples which includes laughing, talking,makeup ,look left, drinking etc

1.4.2. IDENTIFICATION OF FACIAL LANDMARKS

Kazemi and Sullivan presented a method that precisely estimates the positions of facial landmarks using a training set of labeled facial landmarks on images. This method can be used for real-time detection to identify the facial features after detecting faces on an image. This face features detector identifies 68 main positions (x, y coordinates) on human faces, as shown in Figure 2.

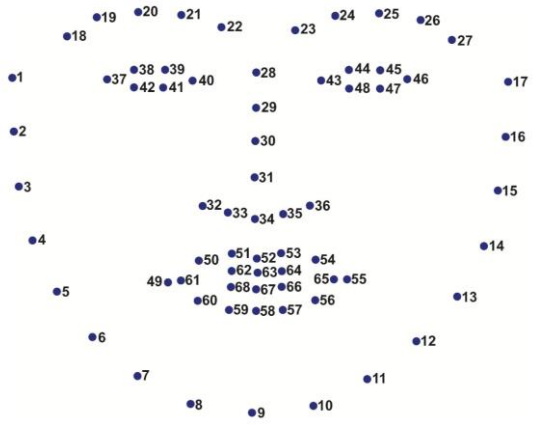

Figure1.2. Illustration of 68 landmarks on a human face

.

1.4.2.1. BLINK DETECTION

To determine the feature positions, we apply facial feature detection including the eyes, eyebrows, nose, ears, and mouth [17,18]. We extract specific facial structures by knowing the index of specific parts of a face. For blink detection, we are interested in eyes features. Each eye is represented by six coordinates, as illustrated in Figure 3, starting at the left corner of the eye and going clockwise around the rest of the area.

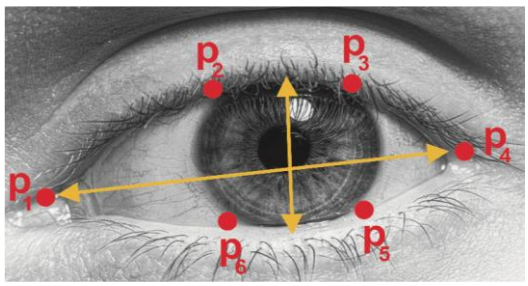

Figure 1.3. Example of 6 facial landmarks related to eyes

1.4.2.2. YAWNING DETECTION

Yawning is characterized by mouth-opening. Like blink detection, facial features are used to detect an open mouth. The mouth-opening value is a parameter used to determine whether a mouth is open. It is calculated by subtracting the mean of all points in the upper mouth with the mean of all points in the lower mouth, as in Equation (2).

Figure 1.4 : Yawning

LIPdistance = jLIPTop □ LIPLowj. (2)

If a mouth-opening value calculated from the frames is higher than a mouth-opening threshold, it is determined to be yawning. An alarm sounds if a person yawns for more than a fixed value threshold. Small loopholes that are caused by talking and eating are ignored.

1.5 DEEP LEARNING

Deep learning is a subset of AI, which is basically a neural organization with at least three layers. Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. It drives numerous computerized reasoning (AI) applications and administrations that further develop mechanization, performing insightful and actual assignments without human mediation. Deep learning innovation lies behind regular items and administrations, (for example, computerized collaborators, voice-empowered TV controllers, and charge card

extortion location) just as arising advancements (like self-driving vehicles). Deep learning research revived and gained unprecedented attention with the availability of “big data” and powerful computational resources in recent years.

The emerging of fast GPUs allows us to train deep models with immense size while the increasingly large data ensures that these models can generalize well. These advantages lead to the tremendous success of deep learning techniques in various research areas and result in immense real-world impactDeep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.

1.5.1 FOUNDATION OF DEEP LEARNING

Machine learning is the research field of allowing computers to learn to act appropriately from sample data without being explicitly programmed. Deep learning is a class of machine learning algorithms that is built upon artificial neural networks. In fact, most of the vital building components of deep learning have existed for decades, while deep learning only gains its popularity in recent years. Deep neural organizations comprise of various layers of interconnected hubs, each expanding upon the past layer to refine and enhance the forecast or order. This movement of calculations through the organization is called forward propagation. The information and result layers of a Deep neural organization are called visible layers. The info/input layer is the place where the profound learning model ingests the information for handling, and the result/output layer is the place where the last expectation or grouping is made. Another cycle called backpropagation utilizes calculations, similar to slope plummet, to work out blunders in expectations and afterward changes the loads and predispositions of the capacity by moving in reverse through the layers with an end goal to prepare the model. Together, forward proliferation and backpropagation permit a neural organization to make forecasts and right for any blunders as needs be. After some time, the calculation turns out to be steadily more precise.

The above depicts the least difficult sort of profound neural organization in the most straightforward terms. Be that as it may, profound learning calculations are amazingly mind boggling, and there are various kinds of neural organizations to resolve explicit issues or datasets.

1.5.1.1 CLASSIC NEURAL NETWORKS

Also known as Fully Connected Neural Networks, it is often identified by its multilayer perceptrons, where the neurons are connected to the continuous layer. It was designed by Fran Rosenblatt, an American psychologist, in 1958. It involves the adaptation of the model into fundamental binary data inputs. There are three functions included in this model: they are:

- **Linear function:** Rightly termed, it represents a single line which multiplies its inputs with a constant multiplier.
- **Non-Linear function:** It is further divided into three subsets:

1. **Sigmoid Curve:** It is a function interpreted as an S-shaped curve with its range from 0 to 1.
2. **Hyperbolic tangent (tanh)** refers to the S-shaped curve having a range of -1 to 1.
3. **Rectified Linear Unit (ReLU):** It is a single-point function that yields 0 when the input value is lesser than the set value and yields the linear multiple if the input is given is higher than the set value.

Works Best in:

1. Any table dataset which has rows and columns formatted in CSV
2. Classification and Regression issues with the input of real values
3. Any model with the highest flexibility, like that of ANNS

1.5.1.2 CONVOLUTIONAL NEURAL NETWORKS (CNN)

CNN is an advanced and high-potential type of the classic artificial neural network model. It is built for tackling higher complexity, preprocessing, and data compilation. It takes reference from the order of arrangement of neurons present in the visual cortex of an animal brain. The CNNs can be considered as one of the most efficiently flexible models for specializing in image as well as non-image data. These have four different organizations:

- It is made up of a single input layer, which generally is a two-dimensional arrangement of neurons for analyzing primary image data, which is similar to that of photo pixels.

- Some CNNs also consist of a single-dimensional output layer of neurons that processes images on their inputs, via the scattered connected convolutional layers.
- The CNNs also have the presence of a third layer known as the sampling layer to limit the number of neurons involved in the corresponding network layers.
- Overall, CNNs have single or multiple connected layers that connect the sampling to output layers.

The CNNs are adequate for tasks, including image recognition, image analyzing, image segmentation, video analysis, and natural language processing. However, there can be other scenarios where CNN networks can prove to be useful like:

- Image datasets containing OCR document analysis
- Any two-dimensional input data which can be further transformed to one-dimensional for quicker analysis
- The model needs to be involved in its architecture to yield output.

1.5.1.2.1 VGG-16 ARCHITECTURE

It is a Convolutional Neural Network (CNN) model proposed by Karen Simonyan and Andrew Zisserman at the University of Oxford. VGG16 is used in many deep learning image classification problems The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is an annual computer vision competition. Each year, teams compete on two tasks. The first is to detect objects within an image coming from 200 classes, which is called object localization. The second is to classify images, each labeled with one of 1000 categories, which is called image classification.
Architecture here input to the network is image of dimensions (224, 224, 3). The first two layers have 64 channels of 3*3 filter size and same padding. Then after a max pool layer of stride (2, 2), two layers which have convolution layers of 256 filter size and filter size (3, 3). This followed by a max pooling layer of stride (2, 2) which is same as previous layer. Then there are 2 convolution layers of filter size (3, 3) and 256 filter. After that there are 2 sets of 3 convolution layer and a max pool layer. Each have 512 filters of (3, 3) size with same padding.This image is then passed to the stack of two convolution layers. In these convolution and max pooling layers, the filters we use is of the size 3*3 instead of 11*11 in AlexNet and 7*7 in ZF-Net. In some of the layers, it also uses 1*1 pixel which is used to manipulate the number of input channels. There is a padding of 1-pixel (same padding) done after each convolution layer to prevent the spatial feature of the image.

1.5.1.3 RECURRENT NEURAL NETWORKS (RNN)

The RNNs were first designed to help predict sequences, for example, the **Long Short-Term Memory (LSTM)** algorithm is known for its multiple functionalities. Such networks work entirely on data sequences of the variable input length. The RNN puts the knowledge gained from its previous state as an input value for the current prediction. Therefore, it can help in achieving short-term memory in a network, leading to the effective management of stock price changes, or other time-based data systems. As mentioned earlier, there are two overall types of RNN designs that help in analyzing problems. They are:

- **LSTMs:** Useful in the prediction of data in time sequences, using memory. It has three gates: Input, Output, and Forget.
- **Gated RNNs:** Also useful in data prediction of time sequences via memory. It has two gates— Update and Reset.

Works Best in:

- **One to One:** A single input connected to a single output, like Image classification.
- **One to many:** A single input linked to output sequences, like Image captioning that includes several words from a single image.
- **Many to One:** Series of inputs generating single output, like Sentiment Analysis.
- **Many to many:** Series of inputs yielding series of outputs, like video classification.

It is also widely used in language translation, conversation modeling, and more.

1.5.1.4 DEEP REINFORCEMENT LEARNING

Before understanding the Deep Reinforcement Learning technique, reinforcement learning refers to the process where an agent interacts with an environment to modify its state. The agent can observe and take actions accordingly, the agent helps a network to reach its objective by interacting with the situation.

Here, in this network model, there is an input layer, output layer, and several hidden multiple layers – where the state of the environment is the input layer itself. The model works on the continuous attempts to predict the future reward of each action taken in the given state of the situation.

Works Best in:

- Board Games like Chess, Poker
- Self-Drive Cars
- Robotics
- Inventory Management
- Financial tasks like asset pricing

1.5.1.5 BACKPROPAGATION

In deep learning, the backpropagation or back-prop technique is referred to as the central mechanism for neural networks to learn about any errors in data prediction. Propagation, on the other hand, refers to the transmission of data in a given direction via a dedicated channel.

The entire system can work according to the signal propagation in the forward direction in the moment of decision, and sends back any data regarding shortcomings in the network, in reverse.

- First, the network analyzes the parameters and decides on the data
- Second, it is weighted out with a loss function
- Third, the identified error gets back-propagated to self-adjust any incorrect parameters

1.6 APPLICATIONS OF DEEP LEARNING

True Deep learning applications are a piece of our regular routines, yet much of the time, they are so all around coordinated into items and administrations that clients know nothing about the mind-boggling information handling that is occurring behind the scenes. A portion of these models incorporate the accompanying:

1.6.1 LAW ENFORCEMENT

Deep learning algorithms can analyze and learn from transactional data to identify dangerous patterns that indicate possible fraudulent or criminal activity. Speech recognition, computer vision, and other deep learning applications can improve the efficiency and effectiveness of investigative analysis by extracting patterns and evidence from sound and video recordings, images, and documents, which helps law enforcement analyze large amounts of data more quickly and accurately.

1.6.2 FINANCIAL SERVICES

Financial institutions regularly use predictive analytics to drive algorithmic trading of stocks, assess business risks for loan approvals, detect fraud, and help manage credit and investment portfolios for clients.

1.6.3 CUSTOMER SERVICE

Many organizations incorporate deep learning technology into their customer service processes. Chatbots—used in a variety of applications, services, and customer service portals—are a straightforward form of AI. Traditional chatbots use natural language and even visual recognition, commonly found in call center-like menus. However, more sophisticated chatbot solutions attempt to determine, through learning, if there are multiple responses to ambiguous questions.

Based on the responses it receives, the chatbot then tries to answer these questions directly or route the conversation to a human user. Virtual assistants like Apple's Siri, Amazon Alexa, or Google Assistant extends the idea of a chatbot by enabling speech recognition functionality. This creates a new method to engage users in a personalized way.

1.6.4 HEALTHCARE

The healthcare industry has benefited greatly from deep learning capabilities ever since the digitization of hospital records and images. Image recognition applications can support medical imaging specialists and radiologists, helping them analyze and assess more images in less time.

1.6.5 AUTOMATED DRIVING

Automotive researchers are using deep learning to automatically detect objects such as stop signs and traffic lights. In addition, deep learning is used to detect pedestrians, which helps decrease accidents.

1.6.6AEROSPACE AND DEFENSE

Deep learning is used to identify objects from satellites that locate areas of interest, and identify safe or unsafe zones for troops.

1.6.7 INDUSTRIAL AUTOMATION

Deep learning is helping to improve worker safety around heavy machinery by automatically detecting when people or objects are within an unsafe distance of machines.

1.6.8 ELECTRONICS

Deep learning is being used in automated hearing and speech translation. For example, home assistance devices that respond to your voice and know your preferences are powered by deep learning applications.

SUMMARY

This chapter gives an overview of Deep learning techniques and their need for avoiding the mishaps that prevailed in the society so far. Also, the scope of Driver drowsiness detection in the earlier stages. Further chapters discuss the related work, existing system, proposed system, conclusion etc.

CHAPTER 2

LITERATURE SURVEY

This section presents some of the most recent research works related to this research and possible solutions suggested by most eminent authors.

2.1 REAL-TIME DRIVER DROWSINESS DETECTION FOR ANDROID APPLICATION USING DEEP NEURAL NETWORKS TECHNIQUES. (JABBAR, R.; Al-KHILAFA), 2018

The American National Highway Traffic Safety Administration (NHTSA). In this paper, a novel approach towards real-time drowsiness detection is proposed. This approach is based on a deep learning method that can be implemented on Android applications with high accuracy. The main contribution of this work is the compression of heavy baseline model to a lightweight model. Moreover, minimal network structure is designed based on facial landmark key point detection to recognize whether the driver is drowsy. The proposed model is able to achieve an accuracy of more than 80%.This method focuses on real-time driver drowsiness detection for an android application using deep neural networks. This study focused only on eye facial landmarks without detecting the yawning of the drivers. Moreover, the method was based on a multilayer perceptron classifier with three hidden layers, which is a limitation that leads to low accuracy.

2.2 DROWSINESS DETECTION BASED ON EYE CLOSURE AND YAWNING DETECTION. (MOHANA AND SHEELA), 2019

This method of drowsiness detection is based on eye closure and yawning detection. They recognized eyes and mouths on faces to detect eye closure and yawning. The limitation of this work is that the eye blink threshold is fixed and the boundary yawn value is 10, consecutively. The algorithm detects drowsiness in a driver using eye closure and yawning as markers. The system is tested on a variety of subjects with limited number of test cases. It accurately detects faces and the required facial features in 85 % of the cases. It is observed that accuracy of detection increases under better illumination conditions. In all the cases of positive feature detection, drowsiness is

promptly detected. Occlusions such as spectacles are no hindrance in eye closure detection. This could be extended to eliminate the difficulties posed by bad lighting conditions

2.3 REAL-TIME DRIVER ALERT SYSTEM USING RASBERRY Pi. (JIE AND LAU) ,2019

This method proposed a vision based real-time driver alert system for monitoring drivers' drowsiness and distraction conditions. However, this study only focused on eye characteristics, ignoring yawns, and the eye-opening threshold was also fixed, leading to limitations on the recognition of people with small eyes. In addition, if the drivers did not continuously stare at the camera for 5 s, the alert threshold was set to 0; thus, it was not possible to detect sleepy drivers.

2.4 A DEEP LEARNING APPROACH TO DETECT DRIVER DROWSINESS (TIBREWAL, M.; SRIVASTAVA), 2021

This method extracted the characteristics of the driver's eyes on each frame using Dlib's API and passed through a classification model to predict the state of drowsiness. Adam was used as the optimizer and the average accuracy reached 94%.

2.5 DETECTION OF DISTRACTED DRIVER BY CONVOLUTIONALNEURAL NETWORK(BHAKTHI BAHETHI),2018

It focus on detecting manual distractions where driver is engaged in other activities than safe driving and also identify the cause of distraction. We present a Convolutional Neural Network based approach for this problem.

We also attempt to reduce the computational complexity and memory requirement while maintaining good accuracy which is desirable in real time applications.

2.6. DRIVER DROWSINESS DETECTION USING DEEP LEARNING. (AJINKYA RAJKAR, N.K.; RAUT), 2021

Introduced a driver drowsiness detection method using deep learning with an average accuracy of 96%. This method included two stages of the pre-processing and drowsiness detection. Haar

feature-based cascade classifiers, a machine learning-based approach, were used to detect the mouth and eye regions of the drivers in pre-processing.

2.7 DRIVER DISTRACTION: A REVIEW OF THE LITERATURE(KRISTIE YOUNG),2019

This paper provides a review of current research on in-vehicle driver distraction, focusing on mobile phone use in particular, given that this device has received the greatest attention in the driver distraction literature.

2.8 DRIVER HAND ACTIVITY ANALYSIS IN NATURALISTIC DRIVING STUDIES: CHALLENGES, ALGORITHMS AND EXPERIMENTAL STUDIES

He had proposed a fusion of classifiers where the image is to be segmented into three regions: wheel, gear and instrument panel to infer actual activity. They also presented a region-based classification approach to detect presence of hands in certain per-defined regions in an image

SUMMARY

The major contribution of this chapter is the study and comparison of related papers. Various techniques related to Deep learning and Driver's drowsiness detection is also described.

CHAPTER 3

DROWSY DRIVER DETECTION SYSTEM

3.1 Drowsy Driver Detection System

A Drowsy Driver Detection System has been developed, using a non-intrusive machine vision based concepts. The system uses a small monochrome security camera that points directly towards the driver's face and monitors the driver's eyes in order to detect fatigue. In such a case when fatigue is detected, a warning signal is issued to alert the driver. This report describes how to find the eyes, and also how to determine if the eyes are open or closed. The algorithm developed is unique to any currently published papers, which was a primary objective of the project. The system deals with using information obtained for the binary version of the image to find the edges of the face, which narrows the area of where the eyes may exist. Once the face area is found, the eyes are found by computing the horizontal averages in the area. Taking into account the knowledge that eye regions in the face present great intensity changes, the eyes are located by finding the significant intensity changes in the face. Once the eyes are located, measuring the distances between the intensity changes in the eye area determine whether the eyes are open or closed. A large distance corresponds to eye closure. If the eyes are found closed for 5 consecutive frames, the system draws the conclusion that the driver is falling asleep and issues a warning signal. The system is also able to detect when the eyes cannot be found, and works under reasonable lighting conditions.

3.2 Concept Design

As seen in the various references, there are several different algorithms and methods for eye tracking, and monitoring. Most of them in some way relate to features of the eye (typically reflections from the eye) within a video image of the driver.

The original aim of this project was to use the retinal reflection (only) as a means to finding the eyes on the face, and then using the absence of this reflection as a way of detecting when the eyes are closed. It was then found that this method might not be the best method of monitoring the eyes fort two reasons. First, in lower lighting conditions, the amount of retinal reflection decreases; and second, if the person has small eyes the reflection may not show, as seen below in Figure

Figure 3.1: Case where no retinal reflection present.

As the project progressed, the basis of the horizontal intensity changes idea from paper was used. One similarity among all faces is that eyebrows are significantly different from the skin in intensity, and that the next significant change in intensity, in the y-direction, is the eyes. This facial characteristic is the centre of finding the eyes on the face, which will allow the system to monitor the eyes and detect long periods of eye closure. Each of the following sections describes the design of the drowsy driver detection system

3.3 System Configuration

3.3.1 Background and Ambient Light

Because the eye tracking system is based on intensity changes on the face, it is crucial that the background does not contain any object with strong intensity changes. Highly reflectiveobject behind the driver, can be picked up by the camera, and be consequently mistakenas the eyes. Since this design is a prototype, a controlled lighting area was set up for testing.

Low surrounding light (ambient light) is also important, since the only significant light lluminating the face should come from the drowsy driver system. If there is a lot of ambient light, the effect of the light source diminishes. The testing area included a black background, and low ambient light (in this case, the ceiling light was physically high, and hence had low illumination). This setup is somewhat realistic since inside a vehicle, there is no direct light, and the background is fairly uniform.

3.2.2.2 Camera

The drowsy driver detection system consists of a CCD camera that takes images of the driver's face. This type of drowsiness detection system is based on the use of image processing technology that will be able to accommodate individual driver differences. The camera is placed in front of the

driver, approximately 30 cm away from the face. The camera must be positioned such that the following criteria are met:

- The driver's face takes up the majority of the image.
- The driver's face is approximately in the centre of the image.

The facial image data is in 480x640 pixel format and is stored as an array through theedefined Picolo driver functions (as described in a later section).

3.2.2.3 Light Source

For conditions when ambient light is poor (night time), a light source must be present to compensate. Initially, the construction of an infrared light source using infrared LED was going to be implemented. It was later found that at least 50 LEDs would be needed so create a source that would be able to illuminate the entire face. To cut down cost, a simple desk blight was used. Using the desk light alone could not work, since the bright light is blinding if 18 looked at directly, and could not be used to illuminate the face. However, light from light bulbs and even daylight all contain infrared light; using this fact, it was decided that if an infrared filter was placed over the desk lamp, this would protect the eyes from a strong and distracting light and provide strong enough light to illuminate the face. A wideband infrared filter was placed over the desk lamp, and provides an excellent method of illuminating the face. The spectral plot of the filter is shown in Appendix A.

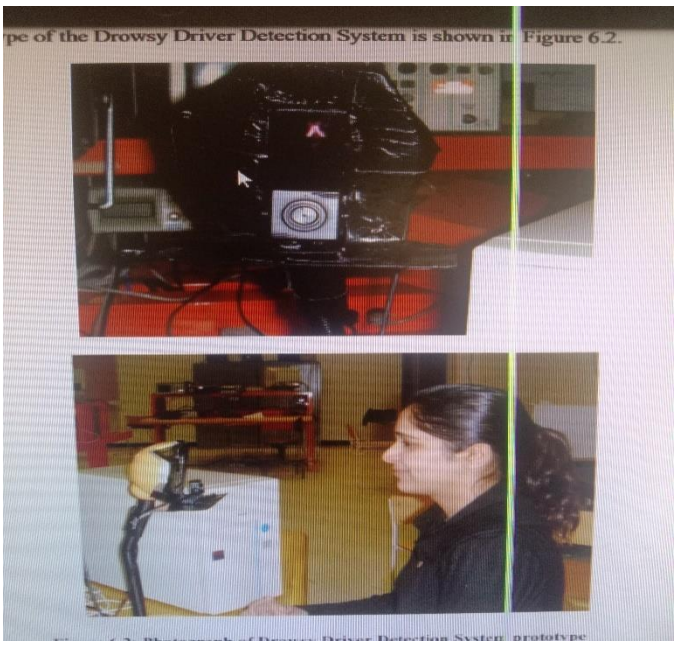

Figure 3.2: Photograph of Drowsy Driver Detection System prototype

3.4 Limitations

With 80% accuracy, it is obvious that there are limitations to the system. The most significant limitation is that it will not work with people who have very dark skin. This is apparent, since the core of the algorithm behind the system is based on binarization. For dark skinned people, binarization doesn't work. Another limitation is that there cannot be any reflective objects behind the driver. The more uniform the background is, the more robust the system becomes. For testing purposing, a black sheet was put up behind the subject to eliminate this problem. For testing, rapid head movement was not allowed. This may be acceptable, since it can be equivalent to simulating a tired driver. For small head movements, the system rarely loses track of the eyes. When the head is turned too much sideways there were some false alarms. 37 The system has problems when the person is wearing eyeglasses. Localizing the eyes is not a problem, but determining whether the eyes are opened or closed is.

CHAPTER 4

AN EFFICIENT DROWSINESS AND

DISTRACTION DETECTOR -MYDRIVE

4.1 Driver Drowsiness Detection

Driver drowsiness detection is a car safety technology which helps prevent accidents caused by the driver getting drowsy. Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads.

4.2 Driver's Distraction

Something that distracts : an object that directs one's attention away from something else turned off her phone to limit distractions One created a distraction while the other grabbed the money. especially : amusement a harmless distraction a book of word puzzles and other distractions.The term driver distraction has been widely discussed and studied, implying that people understand what driver distraction actually means (Regan, Lee, et al., 2008). However, as a scientific concept, driver distraction has been inconsistently defined.The lack of an agreed definition is problematic because it can make interstudy comparisons difficult and can lead to vastly different estimates of the role of distraction in crashes and critical incidents (Gordon, 2008).

Figure 4.1:Class Diagram For Distraction

In this work, we propose two efficient methods for doze alert systems. The first method is based on a combination of two blink and yawn features (EAR, LIP) using facial landmarks. However, adaptive thresholds of blinks and yawns were calculated appropriately for each driver without needing to pre-determine for all drivers. The second method uses advanced deep learning techniques with the transfer learning approach. Facial images are detected with the SSD- ResNet-10, and then forwarded to the proposed networks to detect drowsiness. We designed and perfected two adaptive deep neural networks developed on the advanced networks of MobileNet-V2 and ResNet-50V2 by making improvements in some layers to adapt the drowsiness detection. In addition, we applied the transfer learning approach to train the proposed networks in order to achieve faster learning, no requirement for large training datasets, and an improvement in classification accuracy. The details of the implementation steps of the two proposed methods are presented below.

4.3. Method : Based on Facial Landmarks

The proposed method is inspired by the methods introduced in preceding studies.However, the threshold of eye-opening is fixed for all drivers, leading to inaccuracies for drivers with large or small eyes. To overcome this problem, we improve the determination of the adaptive eye-opening threshold (EARThresh) for each driver. The model of the proposed method is shown in Figure 6. The values of the EAR and EARThresh are first computed for each driver. Then, we determine the drowsiness level of drivers by a comparison of EAR and EARThresh. This work is carried out repeatedly during driving. This method does not rely on the yawning frequency, since it would be inaccurate in cases where drivers wear a mask or talk while driving. The details of this method are described as follows.

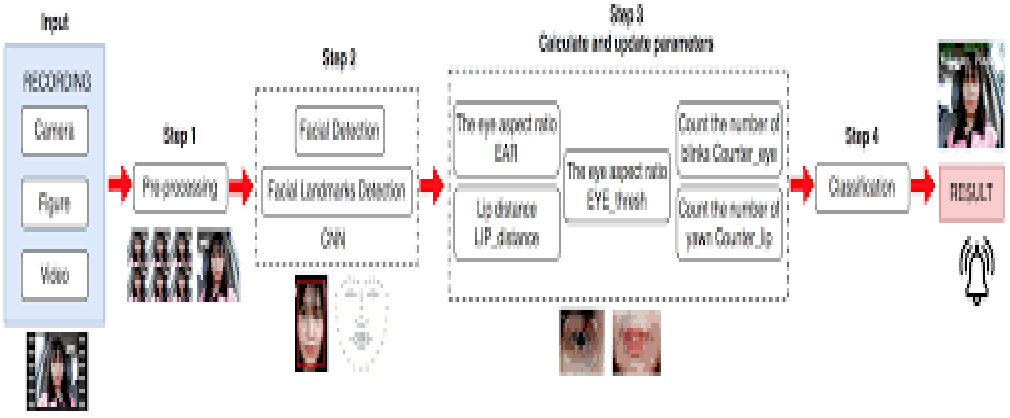

Figure 4.2 Model of the proposed method 1 for detecting drowsiness using facial landmarks.

4.3.1. Step 1: Pre-Processing

In this step, we extract video frames from input videos. The rate of selecting images from videos is 25 frames per second. These images are flipped to accurately identify thefacial landmarks.

4.3.2. Step 2: Detection of Facial Landmarks

We detect and identify facial landmarks on faces in the images. The facial landmarks considered in this work include the eyebrows, eyes, nose, mouth, and jaw

4.3.3. Step 3: Determination of the Eye-Opening Threshold for Each Driver

In this step, we use 32 feature points based on the identification of facial landmarks in step 2. We determine the coordinates of points on the mouth and eyes.

4.3.4. Step 4: Drowsiness Detection and Prediction

The input data are a video tracked from a camera on a vehicle. This video is split into clips with lengths of 300 s. Through the experiments, we tested many different values before reaching the recommended optimal value for CounterEyeLimit = 15 and CounterYawnLimit =25.

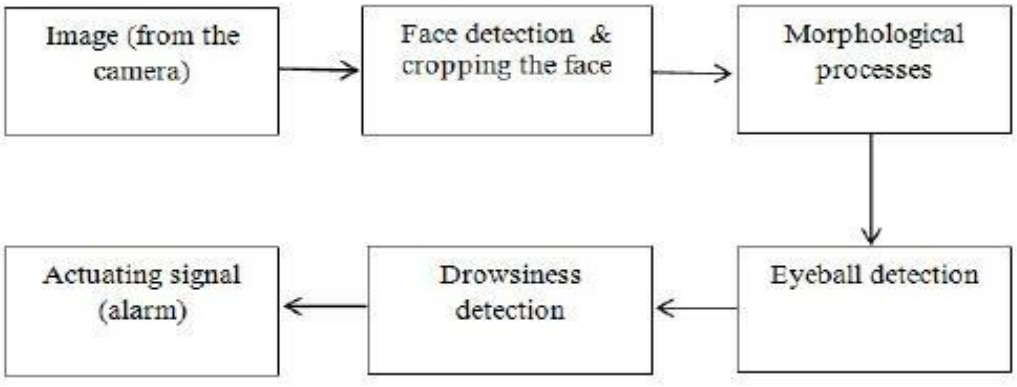

Figure 4.3: Block Diagram Of Driver Drowsiness System

4.4 DATA FLOW DIAGRAM

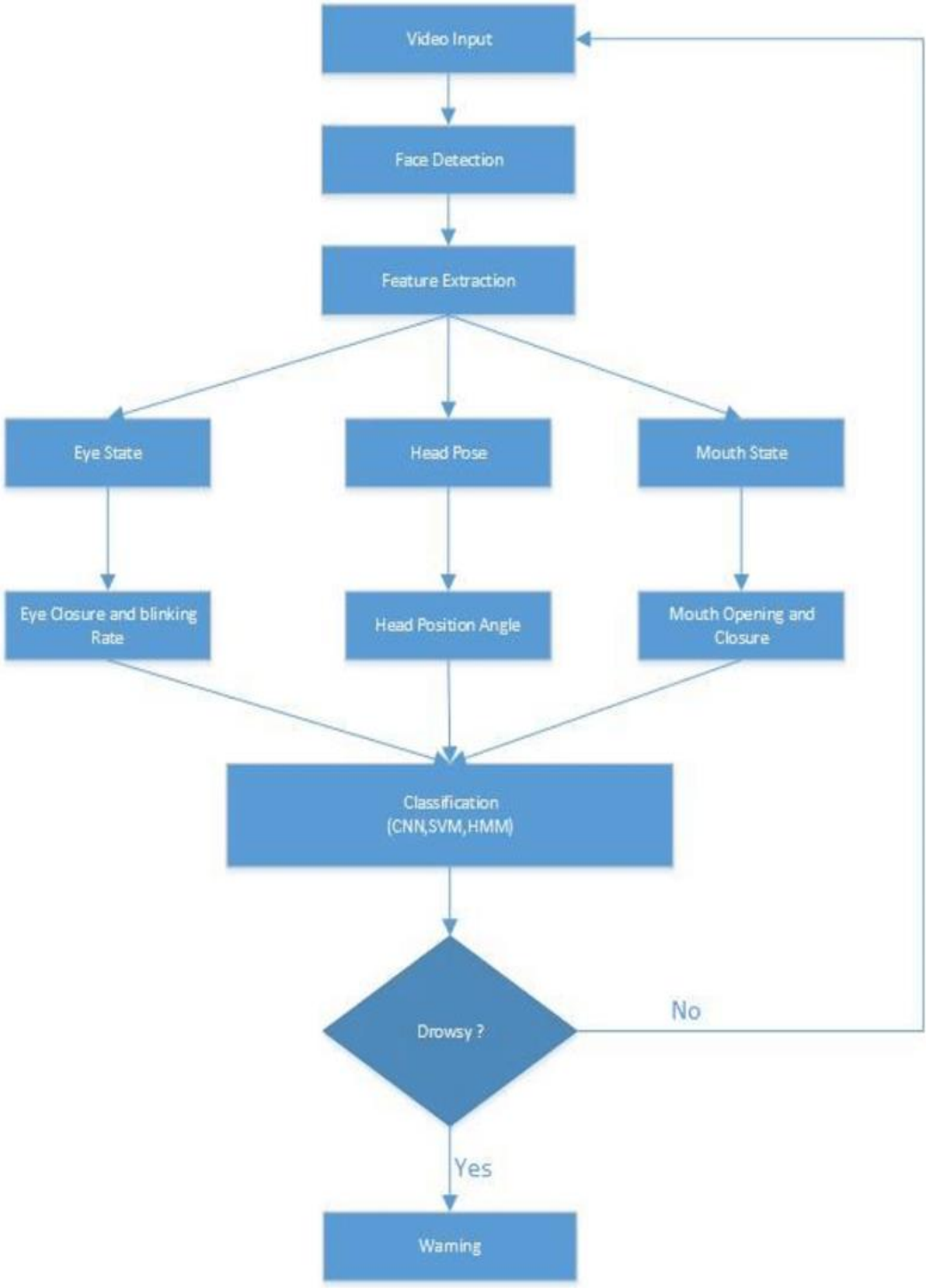

Figure 4.4: Data Flow Graph

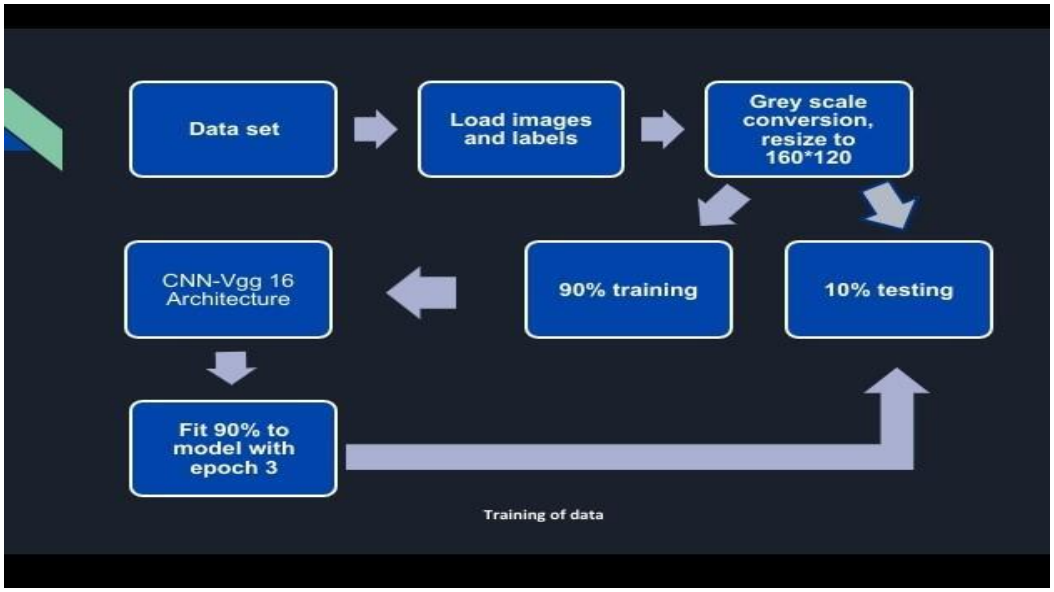

Figure 4.5 : Data Graph For Training Of Data

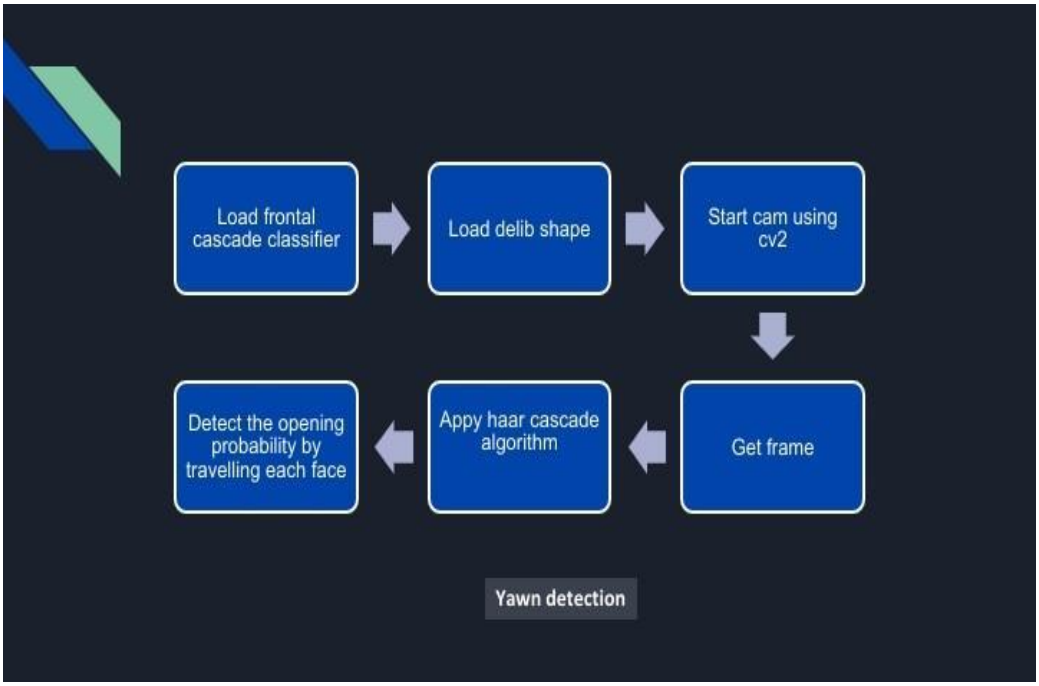

Figure 4.6 : Data Graph for Yawn Detection

CHAPTER 5

IMPLEMENTATION ENVIRONMENT

The requirements for an effective drowsy driver detection system are as follows:

3.1 Hardware Requirements

- **Automotive head unit**

Figure 5.1 :Automotive Head Unit

The head unit provides all the information about the interface and component

- **Webcam**

Figure 5.2: Webcam

it notices the face of the driver and detect it whether the driver is drowsy or distracted by checking all facial expressions

- **Buzzer/speaker**

Figure 5.3: Buzzer/Speaker

when it detected that the driver is drowsy or distracted it produce a sound untill the driver is safe .It ensure the safety of the driver .

3.2 Software Requirements

3.2.1 Python

- **Python 3**

Figure 5.4: Python3

it is the newest vesion of the python programming language.**Python 3** is a newer version of the Python programming language which was released in December 2008. This version was mainly

released to fix problems that exist in Python 2. The nature of these changes is such that Python 3 was incompatible with Python 2. It is **backward incompatible**.

Some features of Python 3 have been back ported to Python 2.x versions to make the migration process easy in Python 3. As a result, for any organization who was using Python 2.x version, migrating their project to 3.x needed lots of changes. These changes not only relate to projects and applications but also all the libraries that form part of the Python ecosystem.

- **Pycharm**

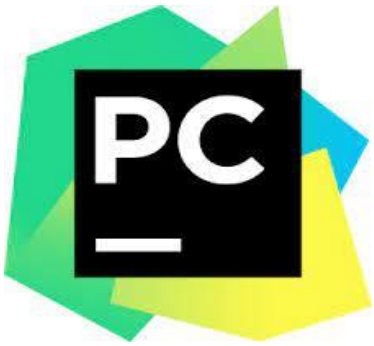

Figure 5.5: Pycharm

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. **PyCharm** is an integrated development environment (IDE) used in computer programming, specifically for the Python programming language. It is developed by the Czech company JetBrains (formerly known as IntelliJ).[5] It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.[6]

PyCharm is cross-platform, with Windows, macOS and Linux versions.

Features:

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages

- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask [professional edition only][8]
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development [professional edition only]
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge
- Support for scientific tools like Matplotlib, NumPy and SciPy [professional edition only]

3.2.2 Packages

- **Tensorflow 1.15.0**：it is free and an open source library. TensorFlow provides a collection of workflows to develop and train models. Here keras Use it as Backend.TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy.If you need more flexibility, eager execution allows for immediate iteration and intuitive debugging. For large ML training tasks, use the Distribution Strategy API for distributed training on different hardware configurations without changing the model definition.TensorFlow has always provided a direct path to production. Whether it's on servers, edge devices, or the web, TensorFlow lets you train and deploy your model easily, no matter what language or platform you use.

  Use TensorFlow Extended (TFX) if you need a full production ML pipeline. For running inference on mobile and edge devices, use TensorFlow Lite. Train and deploy models in JavaScript environments using TensorFlow.js.

- **Keras 2.2.4**：To build our classification model.Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation.

  Keras is:

  -Simple -- but not simplistic. Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter.

-Flexible -- Keras adopts the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what you've already learned.

- **Powerful** -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.

- **OpenCV**：OpenCV is the huge open-source library used for many purposes it plays a major role in real-time operation which is very important in today's systemshere we use open cv for Face and Eye detection.OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

  The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

- **delib 19.7**：Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. Here we are using dlib for Face detection and facial landmark identification

3.3 DATA SET

Firstly the dataset used in this project is the mrlFace dataset available on Kaggle. This dataset consists of 10,000 images of several people's faces with open and closed eyes. The first problem we came across was that this dataset was more of a directory with a dump of images in one place. There was no labelling or order to it. This dataset needed to be arranged and labelled properly and we needed only images of the eyes so this needed to be extracted. For this we used Open CV Haar-

Cascade classifier. This is a feature based cascade classifier. Using its inbuilt functions to detect face and from it to detect eyes region, we ran this process over all the 10,000 images in the dataset and generated eyes dataset

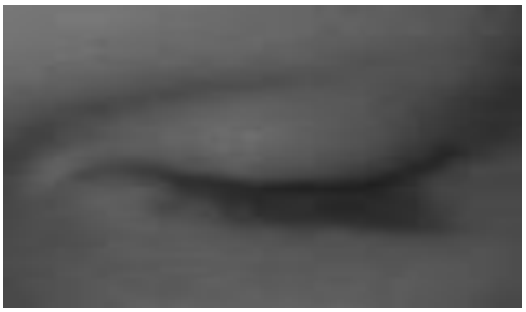

Figure 5.6 :Eye Dataset

Above diagram shows about eye dataset.After this the eye pictures were segregated into labelled folders as open and closed appropriately. This concludedthe dataset acquisition step. Next we began working on the CNN classification model. As mentioned earlier CNN or Convolutional Neural Network is a classification algorithm that works best for categorical data/images. It identifies and does the feature extraction work on its own by using pixel edge detectionmethod, saving us the work to code extraction of individual features. CNN generally consists of 3 layers, Convolution layer, pooling layer and fully connected layer. Thisstructure provides us with a basic blueprint where we only have to choose certain activation functions, values for inbuilt variables, etc. depending upon the desired output.

We used a sequential model with Relu (Rectified Linear Unit) activation. We then converted the 3 channel images (RGB images) into grayscale images with 32 featuredetector. In the pooling layer we are using MaxPooling to reduce the Convolution Matrix. We then add 3 hidden layers with 32, 32 and 64 neurons respectively withMaxPooling. These values and number of layers were agreed upon after several trial and error runs. Next we use the dropout function to reduce the chances of over fitting.Now we flatten the output matrix and use dense layer to create fully connected layer. Sigmoid function along with Softmax activation is used to output values in binary

- **Original VGG-16 architecture**

VGG Net is one of the most influential CNN architecture from literature. It reinforced the idea that networks should be deep and simple. The architecture is shown in fig. 2. It worked well on both image classification as well as local- ization task. VGG uses 3 × 3 filters in all thirteen

convolutional layers, ReLU activation function, 2 × 2 max pooling with stride 2 and categorical cross-entropy loss function. We use the pre-trained ImageNet model weights for initialisation and then fine tune all the layers of network with our dataset. As a preprocessing step, all the images are resized to 224 × 224 and per channel mean of RGB planes is subtracted from each pixel of the image. This has the geometric interpretation of centering the cloud of data aroundthe origin along each dimension. Initial layers of the CNN act as feature extractor and the last layer is softmax classi- fier which classifies the images into one of the predefined categories. However the original model has 1000 output channels corresponding to 1000 object classes of ImageNet.

Hence the last layer is popped and is replaced with softmax layer with 10 classes. Here, the cross entropy loss function is used for performance evaluation.

Figure 5.7: VGG-16 with Regularization From experimentation using original VGG-16 network

It was observed that model is over fitting to the training data. It performs well on the training set achieving almost 100 % accuracy but fails to generalise on the unknown test data. Hence we perform various regularization techniques to re duce the generalization error. Also, LeakyReLU activation function is used instead of ReLU.

CHAPTER 6

RESULT & ANALYSIS

In this,The classification model, runs independently without the knowledge of the user. It performs all the tasks such as face detection and from it the eye region detection and gives it as input to the classification model and gets output. In this way we have achieved Abstraction.

- **Economic Feasibility**: The cost of the project depends upon the training of the model and system requirements for that purpose such as CPU RAM, GPU and disk space. A computer with about 8 GB ram and basic GPU card was used in unison with shared CPU and GPU processing offered by Google Co lab. The disk space offered by the same was also used. The same system was used for the GUI and Android App construction. Hence, by cost-benefit analysis we can conclude that benefit to cost ratio is high.

- **Technical Feasibility**: As mentioned earlier, the task of coding and debugging was made easier by using CNN for the model. As only Smartphone sensors such as camera and speakers are required and much GUI is not required, the app construction coding was also not a very difficult task. As every segment will be coded individually and separately, the risk assessment for each segment is easy.

- **Operational Feasibility**: Efficiency of the project is based upon few factors, namely, the model and its individual accuracy, the front end and its latency, the inter-connectivity. All these components can again be controlled individually and hence accuracy of the project as a whole can be easily controlled.

Fig: screenshot of performance

CHAPTER 7

CONCLUSION AND FUTURE WORK

Most of the traditional methods for drowsiness detection are based on behavioral factors, while some require expensive sensors and devices to measure sleepiness, and may even interfere with the driving process, distracting drivers. Therefore, in this paper, we propose two methods with three scenarios for driver's drowsiness detection systems.

The proposed method with scenario 1 uses facial landmarks to detect drowsiness. This method analyzes the videos and detects drivers' faces in every frame using image processing techniques. Facial landmarks are determined in order to compute the eye aspect ratio (EAR) and the mouth-opening value (LIPdistance) to detect drowsiness based on adaptive thresholds. We propose an improvement of the eye-opening threshold for each driver without using a pre-defined threshold for everyone, as in preceding studies. In each video frame, the frequent detection of eye blinking and yawning will help to properly compute the drowsiness level. The driver is alerted when the blink and yawn thresholds reach the adaptive maximum thresholds.

However, the drowsiness detection based on blinking and yawning is not accurate enough, since drowsiness has different surrounding factors. Therefore, we propose method 2, with two scenarios for a drowsy alert system using deep learning techniques with the transfer learning approach.

REFERENCES

[1] Davies, E.R. “Machine Vision: theory, algorithms, and practicalities”, *Academic Press:* San Diego, 1997.

[2] Dirt Cheap Frame Grabber (DCFG) documentation, file dcfg.tar.z available from http://cis.nmclites.edu/ftp/electronics/cookbook/video/

[3] Eriksson, M and Papanikolopoulos, N.P. “Eye-tracking for Detection of Driver Fatigue”, *IEEE Intelligent Transport System Proceedings* (1997), pp 314-319.

[4] Gonzalez, Rafel C. and Woods, Richard E. “Digital Image Processing”, *Prentice Hall:* Upper Saddle River, N.J., 2002.

[5] Grace R., et al. “A Drowsy Driver Detection System for Heavy Vehicles”, *Digital Avionic Systems Conference, Proceedings, 17th DASC. The AIAA/IEEE/SAE, I36/1-I36/8* (1998) vol. 2.

[6] Perez, Claudio A. et al. “Face and Eye Tracking Algorithm Based on Digital Image Processing”, *IEEE System, Man and Cybernetics 2001 Conference,* vol. 2 (2001), pp 1178-1188.

[7] Singh, Sarbjit and Papanikolopoulos, N.P. “Monitoring Driver Fatigue Using Facial Analysis Techniques”, *IEEE Intelligent Transport System Proceedings* (1999), pp 314-318.

[8] Ueno H., Kanda, M. and Tsukino, M. “Development of Drowsiness Detection System”,*IEEE Vehicle Navigation and Information Systems Conference Proceedings,* (1994), ppA1-3,15-20.

[9] Weirwille, W.W. (1994). “Overview of Research on Driver Drowsiness Definition and Driver Drowsiness Detection,” *14th International Technical Conference on Enhanced Safety of Vehicles,* pp 23-26.

APPENDIX 1

```
input_shape = (120, 140)
num_classes = 10
cnn = CNN(num_classes=num_classes, learning_rate=LEARNING_RATE,
input_shape=input_shape, model='simple')
cnn.load_model()
test_imgs = load_test_data(num_examples=20)
from playsound import playsound



print("-> Starting Video Stream")
# vs = VideoStream(src=args["webcam"]).start()
#vs= VideoStream(usePiCamera=True).start()       //For Raspberry Pi
time.sleep(1.0)


for img in test_imgs:
        pred, lab = cnn.predict_single(img)



        print(pred)
        print(lab)

        state = label_dict[lab[0]]

        if state!="Safe":
                try:
                        playsound("beepas.mp3")
                except:
                        pass

        cv2.rectangle(img, (40, 52), (320, 8), (255,255,255), cv2.FILLED)
        cv2.rectangle(img, (40, 52), (320, 8), (0,0,0), 3)
```

```
        cv2.putText(img, str(state), (42, 40), cv2.FONT_HERSHEY_SIMPLEX, 1.15, (0, 0, 255), 2)
        # plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        # plt.show()

        cv2.imshow("aaaaa",img)
        cv2.waitKey(3000)



from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
import os

def alarm(msg):
    global alarm_status
    global alarm_status2
    global saying

def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

    C = dist.euclidean(eye[0], eye[3])

    ear = (A + B) / (2.0 * C)

    return ear

def final_ear(shape):
    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
```

```
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

    leftEye = shape[lStart:lEnd]
    rightEye = shape[rStart:rEnd]

    leftEAR = eye_aspect_ratio(leftEye)
    rightEAR = eye_aspect_ratio(rightEye)

    ear = (leftEAR + rightEAR) / 2.0
    return (ear, leftEye, rightEye)

def lip_distance(shape):
    top_lip = shape[50:53]
    top_lip = np.concatenate((top_lip, shape[61:64]))

    low_lip = shape[56:59]
    low_lip = np.concatenate((low_lip, shape[65:68]))

    top_mean = np.mean(top_lip, axis=0)
    low_mean = np.mean(low_lip, axis=0)

    distance = abs(top_mean[1] - low_mean[1])
    return distance


ap = argparse.ArgumentParser()
ap.add_argument("-w", "--webcam", type=int, default=0,
            help="index of webcam on system")
args = vars(ap.parse_args())

EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 20
alarm_status = False
alarm_status2 = False
```

```
saying = False
COUNTER = 0

print("-> Loading the predictor and detector...")
#detector = dlib.get_frontal_face_detector()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")    #Faster but less accurate
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')


print("-> Starting Video Stream")
vs = VideoStream(src=args["webcam"]).start()
#vs= VideoStream(usePiCamera=True).start()       //For Raspberry Pi
time.sleep(1.0)

while True:

   frame = vs.read()
   frame = imutils.resize(frame, width=450)
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

   #rects = detector(gray, 0)
   rects = detector.detectMultiScale(gray, scaleFactor=1.1,
                minNeighbors=5, minSize=(30, 30),
                flags=cv2.CASCADE_SCALE_IMAGE)

   #for rect in rects:
   for (x, y, w, h) in rects:
      rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))

      shape = predictor(gray, rect)
      shape = face_utils.shape_to_np(shape)

      eye = final_ear(shape)
      ear = eye[0]
```

```
    leftEye = eye [1]
    rightEye = eye[2]

    distance = lip_distance(shape)

    leftEyeHull = cv2.convexHull(leftEye)
    rightEyeHull = cv2.convexHull(rightEye)
    cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
    cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

    lip = shape[48:60]
    cv2.drawContours(frame, [lip], -1, (0, 255, 0), 1)

    if ear < EYE_AR_THRESH:
       COUNTER += 1

       if COUNTER >= EYE_AR_CONSEC_FRAMES:
          if alarm_status == False:
             alarm_status = True
             t = Thread(target=alarm, args=('wake up sir',))
             t.deamon = True
             t.start()

          try:
             from playsound import playsound
             playsound("beepas.mp3")
             cv2.putText(frame, "DROWSINESS ALERT!", (10,
30),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
          except:
             pass

    else:
       COUNTER = 0
       alarm_status = False
```

```
    if (distance > YAWN_THRESH):
        cv2.putText(frame, "Yawn Alert", (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

        from playsound import playsound

        playsound("beepas.mp3")

        if alarm_status2 == False and saying == False:
            alarm_status2 = True
            t = Thread(target=alarm, args=('take some fresh air sir',))
            t.deamon = True
            t.start()
    else:
        alarm_status2 = False

    cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
                cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
    cv2.putText(frame, "YAWN: {:.2f}".format(distance), (300, 60),
                cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


  cv2.imshow("Frame", frame)
  key = cv2.waitKey(1) & 0xFF

  if key == ord("q"):
      break

cv2.destroyAllWindows()
vs.stop()
```