DRIVER DROWSINESS ALERT SYSTEM USING DEEP LEARNING

A PROJECT REPORT

Submitted by

VENKATA SAI PRAKASH Y (211418104307)

GATTU ASHISH (211418104060)

YELURI HARISH (211418104314)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MAY 2022

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"DRIVER DROWSINESS ALERT SYSTEM USING DEEP LEARNING"** is the bonafide work of "**VENKATA SAI PRAKASH Y (211418104307),GATTU ASHISH (211418104060),YELURI HARISH (211418104314)"** who carried out the project work undermy supervision.

.

SIGNATURE

Dr.S.MURUGAVALLI M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE

NAZARATHPETTAI,

POONAMALLEE,

CHENNAI-600 123.

SIGNATURE

Mr.C.THYAGARAJAN, M.E.,Ph.D.,

ASSISTANT PROFESSOR

DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE

NAZARATHPETTAI,

POONAMALLEE,

CHENNAI-600 123.

Certified that the above mentioned students were examined in the End Semester project viva-voce held on_____.

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION BY THE STUDENT

We **VENKATA SAI PRAKASH Y (211418104307), GATTU ASHISH (211418104060), YELURI HARISH (211418104314)** hereby declare that this project report titled **"DRIVER DROWSINESS DETECTION USING DEEP LEARNING",** under the guidance of **Mr.C.THYAGARAJAN, M.E.,Ph.D**., is the original work done by us and we have not plagiarized or submittedto any other degree in any university by us.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D**. and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Coordinator Dr.N.PUGHAZENDI M.E.,Ph.D.,** and **Project Guide Mr.C.THYAGARAJAN M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

VENKATA SAI PRAKASH Y

GATTU ASHISH

YELURI HARISH

ABSTRACT

Driver's inattention might be the result of a lack of alertness when driving due to driver drowsiness and distraction.Driver distraction occurs when an object or event draws a person's attention away from the driving task.Unlike driver distraction, driver drowsiness involves no triggering event but, instead, is characterized by a progressive withdrawal of attention from the road and traffic demands.Both driver drowsiness and distraction, however, might have the same effects, that is decreased driving performance, longer reaction time, and an increased risk of crash involvement. All over the world Drowsiness has been the significant cause of horrible accidents which is causing deaths and fatalities injuries. Day by Day fatal injuries numbers are increasing globally. From the past many years, researchers have concluded drivers with a lack of sleep and more tiredness which causes drowsiness of the driver. this paper shows a new experimental model is designed for detecting drowsiness of driver is presented to reduce accidents caused by this problem which increases transport safety. To achieve this, two ways are used to detect the drowsiness of a person effectively. First Driver face is captured and eye retina detection and facial feature extraction are done and blinking values are calculated then threshold values are set, which is a old traditional way which uses SVM algorithm for detection of face and eyes and complex classification algorithms are used to classify the frames of the real time video. Secondly With the simple binary classification algorithm the frames will be classified as closed eye or open eye by the deep learning model, and then the system acts accordingly. If he/she closes their eyes for a fraction of time or more than blinking time the system identifies the drowsiness and it warns or alerts them with an alarm and prevents accidents from happening

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

CV- Computer Vision

CNN - Convolutional Neural Network

INTRODUCTION

1. INTRODUCTION

Deep Learning has found huge applications in the fields of Computer vision. Some of the most important applications of computer vision are in the fields that deal with facial data. Face Detection and recognition are being widely used in security-based applications.

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. In modern times, almost everyone in this world uses some sort of transportation every day. Some people are rich enough to have their own vehicles while others use public transportation. 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. While on road, an automobile wields the most power and in irresponsible hands, it can be destructive and sometimes, that carelessness can harm lives even of the people on the road. One kind of carelessness is not admitting when we are too tired to drive. In order to monitor and prevent a

LITERATURE SURVEY

2.LITERATURE SURVEY

LITERATURE REVIEWS:

TITLE 1: Face Detection Using Deep Learning, September 2021, 2nd International Conference On Advances In Computing, Communication, Embedded And Secure Systems (ACCESS)

AUTHOR: Pradnya Kedari; Mihir Kapile; Divya Kadole; Sagar Jaikar

DESCRIPTION: Using deep learning algorithm to identify the basic human face on multiple datasets, including FER-2013 (Facial Expression Recognition 2013)and CK+ (Extended Cohn-Kanade), the accuracy is 60% for FER 2013 dataset, and for CK+, the model achieved significant improvement, highest accuracy was 99.1% and average accuracy was 93%

TITLE 2: Deep Face Recognition: A Survey,March 2020, IEEE Transactions on Affective Computing

AUTHOR: Shan Li and Weihong Deng

DESCRIPTION: Deep neural networks have increasingly been leveraged to learn discriminative representations for automatic FER. Recent deep FER systems generally focus on two important issues: overfitting caused by a lack of

sufficient training data and expression-unrelated variations, such as illumination, head pose and identity bias.

TITLE 3: Efficient Facial Recognition Algorithm Based on Hierarchical Deep Neural Network Structure, March 2019, Department of IT Engineering, Sookmyung Women's University, Seoul, South Korea

AUTHOR: Ji-hae Kim, Byung-gyu Kim, Partha Pratim Roy, Da-mi Jeong

DESCRIPTION: An efficient facial expression recognition algorithm combining appearance feature and geometric feature based on deep neural networks for more accurate and efficient facial expression recognition. The appearance feature-based network extracts the holistic feature of the LBP feature containing the AUs information. The geometric feature-based network extracts the dynamic feature, which is the face landmark change centered on the coordinate movement between the neutral face and the peak emotion.

TITLE 4: Survey on AI-Based Multimodal Methods for Eye Detection, March 2019, High-Performance Modelling and Simulation for Big Data Applications

AUTHOR: Catherine Marechal, Dariusz Mikołajewski, Krzysztof Tyburek,Piotr Prokopowicz, Lamine Bougueroua, Corinne Ancourt, Katarzyna Węgrzyn-Wolska

DESCRIPTION: A novel multimodal implicit emotion recognition system can be built upon an AI-based model designed to extract information on the emotion from different devices. To feed such a model, a video data captured by the camera embedded in the user's device (laptop, desktop, tablet, etc.), an audio signals collected from microphones embedded in mobile devices, and motion signals generated by sensors in wearable devices can be used.

TITLE 5: Facial recognition using deep learning: review and insights, August 2020, The 2nd International Workshop on the Future of Internet of Everything (FIoE), Belgium

AUTHOR: Wafa Mellouk, Wahida Handouzi

DESCRIPTION: Recent research on FER systems has been discussed, differentarchitectures of CNN and CNN LTSM have also been elaborated. FER is one ofthe most important ways of providing information about the emotional state, but they are always limited by learning only the six-basic emotion plus neutral.

TITLE 6: Deep Learning for Understanding Faces: Machines May Be Just as Good, or Better than Humans, Jan. 2018, IEEE Signal Processing Magazine, Volume: 35, Issue: 1

AUTHOR: Rajeev Ranjan; Swami Sankaranarayana Ankan Bansal; Navaneeth Bodla

DESCRIPTION: Different modules involved in designing an automatic face recognition system and the role of deep are discussed and learned for each of

TITLE 7: Cost-effective real-time recognition for human face and eye detection and segmentation using deep learning with normalized facial cropping preprocess March 2021, Multimedia Tools and Applications

AUTHOR: Ta-Te Lu, Sheng-Cheng Yeh, Chia-Hui Wang, Min-Rou Wei

DESCRIPTION: The EAGR system first applies normalized facial cropping (NFC) as a preprocessing method for training data before data augmentation, then uses convolution neural network (CNN) as three training models for recognizing seven emotions (six basics plus one neutral emotion), four age groups, and two genders.

TITLE 8: OpenCV and CV2 A Critical Review Study: A Critical Review Study June 2018, Universiti Sains Malaysia, Penang

AUTHOR: Rasha Ragheb Atallah, Amirrudin Kamsin, Maizatul Akmar Ismail, Sherin Ali Abdelrahman, Saber Zerdoumi

DESCRIPTION: A complete survey of the state-ofthe art techniques for age estimation and face recognition have been reviewed and discussed via face images. The results of this study indicated that the SVM (99.80%) and the LBP (98.7%) had the highest detection accuracy rates, along with GAP (99.85 %).

TITLE 9: Training a Deep Learning model using Conv2D module, The Fifth Information Systems International Conference 2019

AUTHOR: Pipit Utami, Rudy Hartanto, Indah Soesanti

DESCRIPTION: The trend of developing recognition classifier is the use of CNN and modified CNN. The things that need to be explored are the description of the types of FET (needs to be oriented towards academic emotion) and the suitability of the CNN algorithm modifications as the solution to four problems that may arise during teaching

.

TITLE 10: Classification Algorithms in Machine Learning, July 2020, Computer Vision and Image Understanding Volume 196, 102961

AUTHOR: Alice Othmania, Abdul Rahman Taleb, Hazem Abdelkawy, Abdenour Hadid

DESCRIPTION: The framework based on Xception network outperforms the state-of-the-art methods based on deep or shallow learning for classification of images with an MAE of 2.35 years when pre-trained on ImageNet and an MAE of 2.01 when pre-trained on CASIAWeb face dataset.

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

The existing system of driver drowsiness detection system has following disadvantages. Mainly, using of two cameras in the system one for monitoring the head movement and the other one for facial expressions. The other disadvantage is aging of sensors and all these sensors are attached to the driver's body which may affect the driver. So, to overcome all these disadvantages we designed a system in which a live camera is used for monitoring the driver drowsiness condition and alert the driver which reduces the road accidents.

- Most of the existing systems use SVM (Support Vector Machine) algorithm to classify the components in the input video.
- While cropping the region of interest components in the video is not accurate. Sometimes it will show regions wrong.
- To sense the eyes first we have to create boundary boxes for that and a classification algorithm is then used to classify it as closed or open.
- The algorithm of SVM will not support. Another existing alternative is to usea formula to determine the distance between the eyelids, which makes the process a bit more complex.

Applying a formula, and classifying based on theresult in a slow process compared to the proposed here. So, any of these methods are not efficient. Though there are methods in which different sensorsare used to collect data like the heart rate, pulse rate etc., and make calculationsand predictions usingthis data. This is also complex and time consuming, andsometimes the variations in heart rate and pulse rate maybe due to various medical conditions.

3.1.1 DISADVANTAGES:

- **SVM algorithm** is not suitable for large data sets.
- **SVM** does not perform very well when the data set has more noise.
- In cases where the number of features for each data point exceeds the number of training data samples, the **SVM** will underperform.
- While travelling in a vehicle, the human face won't be still, SVM algorithm fails to track the moving face and the ROI doesn't always follow face efficiently
- Slow process(Due to application of a formula to calculate the distance between eye lids).
- Unreliable, as it is not fast enough and accurate

3.2 PROPOSED SYSTEM:

Drowsiness Detecting System captures and the live video of the driver, and to sense the eyes first we have to create boundary boxes for that, and then identify the face and then for the eyes, which is then classified using a classification algorithm, which the algorithm of SVM will not support. With the simple binaryclassification algorithm, the frames will be classified as closed eyeor open eye by the deep learning model, and then the system acts accordingly.If he/she closes their eyes for a fraction of time or more than blinking time the system identifies the drowsiness and it warns or alerts them with an alarm and prevents accidents from happening

The first step is to acquire and preprocess the video frames, which are going to be used as the actual inputs for the classification process. The features extraction process was used to achieve this. This process helps in reducing the dimensionality of the raw inputs (i.e. each video frame pixels), and also in selecting meaningful variables through a combination of these raw inputs. Then, the drowsiness classification process is used to classify the driver's state into either an awake or drowsy state using decision tree algorithm. If it is classified as closed then warning sound will be played else the next frame will be following the same process

As the process is not complex and simple compared to the existing systems it is Faster compared to the existing system. The training dataset had images of humans of different genders, colour, age, with and without spectacles Works even in low light conditions. The system has less hardware requirements. As the system uses fewer resources therefore the cost of the system is less. There by reducing accidents caused due to driver drowsiness

3.2.1 ADVANTAGES:

- Faster compared to the existing system.
- Thanks to HAAR algorithm by Intel, this algorithm so accurately creates a ROI around the human face and tracks it no matter however the movement is, it never loses the track of it.
- Works even in low light conditions.
- The system has less hardware requirements.
- As the system uses fewer resources therefore the cost of the system is less.
- There by reducing accidents caused due to driver drowsiness.

3.3 FEASIBILITY STUDY:

3.3.1 Introduction:

A Feasibility Study includes a nitty gritty evaluation of the need, value, and common sense of a proposed enterprise, such as framework development. The procedure of outlining and executing record-keeping frameworks has adequate responsibility and asset suggestions for an association. Possibility study will enable you to settle on educate and straightforward choice at urgent focuses during the developmental procedure to decide if it is operationally, economically and in fact reasonable to deliver with a specific strategy.

3.3.2 Financial Feasibility:-

Being a user friendly softwarewill have no cost .Since we cannot transfer any multimedia file ,the bandwidth required is very low. In this, we want to buy domain and data will be fetched from the server, so we also buy separate host. It reduces paper and printing cost and also the usage of paper is declined. Since, it is one time investment it will become profitable. It will apply in the local Universities and Colleges.

In addition to the reduction of cost, the extra effort that was put for making paper and manual updating the attendance of the students, since the process is automated.

From these it's clear that this project is financially feasible.

3.3.3 Technical feasibility:

This project is mobile based application .The main technologies that are associated with project are

- ANACONDA

- TENSORFLOW(BACK END)
- NUMPY
- KERAS
- PYGAME
- BINARY CLASSIFICATION
- PYTHON
- HAAR FILES by INTEL

Each of the technologies are freely available and the technical skills required are manageable. Time limitations of the product and the ease of implementing using these technologies are synchronized.

This requires less bandwidth to transfer data that stored in database, because in this we cannot transfer any multimedia file. From this it is clear that our project is technically feasible.

3.3.4 Resource Feasibility:

Resources that are required for our project are:

- Programming Device
- Hosting Space
- Programming Tools
- Programming Individuals

So it's clear that project has required resource feasibility.

3.4 HARDWARE REQUIREMENTS:

3.4.1 CAMERA:

A camera is an optical instrument that captures a visual image. At a basic level, cameras consist of sealed boxes (the camera body), with a small hole (the aperture) that allows light through to capture an image on a light-sensitive surface (usually photographic film or a digital sensor).

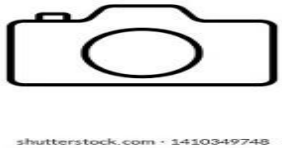


FIG.3.1 **CAMERA**

3.4.2 SMART PHONE/LAPTOP:

Smart phones (or) Laptops (or) Tablets is used to view the person's identical movement of the eyes. Which helps to execute the appendices and state the condition of the person he/she who is driving the car weather they are drowsy or else they are not drowsy.

FIG.3.2 **DEVICES TO CAPURE OUR VIDEO**

3.5 SOFTWARE REQUIREMENTS:

3.5.1 ANACONDA

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field. Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues. It gives you a seamless environment which simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready

How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let's discuss some fundamental commands you can use to start using this package manager.

Listing All Environments

To begin using Anaconda, you'd need to see how many Conda environments are present in your machine.

conda env list

It will list all the available Conda environments in your machine.

Creating a New Environment

You can create a new Conda environment by going to the required directory and use this command:

conda create -n <your_environment_name>

You can replace <your_environment_name> with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y:

proceed ([y])/n)?

On the other hand, if you want to create an environment with a particular version of Python, you should use the following command:

conda create -n <your_environment_name> python=3.6

Similarly, if you want to create an environment with a particular package, you can use the following command:

conda create -n <your_environment_name>pack_name

Here, you can replace pack_name with the name of the package you want to use.

If you have a .yml file, you can use the following command to create a new Conda environment based on that file:

conda env create -n <your_environment_name> -f <file_name>.yml

We have also discussed how you can export an existing Conda environment to a .yml file later in this article.

Activating an Environment

You can activate a Conda environment by using the following command:

conda activate <environment_name>

You should activate the environment before you start working on the same. Also, replace the term <environment_name> with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command:

conda deactivate

Installing Packages in an Environment

Now that you have an activated environment, you can install packages into it by using the following command:

conda install <pack_name>

Replace the term <pack_name> with the name of the package you want to install in your Conda environment while using this command.

Updating Packages in an Environment

If you want to update the packages present in a particular Conda environment, you should use the following command:

conda update

The above command will update all the packages present in the environment. However, if you want to update a package to a certain version, you will need to use the following command:

conda install <package_name>=<version>

Exporting an Environment Configuration

Suppose you want to share your project with someone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can create an environment configuration .yml file and share it with that person. Now, they can create an environment like your one by using the .yml file.

For exporting the environment to the .yml file, you'll first have to activate the same and run the following command:

conda env export ><file_name>.yml

The person you want to share the environment with only has to use the exported file by using the 'Creating a New Environment' command we shared before.

Removing a Package from an Environment

If you want to uninstall a package from a specific Conda environment, use the following command:

conda remove -n <env_name><package_name>

On the other hand, if you want to uninstall a package from an activated environment, you'd have to use the following command:

conda remove <package_name>

Deleting an Environment

Sometimes, you don't need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment.

FIG.3.3 **ANACONDA LOGO**

3.5.2 TENSORFLOW (BACK END):

Machine learning is a complex discipline. But implementing machine learning models is far less daunting and difficult than it used to be, thanks to machine learning frameworks—such as Google's TensorFlow—that ease the process of acquiring data, training models, serving predictions, and refining future results.

Created by the Google Brain team, TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

FIG.3.4 **TENSORFLOW**

3.5.3 NUMPY:

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

At the core of the NumPy package, is the *ndarray* object. This encapsulates *n*-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences

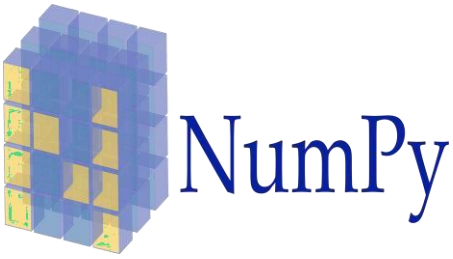


FIG.3.5 **NUMPY**

3.5.4 KERAS:

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. *Being able to go from idea to result as fast as possible is key to doing good research.* Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

Keras empowers engineers and researchers to take full advantage of the scalability and cross-platformcapabilities of TensorFlow 2: you can run Keras on TPU or on large clusters of GPUs, and you can exportyour Keras models to run in the browser or on a mobile device.

***FIG.3.6* KERAS**

3.5.5 PYGAME:

***FIG.3.7* PYGAME**

PYGAME is a Python wrapper for the SDL library, which stands for **Simple DirectMedia Layer**. SDL provides cross-platform access to your system's underlying multimedia hardware components, such as sound, video, mouse, keyboard, and joystick. pygame started life as a replacement for the stalled PySDL project. The cross-platform nature of both SDL and pygame means you can write games and rich multimedia Python programs for every platform that supports them.Game programming is very rewarding nowadays and it can also be used in advertising and asa teaching tool too. Game development includes mathematics, logic, physics, AI, and much more and it can be amazingly fun.

SYSTEM DESIGN

4.SYSTEM DESIGN

UML DIAGRAMS:

4.1 ENTITY-RELATIONSHIP DIAGRAM:

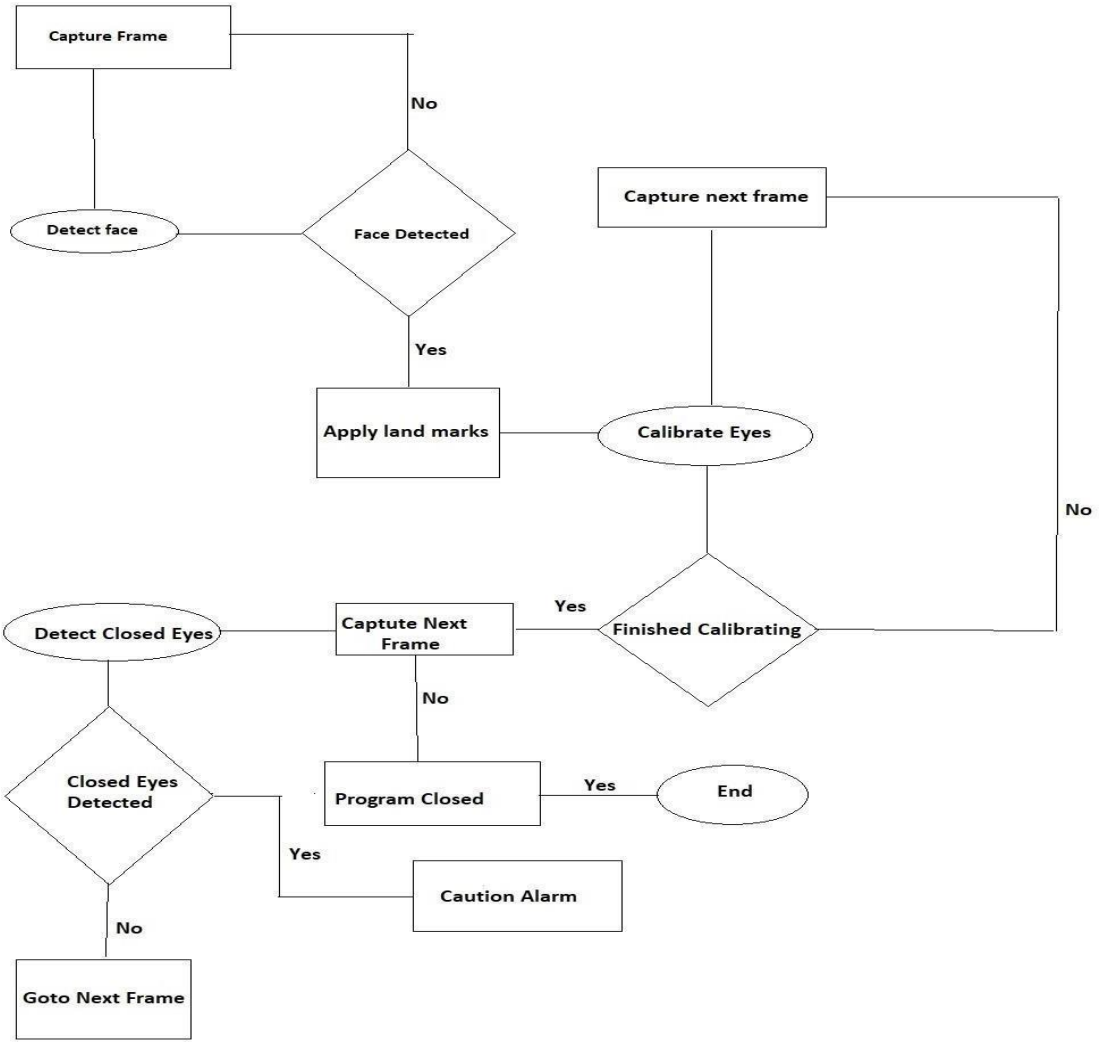


FIG.4.1 **ENTITY-RELATIONSHIP DIAGRAM**

The Above ER Diagram shows the process of how the drowsiness is detected when a person he/she driving and gives the caution alarm if the respective person is closing his eyes more than the blinking time period. In the above Fig.4.1 the entities and attributes used are capturing of the frame, detecting of the face, applying of the land marks, the process of calibrating eyes, detection of the eyes, warning of the caution alarm and the relationships are conformation of the face detection, finishing the process of calibrating the eyes, detection of the closed eyes.

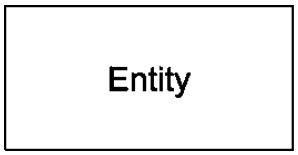
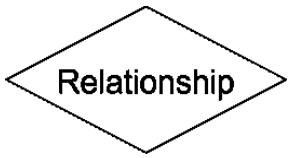
Symbol Name	Symbol	Description
Entity		An entity is represented by a rectangle which contains the entity's name.
Attribute		In the Chen notation, each attribute is represented by an oval containing attribute's name
Strong Relationship		A relationship where entity is existence-independent of other entities, and PK of Child doesn't contain PK component of Parent Entity. A strong relationship is represented by a single rhombus
One or More		It represents One or More
Many - to - Many		It represents a one through many on both sides of a relationship

TABLE 4.1 **E-R DIAGRAM SYMBOL DESCRIPTION**

4.2 DFD DIAGRAM:

4.2.1 Level 0 DFD:

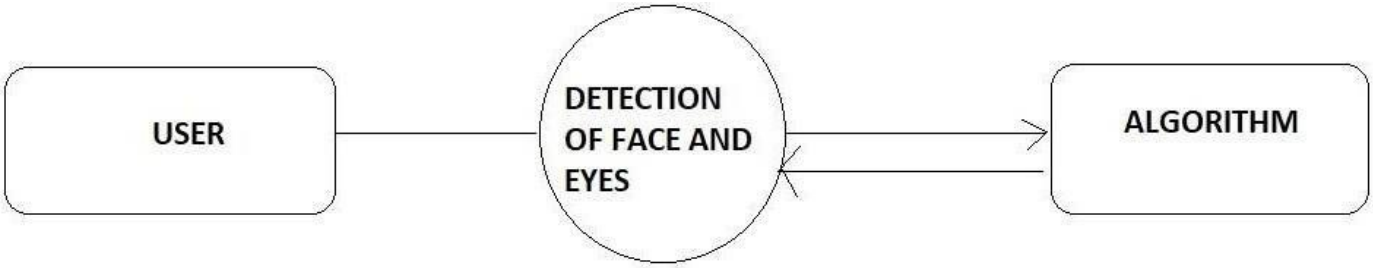


FIG.4.2 **LEVEL-0-DFD**

4.2.2 Level 1 DFD:

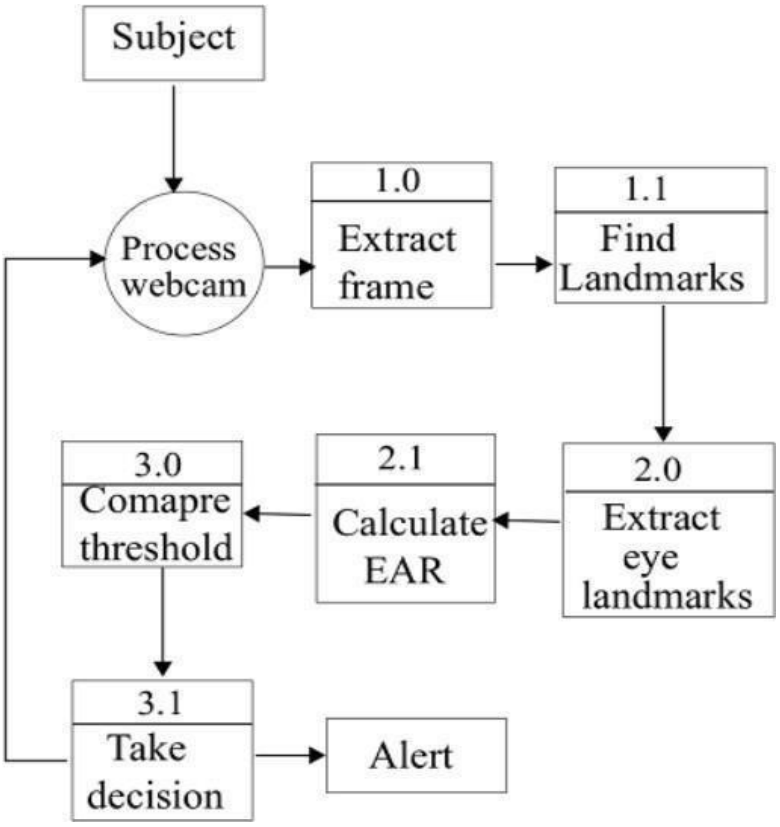


FIG.4.3 **LEVEL-1-DFD**

4.3 USECASE DIAGRAM:

Depicts the various users of the system and how they are going to use the system to meet the requirement objectives.Use Case Diagram for EMV Chip Embedded Identity Card/Permanent Hallticket for college students contains Four Actors, some Usecases and DataSource.

Below are the Symbols used in Usecase diagram with Description:

Symbol Name	Symbol	Description
Actor		Actors are the users of a system.
Usecase		Label the ovals with verbs that represent the system's functions.
Data Source		A Non-Human Actor is represented by this symbol.

TABLE 4.2 **USE CASE DIAGRAM SYMBOL DESCRIPTION**

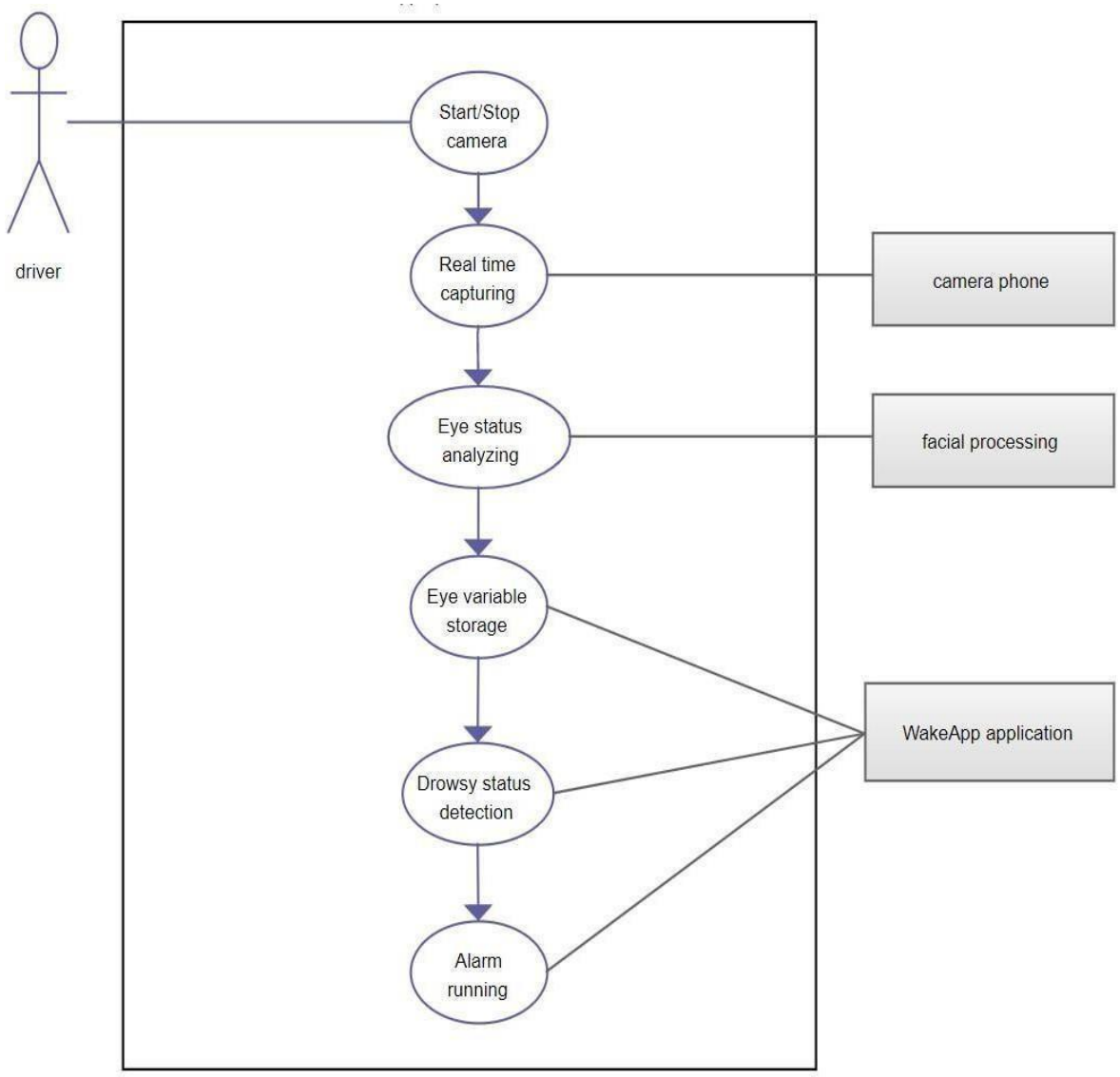
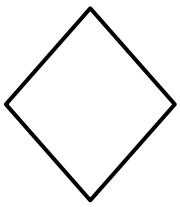


FIG.4.4 **USECASE DIAGRAM**

4.4 ACTIVITY DIAGRAM:

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

Symbol Name	Symbol	Description
Start/Initial State		A small filled circle followed by an arrow represents start point for any activity diagram.
Activity State		An action state represents the non-interruptible action of objects.
Decisions and Branching		A diamond represents a decision with alternate paths.The outgoing alternates should be labelled with a condition or guard expression. You can also label one of the paths "else."

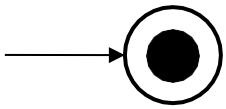
Final State		An arrow pointing to a filled circle nested inside another circle represents the final action state.

TABLE 4.3 **ACTIVITY DIAGRAM SYMBOL DESCRIPTION**

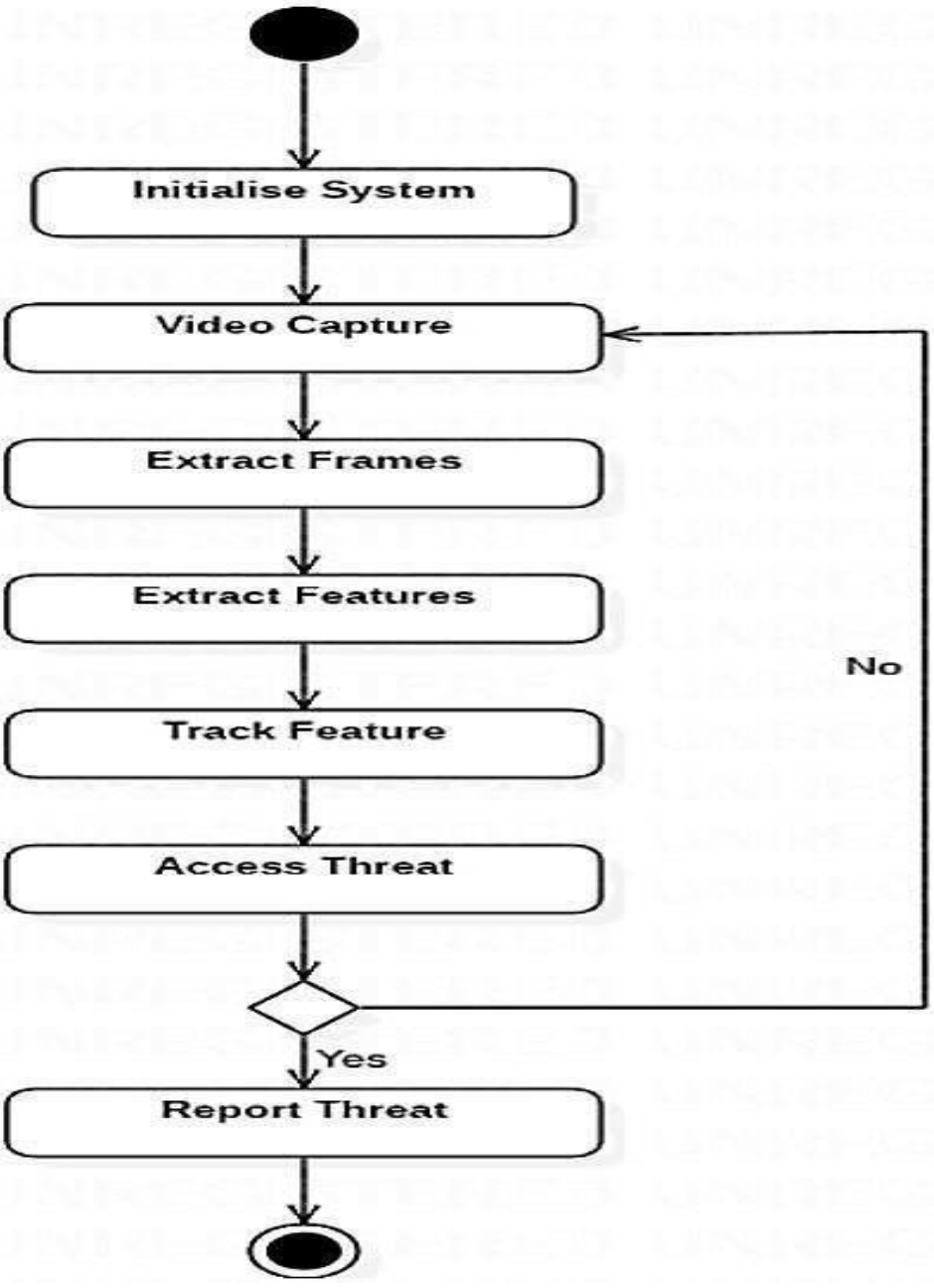


FIG.4.5 **ACTIVITY DIAGRAM**

4.5 CLASS DIAGRAM:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

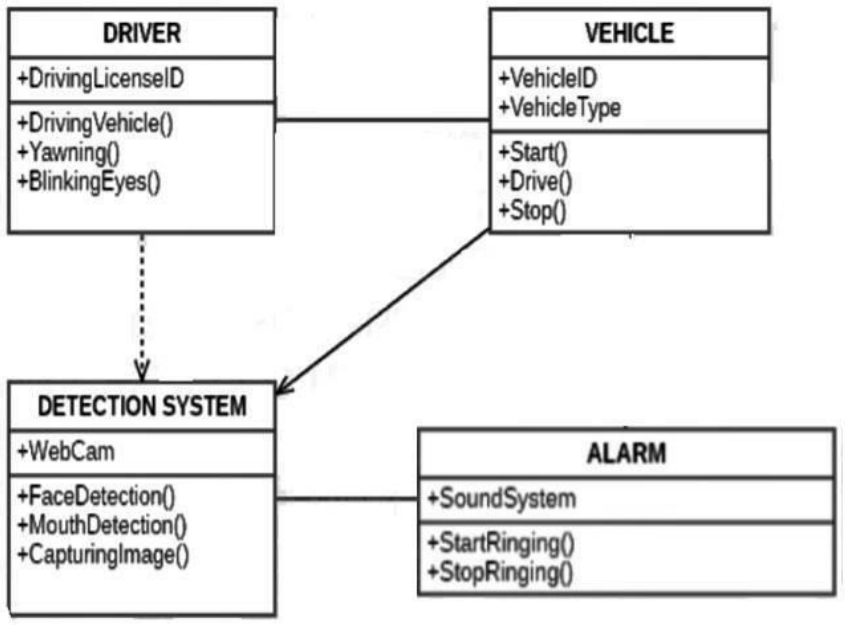


FIG.4.6 **CLASS DIAGRAM**

4.6 COLLABORATION DIAGRAM:

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements

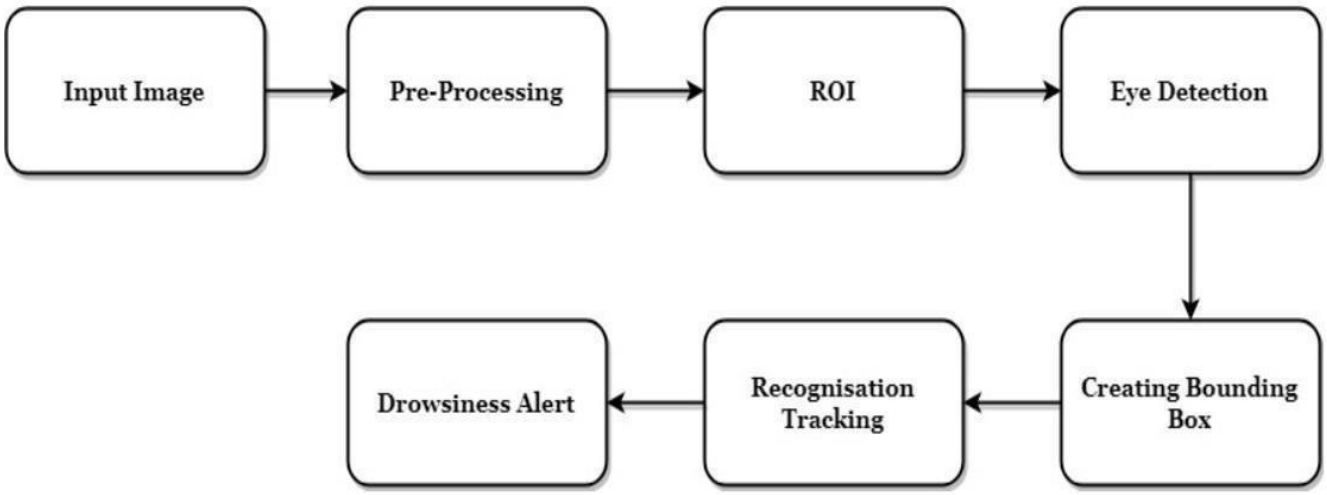


FIG.4.7 **COLLABORATION DIAGRAM**

4.7 ARCHITECTURE DIAGRAM:

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. An architecture diagram is a visual representation of all the elements that make up part, or all, of a system. Above all, it helps the engineers, designers, stakeholders — and anyone else involved in the project — understand a system or app's layout. Think of it as being a bit like a blueprint to a building: You can see the thing **as a** whole, as well as different kinds of interior views, and things like pipes, walls, floorplans, and so on

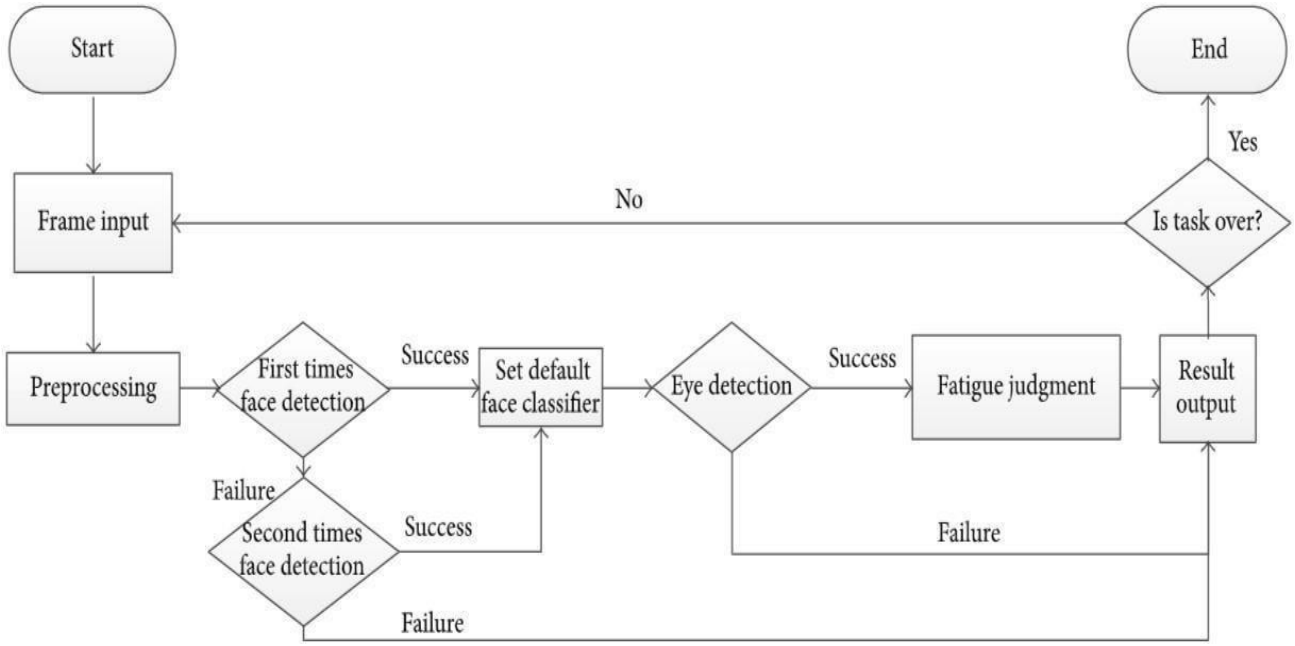


FIG.4.8 **ARCHITECTURE DIAGRAM**

MODULE DESCRIPTION

5. LIST OF MAIN MODULES

- Module 1: Face Detection
- Module 2: Classification of Image

5.1 MODULE 1: Face and Eye Detection using Haar Cascades

Object Detection using Haar feature-based cascade classifiers is an effective Now all possible sizes and locations of each kernel is used to calculate plenty of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find sum of pixels under white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels. Nice, isn't it? It makes things super-fast.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by **Adaboost**.

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated.Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).

around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).

So now you take an image. Take each 24x24 window. Apply 6000 features to it.Check if it is face or not. Wow.. Wow.. Isn't it a little inefficient and time consuming? Yes, it is. Authors have a good solution for that.

In an image, most of the image region is non-face region. So it is a better idea tohavea simple method to check if a window is not a face region. If it is not, discardit in a single shot. Don't process it again. Instead focus on region where there canbe a face. This way, we can find more time to check a possible face region.

For this they introduced the concept of **Cascade of Classifiers**. Instead ofapplying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

Authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 featuresin first five stages. (Two features in the above image is actually obtainedas the best two features from Adaboost). According to authors, on an average, 10features out of 6000+ are evaluated per sub-window.

So this is a simple intuitive explanation of how Viola-Jones face detection works.Read paper for more details or check out the references in Additional Resourcessection.

Haar-cascade Detection in OpenCV

OpenCV comes with a trainer as well as detector. If you want to train your own classifier for any object like car, planes etc. you can use OpenCV to create one. Its fulldetails are given here: Cascade Classifier Training.

First we need to load the required XML classifiers. Then load our input image (or video) in grayscale mode.

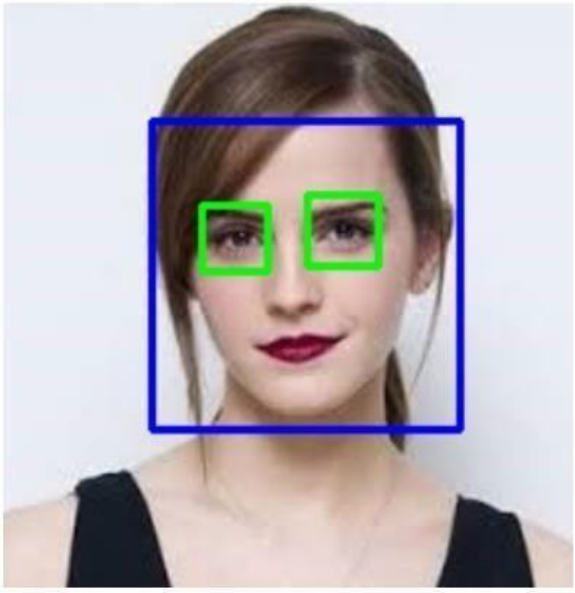
```
import numpy as np
import cv2 face_cascade
=
cv2.CascadeClassifier('haarcascade_frontalface_default.xml ')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

img = cv2.imread('sachin.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Now we find the faces in the image. If faces are found, it returns the positions of detected faces as Rect(x,y,w,h). Once we get these locations, we can create a ROI for the face and apply eye detection on this ROI (since eyes are always on the face !!!).

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
    img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
roi_gray = gray[y:y+h, x:x+w]
roi_color = img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
cv2.imshow('img',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Result looks like below:



5.2 MODULE 2: Classification

We are using CNN classifier for predicting the eye status. To feed our image into the model, we need to perform certain operations because the model needsthe correct dimensions to start with. First, we convert the color image into grayscale using **r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY)**.

Then, we resize the image to 24*24 pixels as our model was trained on 24*24 pixel images **cv2.resize(r_eye, (24,24))**. We normalize our data for better convergence **r_eye = r_eye/255** (All values will be between 0-1). Expand the dimensions to feed into our classifier. We loaded our model using **model = load_model('models/cnnCat2.h5')** . Now we predict each eye with our model

lpred = model.predict_classes(l_eye). If the value of lpred[0] = 1, it states that eyes are open, if value of lpred[0] = 0 then, it states that eyes are closed.

5.2.1 BINARY CLASSIFICATION

There are many metrics that can be used to measure the performance of a classifier or predictor; different fields have different preferences for specific metrics due to different goals. In medicine sensitivity and specificity are often used, while in information retrieval precision and recall are preferred. An important distinction is between metrics that are independent of how often each category occurs in the population (the *prevalence*), and metrics that depend on the prevalence – both types are useful, but they have very different properties.

Given a classification of a specific data set, there are four basic combinations of actual data category and assigned category: true positives TP (correct positive assignments), true negatives TN (correct negative assignments), false positives FP (incorrect positive assignments), and false negatives FN (incorrect negative assignments).

Assigned Actual	**Test outcome *positive***	**Test outcome *negative***
Condition positive	True *positive*	False *negative*
Condition negative	False *positive*	True *negative*

TABLE 5.1 **BINARY CLASSIFIER**

5.3 METHODOLOGY/ALGORITHM:

Step 1 – Take Image as Input from a Camera

With a webcam, we will take images as input. So to access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, cv2.VideoCapture(0) to access the camera and set the capture object (cap). cap.read() will read each frame and we store the imagein a frame variable.

Step 2 – Detect Face in the Image and Create a Region of Interest (ROI)

To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don't need color information to detect the objects. We willbe using haar cascade classifier to detect faces. This line is used to set our classifier face = cv2.CascadeClassifier(' path to our haar cascade xml file'). Then we perform the detection using faces = face.detectMultiScale(gray). It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object. Now we can iterate over the faces and drawboundary boxes for each face.

```
for (x,y,w,h) in faces:
cv2.rectangle(frame, (x,y), (x+w, y+h), (100,100,100), 1 )
```

Step 3 – Detect the eyes from ROI and feed it to the classifier. The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in leye and reye respectively then detect the eyes using left_eye = leye.detectMultiScale(gray). Now we need to extract only the eyes data from the full image. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame with this code.

l_eye = frame[y : y+h, x : x+w]

l_eye only contains the image data of the eye. This will be fed into our CNN classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into r_eye.

Step 4 – Classifier will Categorize whether Eyes are Open or Closed

We are using CNN classifier for predicting the eye status. To feed our image into the model, we need to perform certain operations because the model needs the correct dimensions to start with. First, we convert the color image into grayscale using r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY). Then, we resize the image to 24*24 pixels as our model was trained on 24*24 pixel images cv2.resize(r_eye, (24,24)). We normalize our data for better convergence r_eye = r_eye/255 (All values will be between 0-1). Expand the dimensions to feed into our classifier.

We loaded our model using model = load_model('models/cnnCat2.h5') . Now we predict each eye with our model

lpred = model.predict_classes(l_eye). If the value of lpred[0] = 1, it states that eyes are open, if value of lpred[0] = 0 then, it states that eyes are closed.

Step 5 – Calculate Score to Check whether Person is Drowsy

The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score. We are drawing the result on the screen using cv2.putText() function which will display real time status of the person.

cv2.putText(frame, "Open", (10, height-20), font, 1, (255,255,255), 1, cv2.LINE_AA)

A threshold is defined for example if score becomes greater than 15 that means the person's eyes are closed for a long period of time. This is when we beep the alarm using sound.play()

TESTING

6.TESTING

6.1 Testing Objectives

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTS

6.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

6.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent.

6.2.3 Functional Testing

Functional testing provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

6.2.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing predriven process links and integration points.

6.2.5 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirement.

6.3 TESTCASES AND RESULTS:

TEST ID	TEST CONDITION	SYSTEM BEHAVIOUR	SYSTEM BEHAVIOUR	EXPECTED RESULT
T01	NSGW	Straight Face,Good Light,With Glasses	Non Drowsy	Non Drowsy
T02	DTGN	Tilted Face,Good Light,No Glasses	Drowsy	Drowsy
T03	DTGY	Tilted Face,Good Light,With Glasses	Drowsy	Drowsy
T04	NSDN	Straight Face,Dim Light,Without Glasses	Non Drowsy	Non Drowsy

TABLE 6.1 TEST CASES AND POSSIBLE RESULTS

CONCLUSION

7. CONCLUSION

7.1 Conclusion:

It completely meets the objectives and requirements of the system. The framework has achieved an unfaltering state where all the bugs ghkgkghave been disposed of. The framework cognizant clients who are familiar with the framework and comprehend it's focal points and the fact that it takes care of the issue of stressing out for individuals having fatigue-related issues to inform them about the drowsiness level while driving.

7.2 Future Enhancement:

The model can be improved incrementally by using other parameters likeblink rate, yawning, state of the car, etc. If all these parameters are used itcan improve the accuracy by a lot. We plan to further work on the project by adding a sensor to track theheart rate in order to prevent accidents caused due to sudden heart attacks to drivers. Same model and techniques can be used for various other uses like Netflix and other streaming services can detect when the user is asleepand stop the video accordingly. It can also be used in application that prevents user from sleeping

APPENDICES

8.APPENDIX 1

8.1 CODING:

Trained.py

```
import os
from keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
from keras.utils.np_utils import to_categorical
import random,shutil
from keras.models import Sequential
from keras.layers import Dropout,Conv2D,Flatten,Dense, MaxPooling2D, BatchNormalization
from keras.models import load_model


def generator(dir, gen=image.ImageDataGenerator(rescale=1./255), shuffle=True,batch_size=1,target_size=(24,24),class_mode='categorical' ):
 return gen.flow_from_directory(dir,batch_size=batch_size,shuffle=shuffle,color_mode='grayscale',class_mode=class_mode,target_size=target_size)


BS= 32
TS=(24,24)
train_batch= generator('data/train',shuffle=True, batch_size=BS,target_size=TS)
valid_batch= generator('data/valid',shuffle=True, batch_size=BS,target_size=TS)
```

```
SPE= len(train_batch.classes)//BS
VS = len(valid_batch.classes)//BS
print(SPE,VS)
# img,labels= next(train_batch)
# print(img.shape)

model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(24,24,1)),
    MaxPooling2D(pool_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool_size=(1,1)),
#32 convolution filters used each of size 3x3
#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(1,1)),

#64 convolution filters used each of size 3x3
#choose the best features via pooling

#randomly turn neurons on and off to improve convergence
    Dropout(0.25),
#flatten since too many dimensions, we only want a classification output
    Flatten(),
#fully connected to get all relevant data
    Dense(128, activation='relu'),
```

```
#one more dropout for convergence' sake :)
    Dropout(0.5),
#output a softmax to squash the matrix into output probabilities
    Dense(2, activation='softmax')
])
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accurac
y'])


model.fit_generator(train_batch,
validation_data=valid_batch,epochs=15,steps_per_epoch=SPE
,validation_steps=VS)


model.save('models/cnnCat2.h5', overwrite=True)
```

Detection.py

```
import cv2
import os
from keras.models import load_model
import numpy as np
import pygame
from pygame import mixer
import time

mixer.init()
sound = mixer.Sound('alarm (2).wav')
```

```
face = cv2.CascadeClassifier('haar cascade files\haarcascade_frontalface_alt.xml')
leye = cv2.CascadeClassifier('haar cascade files\haarcascade_lefteye_2splits.xml')
reye = cv2.CascadeClassifier('haar cascade
files\haarcascade_righteye_2splits.xml')



lbl=['Close','Open']

model = load_model('Trained\cnnCat2.h5')
path = os.getcwd()
cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
count=0
score=0
thicc=2
rpred=[99]
lpred=[99]

while(True):
    ret, frame = cap.read()
    height,width = frame.shape[:2]

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
    faces =
face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))
    left_eye = leye.detectMultiScale(gray)
    right_eye =  reye.detectMultiScale(gray)
cv2.rectangle(frame, (0,height-50) , (200,height) , (0,0,0) , thickness=cv2.FILLED
)
for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) , 1 )

    for (x,y,w,h) in right_eye:
        r_eye=frame[y:y+h,x:x+w]
        count=count+1
        r_eye = cv2.cvtColor(r_eye,cv2.COLOR_BGR2GRAY)
        r_eye = cv2.resize(r_eye,(24,24))
        r_eye= r_eye/255
        r_eye=  r_eye.reshape(24,24,-1)
        r_eye = np.expand_dims(r_eye,axis=0)
        rpred = model.predict(r_eye)
        rpred = np.argmax(rpred, axis=1)
        if(rpred[0]==1):
            lbl='Open'
        if(rpred[0]==0):
            lbl='Closed'
        break

    for (x,y,w,h) in left_eye:
```

```
        l_eye=frame[y:y+h,x:x+w]
        count=count+1
        l_eye = cv2.cvtColor(l_eye,cv2.COLOR_BGR2GRAY)
        l_eye = cv2.resize(l_eye,(24,24))
        l_eye= l_eye/255
        l_eye=l_eye.reshape(24,24,-1)
        l_eye = np.expand_dims(l_eye,axis=0)
        lpred = model.predict(l_eye)
        lpred = np.argmax(lpred, axis=1)
        if(lpred[0]==1):
            lbl='Open'
         if(lpred[0]==0):
           lbl='Closed'
        break


    if(rpred[0]==0 and lpred[0]==0):
        score=score+1
        cv2.putText(frame,"Closed",(10,height-20), font,
1,(255,255,255),1,cv2.LINE_AA)
    # if(rpred[0]==1 or lpred[0]==1):
    else:
        score=score-1
        cv2.putText(frame,"Open",(10,height-20), font,
1,(255,255,255),1,cv2.LINE_AA)
    if(score<0):
        score=0
```

```
    cv2.putText(frame,'Score:'+str(score),(100,height-20), font,
1,(255,255,255),1,cv2.LINE_AA)
    if(score>8):
        #person is feeling sleepy so we beep the alarm
        cv2.imwrite(os.path.join(path,'image.jpg'),frame)
        cv2.putText(frame, 'You are Drowsy', (80,height-60), font, 2, (209, 80, 0,
255), 3, cv2.LINE_AA)
        try:
            sound.play()
 except:  # isplaying = False
            pass
        if(thicc<16):
            thicc= thicc+2
        else:
            thicc=thicc-2
            if(thicc<2):
                thicc=2
        cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thicc)
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

8.2 SCREEN SHOTS:

FIG.8.1 **SCREEN SHOT WHEN YOU ARE FEELING DROWSY WITH SPECTACLES IN GOOD LIGHT CONDITIONS**

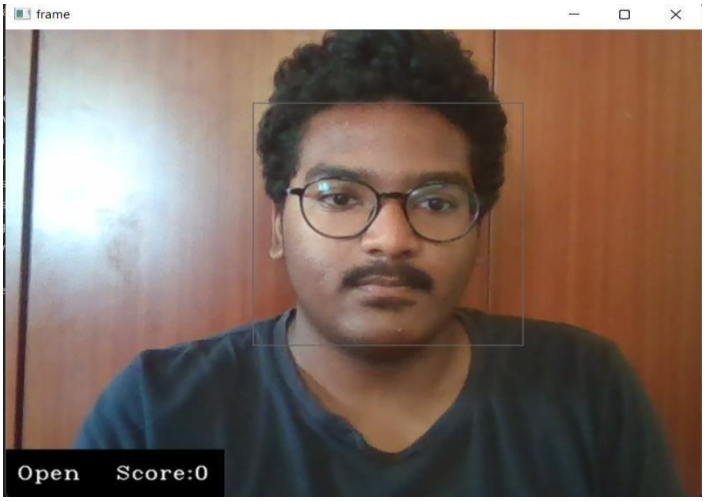


FIG.8.2 **SCREEN SHOT OF WHEN YOU ARE NOT FEELING DROWSY WITH SPECTACLES IN GOOD LIGHT CONDITIONS**

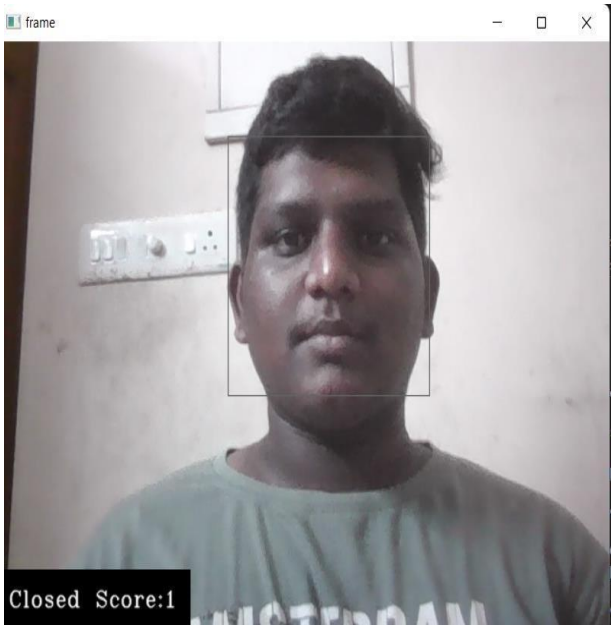


FIG.8.3 **SCREEN SHOT WHEN YOU ARE NOT FEELING DROWSY WITHOUT SPECTACLES IN LOW LIGHT CONDITIONS**

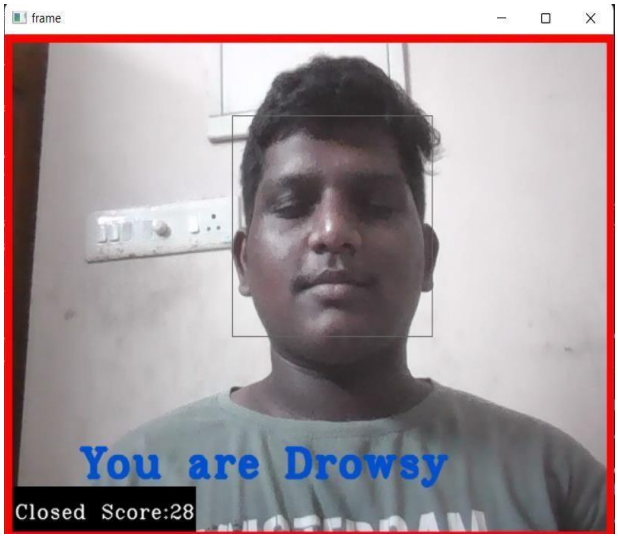


FIG.8.4 **SCREEN SHOT WHEN YOU ARE FEELING DROWSY WITHOUT SPECTACLES IN LOW LIGHT CONDITIONS**

BIBILIOGRAPHY

9. BIBILIOGRAPHY

[1]. Z. Wei, P. Zhao, and S. Ai, "Efficiency Evaluation of Beijing Intelligent Traffic Management System Based on Super-DEA," in Journal of TransportationSystems Engineering and Information Technology, vol. 12, no. 3, pp. 19–23, Jun.2012.

[2]. S. C. Davis, S. W. Diegel, and R. G. Boundy, Transportation Energy Data Book: Edition 31, 31st ed. United States: Center for Transportation Analysis Energy and Transportation Science Division, 2012.

[3]. J. He, Z. Zeng, and Z. Li, "Benefit Evaluation Framework of Intelligent Transportation Systems," in Journal of Transportation Systems Engineering and Information Technology, vol. 10, no. 1, pp. 81–87, Feb. 2010.

[4]. MIROS. (2011). General Road Accident Data in Malaysia (1995 – 2010). [Online]. Available: http://www.miros.gov.my/web/guest/road

[5]. M. Irfan, M.M. T. N. Baig, R. M. Hashmi, F. H. Khan, K.Shehzad and A. Ali, "Management of Location Based Advertisement Services using Spatial Triggers in Cellular Networks," in International Journal of Computer Science and Information Security (IJCSIS),vol. 6, no. 1, pp. 181-185, Jan. 2009.

[6]. S. Cojocaru, E. Barson, G. Batrinca and P. Arsenie, “GPS-GLONASS- GALILEO: A Dynamical Comparison,” in Journal of Navigation, vol. 62, no. 1,pp. 135-180, Jan. 2009.

[7]. C. J. Hegarty and E. Chatre, “Evolution of the Global Navigation Satellite System (GNSS),” in Proceedings of the IEEE, vol. 96, no. 12, pp. 1902– 1917, Dec. 2008.

[8]. C. F. Prades, L. L. Presti, and E. Falletti, “Satellite Radiolocalization from GPS to GNSS and Beyond: Novel Technologies and Applications for Civil MassMarket,” Proceedings of the IEEE, vol. 99, no. 11, pp. 1882–1904, Nov. 2011.