A

Mini Project Report On

DRIVER DROWSINESS DETECTION USING OPEN CV

Submitted in partial fulfilment of the Requirements for the award of degree of

Bachelors of Technology

In

COMPUTER SCIENCE & ENGINEERING

By

PODDUTOORI ANJANIUMA	**20681A0549**
ERASARAPU SUPRIYA	**20681A0514**
ADITYA MAMUNURI	**20681A0539**
MARATI KALPANA	**20681A0541**

Under the Esteemed Guidance of

Mr. A. V. VAMSHI KRISHNA

(Assistant professor)

Department of Computer Science & Engineering

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENC*E*

Colombo Nagar, Yeshwanthapur, Jangaon-506167, Telangana

2023-2024

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon -506167, Telangana

Department of Computer Science and Engineering

2023-2024

CERTIFICATE

This is to certify that is a Bonafide record of project work Entitled ***"Driver drowsiness detection using open CV"*** *Carried out by Anjaniuma(20681A0549), Supriya(20681A0514), Aditya(20681A0539), Kalpana(20681A0541) during academic year 2023-2024, in partial fulfillment of the requirement of the award of degree of Bachelor of Technology in computer Science & Engineering offered by Christu Jyothi Institute of Technology & Science, Yeshwanthapur, Jangaon.*

Project co-guide	**Project Guide**	
Mrs. R. DIVYAVANI	**Mr. A. V. VAMSHI KRISHNA**	**MR. M.RAMARAJU**
Assistant Professor	**Assistant Professor**	**Assistant Professor & HOD**

External Examiner

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon-506167, Telangana

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

2023-2024

DECLARATION

*We hereby declare that the document entitled "**Driver drowsiness detection using open CV Technology**" submitted to the **Christu Jyothi Institute of Technology & Science** in partial fulfillment of requirements for the award of the degree of **Bachelor of Technology** in computer Science and Engineering is a record of an original work done by us under the guidance of **Mr. A.V. VAMSHI and Mrs. R. DIVYAVANI** and this document has not been submitted to any other university for the award of any other degree.*

ANJANI UMA	**20681A0549**
SUPRIYA	**20681A0514**
ADITYA	**20681A0539**
KALPANA	**20681A0541**

ACKNOWLEDGEMENT

On the submission of our project entitled **"DRIVER DROWSINESS DETECTION USING OPEN CV"**, we would like to extend our gratitude and sincere thanks to our guide

Mr. A. V. VAMSHI KRISHNA & **Mrs. R. DIVYAVANI** Department of Computer Science and Engineering for their valuable and timely suggestions.

We would also like to extend our sincere thanks to **Mr. M. RAMARAJU**, Assistant Professor, Head of the Department, Computer Science & Engineering for his valuable suggestions and motivating guidance during our project work

We would like to thank our Director **Rev.Fr. D. VIJAYA PAUL REDDY**

for giving us the opportunity to carry out our project work.

We endow our sincere thanks to Principal **Dr. S. CHANDRASHEKAR REDDY** for his consistent cooperation and encouragement.

We are very thankful to our teachers for providing the required background during the project work. We would also like to extend our gratitude to our friends and those who are directly or indirectly helped us in completing our project work.

ANJANIUMA **20681A0549**

SUPRIYA **20681A0514**

ADITYA **20681A0539**

KALPANA **20681A0541**

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY& SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision and Mission of the Institute

Vision

To be a centre of eminence to mould young, fresh minds into challenging computer science professionals with ethical values.

Mission

1. Enrich the knowledge and wisdom with repository of books and modernized laboratory aided by dedicated resources.

2. Organize training and activities on upcoming techniques, and inter-personal skills.

3. Develop the ability to provide sustainable solutions to real world situations with collaborations.

Head of Department

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY& SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Program Educational Objectives

PEO 1 Graduates of B. Tech (CSE) are able to formulate, analyse, and solve hardware and software problems within the constraints and pursue the research.

PEO 2 Demonstrate knowledge in core areas of computer science and related engineering to comprehend engineering trade-offs to create novel products.

PEO3 Show the awareness of life-long learning needed for a successful professional career and exhibit ethical values, excellence, leadership, and social responsibilities.

Head of Department

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY& SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Program outcomes

PO No	Program Outcomes
PO1	Engineering knowledge apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis identity, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
PO3	Design/development of solutions design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tools usage Create, Select, and apply appropriate techniques resources, and modern engineering and IT tools including predictions and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society apply reasoning informed by the contextual knowledge to assess societal and environment contexts, and demonstrate the knowledge of, and need for sustainable development.
PO7	Environment and sustainability understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
P09	Individual and team work function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
PO10	Communication communicates effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations.
PO11	Project management and finance demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning recognizes the need for, and have the preparation and ability to engage and lifelong learning in the broadest context of technology change.

Head of Department

Program Specific Outcomes

PSO1 Proficiency skill:

Understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics and networking for efficient design of computer-Based systems of varying complexity.

PSO2 Problem-Solving Skills:

Apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.

PSO3 Successful Career and Entrepreneurship:

Employ modern computer language, environments and platforms in creating innovative career paths to be entrepreneur and a zest for higher studies.

Head of Department

ABSTRACT

Drowsiness is one of the major cause these days for road accidents. So in order to avoid such accidents we're developing a system which is a drowsiness alert system. In such system we recognize the driver's face and give alerts whenever the driver is drowsy and he blinks his eyes frequently as an indication that he is sleepy. So an Alarm is ringed to maintain their safety.

By using ARTIFICIAL INTELLIGENCE (AI) Technology, we're building this system. Firstly, the image of the driver is captured and it is identified using face recognition techniques and once the driver is in the vehicle and he starts the vehicle to drive, for instance if he feels drowsy there will be an Alarm/Alert so that he can wake up himself, take a break and then drive the vehicle. Computer Vision(CV) is used to design this system by taking the Eye Marks which determine the EAR(Eye Aspect Ratio) to check whether the driver is drowsy. This model can predict the 90% accuracy.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Driver drowsiness detection stands as a critical innovation in the realm of automotive safety, aimed at preventing accidents caused by driver fatigue and inattentiveness. This technology harnesses the power of computer vision, particularly through the utilization of the OpenCV (Open Source Computer Vision Library) framework, to monitor a driver's facial expressions and eye movements. By intelligently analysing these visual cues, driver drowsiness detection systems can issue timely alerts or warnings to ensure the driver remains vigilant and responsive during their journey.

The concept of driver drowsiness detection addresses a pressing concern that has persisted since the inception of the automobile. Fatigue-induced accidents account for a significant portion of road fatalities, underscoring the importance of developing advanced solutions to mitigate this risk. Traditional methods, such as audio alerts or vibration-based systems, have limitations in accurately gauging a driver's level of alertness. In contrast, computer vision-based drowsiness detection offers a more precise and adaptive approach, as it directly observes and interprets the driver's facial features.

OpenCV, a widely adopted open-source computer vision library, serves as a fundamental building block for developing driver drowsiness detection systems. It equips developers with a robust set of tools to handle image and video processing tasks, including face detection and eye tracking. Leveraging OpenCV's capabilities, engineers and researchers can devise sophisticated algorithms that analyse a driver's face in real-time, identifying crucial signs of drowsiness such as drooping eyelids or prolonged eye closure.

The core process of driver drowsiness detection involves several key stages. First, the system employs face detection algorithms to locate and isolate the driver's face within the

camera's field of view. Subsequently, the focus shifts to the eyes, utilizing techniques like Haar cascades or deep learning-based models to precisely track the movement and openness of the driver's eyelids. By continuously monitoring these parameters, the system can discern patterns indicative of drowsiness, such as prolonged eye closure or erratic blinking.

To ensure the effectiveness of drowsiness detection, it is crucial to establish a baseline of normal behaviour for each individual driver. Machine learning techniques can be employed to create personalized models that adapt to the unique characteristics of a driver's facial expressions and eye movements. By training these models with a diverse range of data, including different lighting conditions and driver appearances, the system can become more robust and capable of accurately differentiating between normal and drowsy states.

In conclusion, the integration of OpenCV within the realm of driver drowsiness detection revolutionizes road safety by harnessing the power of computer vision to prevent accidents caused by driver fatigue.

This technology represents a significant leap forward from traditional warning systems, offering a more nuanced and precise approach to identifying drowsiness cues. By capitalizing on OpenCV's capabilities and combining them with machine learning advancements, developers can create intelligent systems that enhance driver awareness and ultimately save lives on the road. As we delve deeper into the intricacies of driver drowsiness detection using OpenCV, we will explore the underlying algorithms, methodologies, and real-world applications that underscore its importance in modern automotive safety.

1.1 FACTS AND STATISTICS

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. Fatigue combined with bad infrastructure in developing countries like India is a recipe for disaster. Fatigue, in general, is very difficult to measure or observe unlike alcohol and drugs, which have clear key indicators and tests that are available easily. Probably, the best solutions to this problem are awareness about fatigue-related accidents and promoting drivers to admit fatigue 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. When there is an increased need for a job, the wages associated with it increases leading to more and more people adopting it. Such is the case for driving transport vehicles at night. Money motivates drivers to make unwise decisions like driving all night even with fatigue. This is mainly because the drivers are not themselves aware of the huge risk associated with driving when fatigued. 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.

2.LITERATURE SURVEY

A. Safe Driving By Detecting Lane Discipline and Driver Drowsiness-[Yashika Katyall, Suhas Alur,Shipra Dwivedi,(2014)]

This paper presents a real time lane detection and driver fatigue or driver drowsiness det ection system Road accidents have become all too prevalent in today's environment. The y not only inflict property damage, but they also put people's lives in danger while travel ling. Given its extent and the resulting negative effects on the economy, public health, sa fety, and the general welfare of the people, road safety is a national priority. Rough drivi ng, drunk driving, inexperience, jumping signals, and disregarding signboards are all po ssible causes of road accidents. The paper is divided into two sections. To begin, the Ho ugh Transform is used to detect lanes. Second, driver eye detection for sleepiness detecti on. As a result, the attention is mostly on the driver's weariness and adherence to lane di scipline (Katyall, Alur, & Dwivedi, 2014).

B. Monitering Driver’s Drowsiness Status at Night Based on Computer Vision-[Vidhu Valsan A,Paul P Mathai,Lerin Babu(2021)]

This research describes a real-time tiredness driving detection system that operates at nig ht. The location of facial landmarks on the driver's face is determined by employing one shape predictor and then computing eye aspect ratio, mouth opening ratio, and yawning f requency.

The values of these parameters are used to identify drowsiness.

The thresholds are established using an adaptive thresholding approach. Offline implem entation of machine learning techniques was also done. The proposed approach was test ed in both real-time and on the Face Dataset. The system's accuracy and robustness are d emonstrated by the experimental findings (Valsan, Paul, & Babu, 2021).

C. Intelligent Driver Drowsiness Detection through Fusion of Yawning and Eye Closur e [M. Omidyeganeh, A. Javadtalab, S. Shirmohammadi,(2011)]

Drowsy driving is a key component in the majority of car accidents. We describe a robu st and clever approach for detecting driver tiredness in this study, which combines eye cl osure and yawning detection methods. A camera fitted in the automobile captures the dri ver's face look in this method. The eye and mouth portions of the face are then removed and examined for symptoms of driver tiredness. Finally, during the fusion phase, the dri ver's condition is assessed, and if sleepiness is identified, a warning message is issued to the driver. Our tests show that the proposed approach is quite effective (Mandal, Li, Wa ng, & Jie, 2016).

D. Portable Prevention and Monitoring of Driver's Drowsiness Focuses to Eyelid Mov ement Using Internet of Things [Menchie Miranda, Alonica Villanueva, Mark Jomar Bu o, Reynald Merabite, Sergio Paulo Perez, John Michael Rodriguez,(2018)]

Since the number of vehicular accidents in the Philippines has been increasing year after year, this paper offers a sleepiness prevention device. Current safety measures are used t o boost driver awareness, such as the construction of standard rumble strips on highways , GPS, speed limiters, sensors, and other research that employ signal processing incorpor ated in a costly car. The system makes use of the internet of things to allow the car owne r to keep track of the driver's tiredness at all times throughout working hours. The curren t study focuses on eyelid movement, which was not covered in the prior study. This sugg ested system continually detects the driver's eyelid movements, and if sleepiness is ident ified, the gadget notifies him with a random-typed sound. It sends the report to the car o wner automatically via internet access from the web application (Miranda, Villanueva, B uo, & Merabite, 2018).

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

PCA based method which was used earlier for eye/face detection failed as it does not support night driving conditions and has a lower accuracy rate. So to overcome this drawback LBPH algoritm is used The CNN, DNN based technique used for face recognition doesn’t suppport realtime face recognition and also has a lower accuracy rate.

3.1.1 DRAWBACKS

- Few of the models don’t support night driving conditions or glasses, etc.
- Only few systems support real-time face recognition
- No existing systems including both drowsiness detection system and vehicle security system using face recognition

3.2 PROPOSED SYSTEM

The image of the driver is captured and it is identified using face recognition techniques and once the driver is in the vehicle and he starts the vehicle to drive, for instance if he feels drowsy there will be an Alarm/Alert so that he can wake up himself, take a break and then drive the vehicle. Computer Vision(CV) is used to design this system by taking the Eye Marks which determine the EAR(Eye Aspect Ratio) to check whether the driver is drowsy.

3.2.1 ADVANTAGES

1. **Real-time Monitoring:** The system operates in real time, continuously monitoring the driver's eyes and facial expressions while they are driving. This allows for prompt detection of drowsiness and quick response.

2.**Non-Intrusive:** The system does not require any additional sensors attached to the driver's body. It simply uses a camera to capture the driver's face, making it non-intrusive and comfortable for the driver.

3.**Customizable Alerts:** The system can be designed to provide customizable alerts, such as audible alarms or visual notifications. This allows drivers to receive alerts in a way that suits their preferences, without causing panic or distraction.

4.**Early Detection:** By analyzing the Eye Aspect Ratio (EAR), the system can detect signs of drowsiness before the driver's eyes completely close. This enables early detection, giving the driver a chance to take a break or regain alertness before the situation worsens.

5.**Safety Enhancement:** The primary goal of the system is to enhance driver safety. By notifying drivers of their drowsy state, the system helps prevent accidents caused by fatigue-related impairments.

6.**Adaptability:** The system can adapt to different lighting conditions, driving environments, and driver behaviors. With proper calibration and tuning, it can be effective in various situations.

7.**Reduced Accidents:** Drowsy driving is a significant factor in road accidents. This system contributes to reducing accidents by addressing one of the main causes of impaired driving.

8.**User-Friendly:** The system's simplicity and reliance on existing hardware (a camera) make it user-friendly for both drivers and vehicle manufacturers.

9.**Cost-Effective:** Compared to more complex sensor-based systems, a camera-based drowsiness detection system can be more cost-effective to implement and maintain.

10.**Integration with Vehicle Systems:** The system can be integrated with the vehicle's existing alert systems, such as lane departure warning and collision avoidance, creating a comprehensive safety ecosystem.

11.**Continuous Monitoring:** Unlike manual methods (e.g., drivers splashing water on their face), the CV-based system offers continuous and unbiased monitoring, ensuring that drivers are always monitored for drowsiness.

12.**Data Collection and Analysis:** The system can collect data over time, which can be analyzed to identify patterns of drowsiness, times of day when drivers are most vulnerable, and other insights that can lead to better driver safety practices

3.2 HARDWARE REQUIREMENT SPECIFICATION

- **Processor**: Intel Core i3.
- **RAM:** 8 GB RAM
- **Hard Disk:** 100 GB
- Web cam / In-built laptop camera

3.3 SOFTWARE REQUIREMENT SPECIFICATION

Anaconda Navigator: Anaconda navigator is a graphical consumer interface(GUI) this is blanketed within the anaconda distribution, permitting you to effortlessly release packages and control conda applications, environments and channels with out the use of commandline commands. Navigator can look for applications in anaconda. Org or the

nearby anaconda repository. It's miles to be had for windows, MacOS and Linux

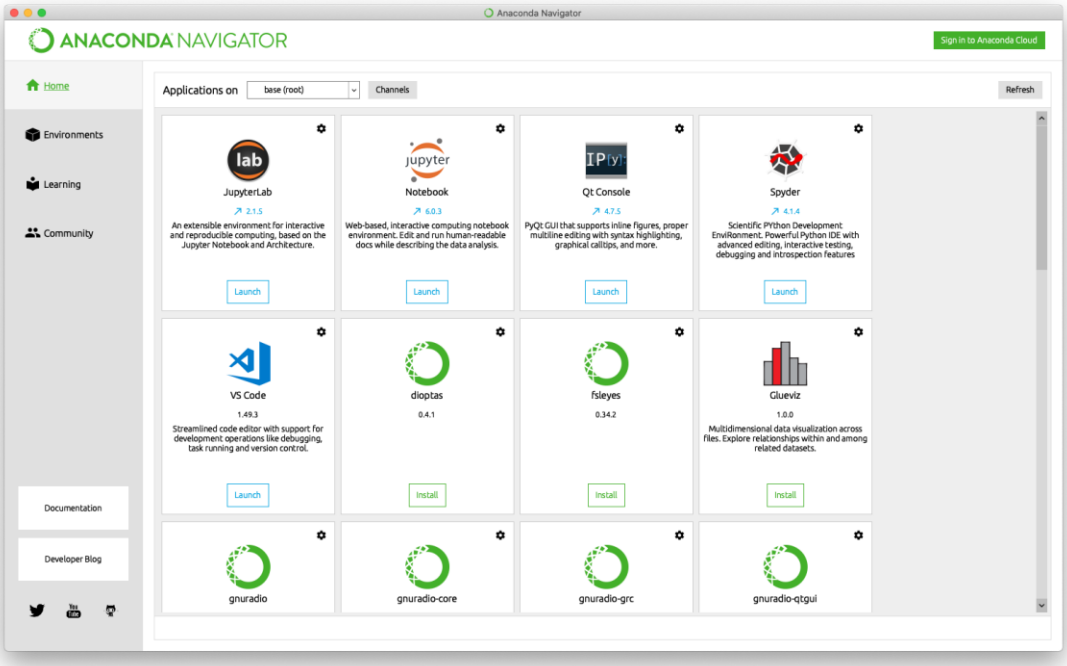

Fig 3.3: Anaconda home tab

Jupyter Notebook: The jupyter notebook expands the console-primarily based approach to interactive computing in an approximate. new route, Dispensing a bendy internetprimarily based application to capture the whole computing method: enlarging, logging and applying code, as well as speaking outcomes. The jupyter pocket book combines two elements: 1. A web application 2. Notebook documents

Python 3: Python is the basis of the program that we wrote in the system. It utilizes many of the python libraries.

Windows/ linux/ Mac OS

3.4 FUNCTIONAL REQUIREMENTS

- **D-lib:** D-lib is a toolkit for growing actual-world device gaining knowledge of and records analysis packages. It is used to become aware of the face and to mark the facial landmarks which may be used for a wide array of functions. The frontal face detector on the d-lib works properly.
- **Open CV:** Opencv is a library the usage of which we will broaden actual-time computer vision programs. It in particular used for photo processing, video seize and assessment in conjunction with capabilities like face detection and object detection. Opencv is the big open-supply library Used for laptop imaginative and prescient and also system gaining knowledge of and image processing and now it plays a prime function in actual-time operation which can be very critical in current systems. Through the use of it, you could process photographs, videos, real-time motion pictures to understand gadgets, faces, or even handwriting of a human which can be used for a big selection of functions
- **Operating System:** An operating system, or "os" is software program that interacts with the hardware and permits one-of-a-kind applications to run. ... Every laptop pc, pill, and cellular telephone consists of an operating system that gives simple functionality for the device. Common desktop jogging structures consist of home windows, Mac Os x, and Linux.

- **Numpy:** Numpy is a library that deals with massive multi-dimensional arrays and matrices. It does high-stage mathematical paintings on these mathematical systems. The numpy array is a widespread data shape for handling images, filter kernels and function factors.

- **Scipy:** Scipy is an open-source python library that is used to clear up scientific and mathematical problems. It is constructed at the numpy extension and allows the user to control and visualize statistics with a large range of excessive-level commands.

- **Imutils:** Imutils are a sequence of functions used to make fundamental photo processing functions which incorporates translation, rotation of images, resizing images, skeletonization, and displaying matplotlib pictures much less complex with openCV and each python 2.7 and python .3. It also helps in sorting contours and detecting edges.

4.SYSTEM DESIGN

4.1 Introduction to UML

The Unified Modelling Language (UML) enables a software engineer to define an analysis model using modelling notation, which is governed by a set of syntactic, semantic, and pragmatic principles. A UML system is represented by five separate views that describe the system from unique points of view. Each view is specified by the following collection of diagrams:

1. View of the User Model This view depicts the system as seen by the users. The analysis representation describes a usage situation from the standpoint of the endusers.

2. View of the Structural Model The data and functionality in this model are obtained from within the system. The static structures are represented by this model view.

3. Behavioral Model Perspective It depicts the interactions of collection between various structural elements described in the user model and structural model view, representing the dynamic of behavioral as system parts.

4. View of the Implementation Model The structural and behavioral components of the system are shown as they will be built in this view. View of the Environmental Model The structural and behavioral features of the environment in which the system is to be deployed are depicted in this view.

4.2 UML Diagrams

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions,

artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

4.2.1 Use Case Diagram

A use case diagram is used to describe a system's dynamic behavior. It contains the functionality of the system by incorporating use cases, actors, and their interactions. It simulates the jobs, services, and functionalities required by an application's system. It represents a system's high-level functionality as well as how the user interacts with the system. Following are the purposes of a use case diagram given below:

1. It gathers the system’s needs.

2. It depicts the external view of the system.

3. It recognizes the internal as well as external factors that influence the system.

4. It represents the interaction between the actors. After both actors and use cases have been identified, the relationship between the actor and the use case/system is examined

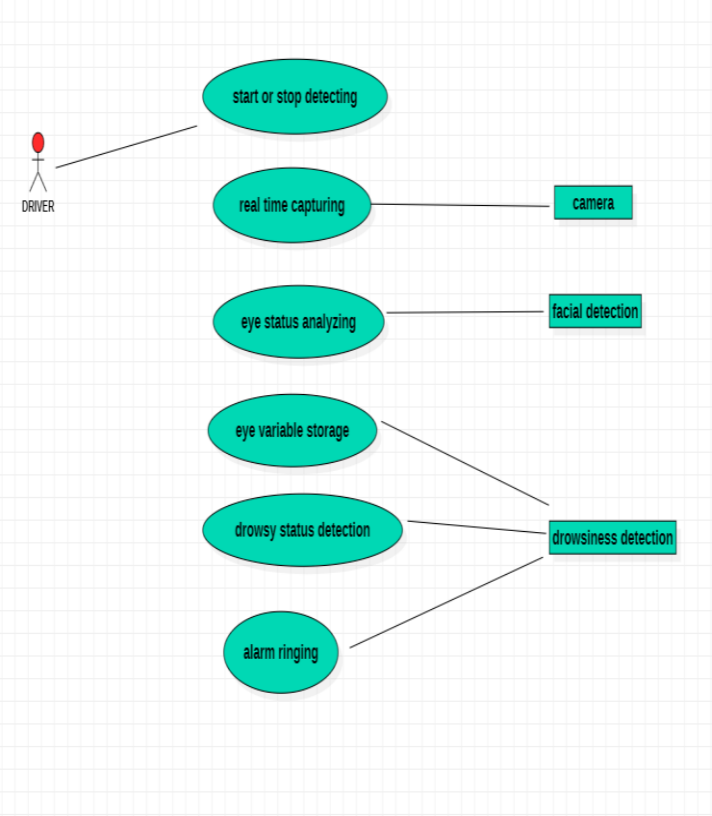

Fig 4.2.1: Use Case Diagram

4.2.2 Sequential Diagram

The sequence diagram, also known as an event diagram, depicts the flow of messages in the system. It aids in visualizing a variety of dynamic circumstances. It depicts communication between any two lifelines as a time-ordered series of events in which these lifelines participated during the run duration. The lifeline is represented by a vertical bar in UML, whereas the message flow is represented by a vertical dotted line that spans the bottom of the page. It incorporates both iterations and branching. The sequential diagram's objective is to model high-level interaction among active objects in a system.

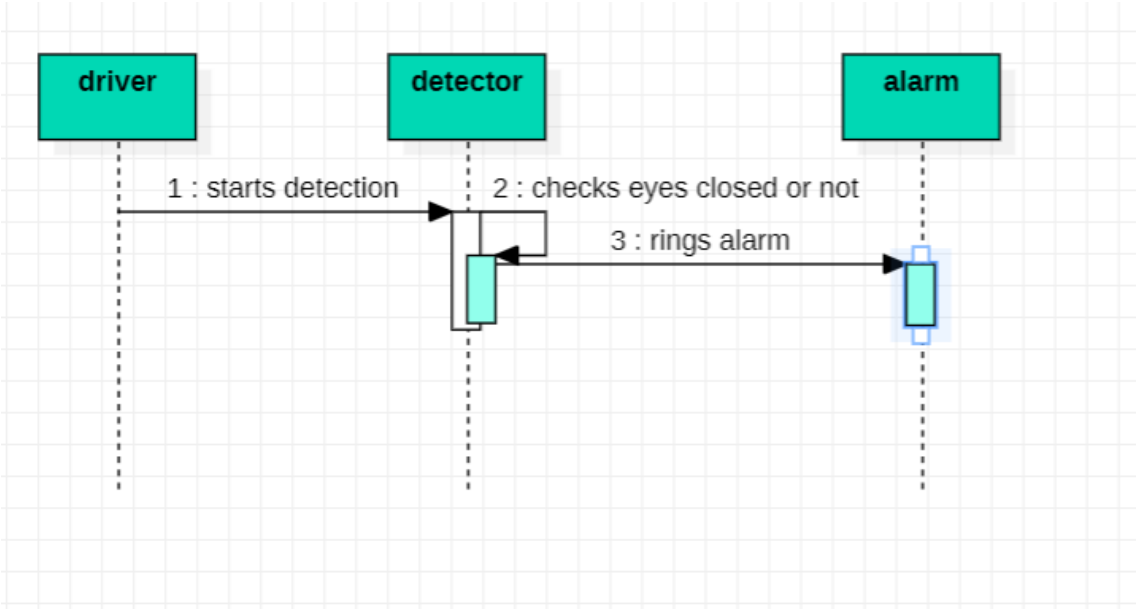

Fig 4.2.2: sequence diagram

4.2.3 Class diagram

A static view of an application is depicted in the class diagram. It reflects the many sorts of objects in the system as well as their relationships. A class is made up of objects, and it may inherit from other classes. A class diagram is used to visualize, explain, and document numerous parts of a system, as well as to create executable software code. It displays the software system's properties, classes, functions, and relationships to provide an overview. It organizes class names, characteristics, and functions into a discrete compartment to aid in program development. A structural diagram is a collection of classes, interfaces, linkages, collaborations, and constraints. Class diagrams are used to create a static view of an application. It is the only construction diagram that can be mapped with object-oriented languages. It is one of the most widely used UML diagrams. The following is the purpose of class diagrams:

1. It studies and designs an application's static view.

2. It defines a system's primary tasks.

3. It serves as the foundation for component and deployment diagrams.

4. It employs both forward and backward engineering. unwanted properties would result in a complex diagram.

5. Notes can be used by the developer to describe the aspects of a diagram as needed.

6. Before producing the final edition, the schematics should be redrawn and modified as many times as possible to ensure accuracy

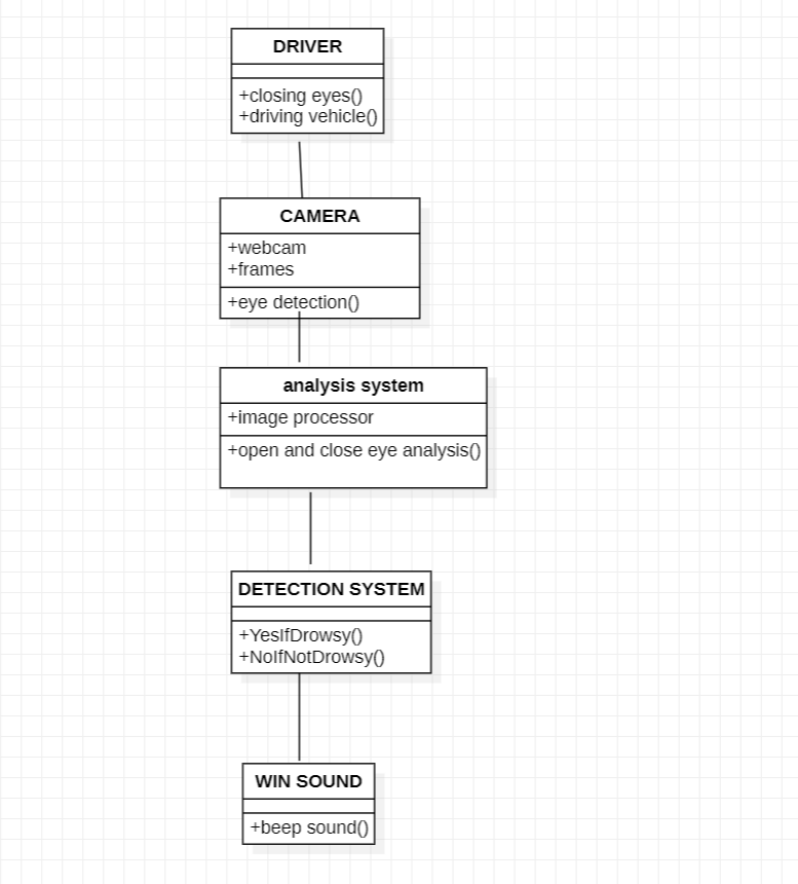

Fig 4.2.3:class diagram

4.2.4 Activity diagram

The activity diagram in UML is intended to show the flow of control inside the system rather than the implementation. It simulates simultaneous and sequential activity. The activity diagram aids in visualizing the workflow from one activity to the next. It emphasizes the state of flow and the order in which it occurs. The flow can be sequential, branching, or concurrent, and the activity diagram has forked, joined, and so on to accommodate such flows. An object-oriented flowchart is another name for it. It includes activities that are made up of a series of actions or processes that are used to model the behavioral diagram. An event is represented as an activity diagram with nodes connected by edges. They can be added to any modelling element to model the behavior of activities. It is 21 capable of simulating use cases, classes, interfaces, components, and collaborations. It is mostly used to model processes and workflows. It imagines the system's dynamic behavior and builds a runnable system that integrates forward and reverse engineering. It lacks the message component, implying that message flow is not depicted in an activity diagram. It is similar to, but not precisely the same as, a flowchart. It is used to represent the flow of many activities. Following are the rules that are to be followed for drawing an activity diagram:

1. A meaningful name should be given to each and every activity.
2. Identify all of the constraints.
3. Acknowledge the activity associations.

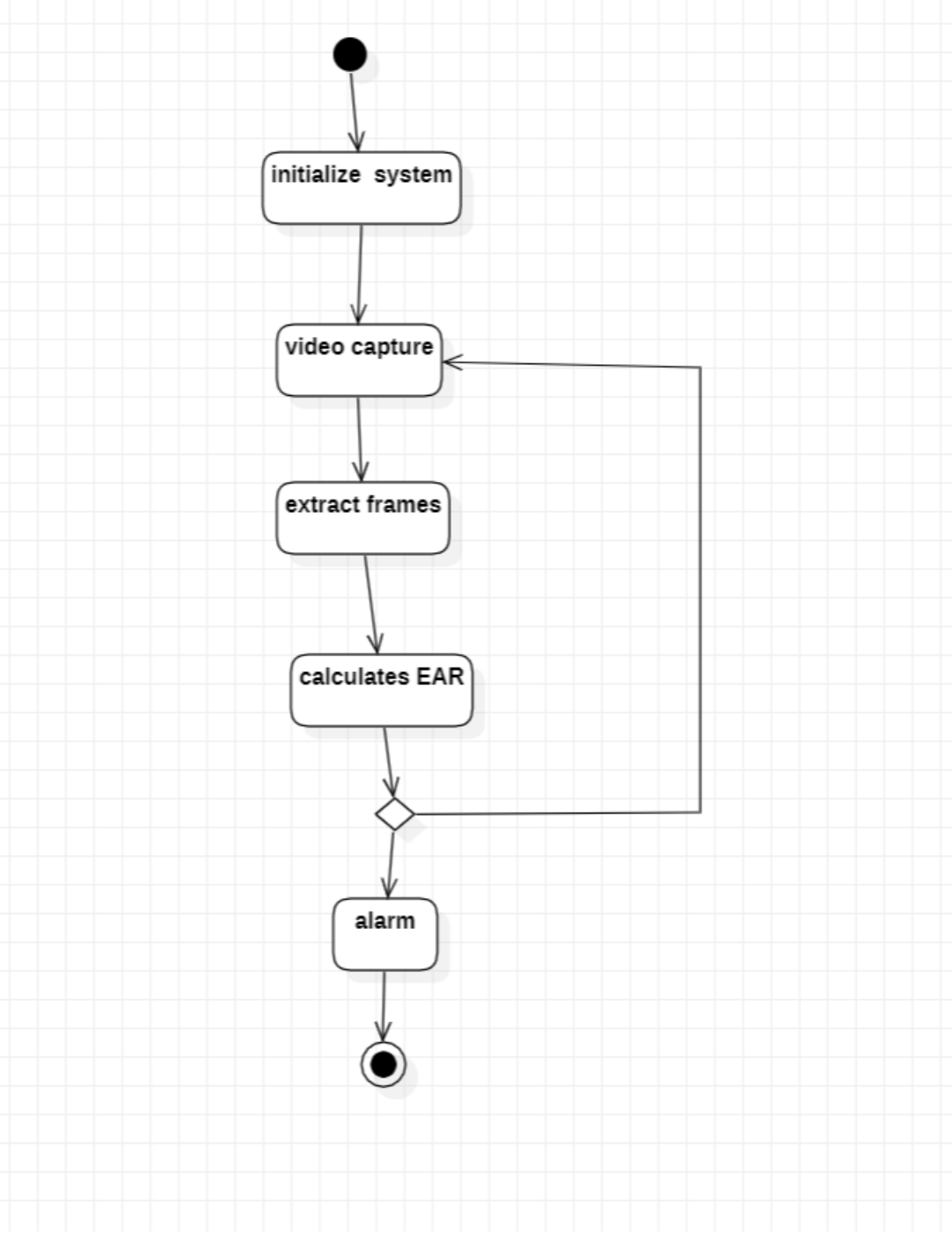

Fig 4.2.4: Activity diagram

4.3 CONTROL FLOW GRAPH

A **Control Flow Graph (CFG)** is the graphical representation of control flow or applications. Control flow graphs are mostly used in static analysis as well as compiler applications, as they can accurately represent the flow inside of a program unit. The control flow graph was originally developed by *Frances E. Allen*.

Characteristics of Control Flow Graph:

- Control flow graph is process oriented.
- Control flow graph shows all the paths that can be traversed during a program execution.
- Control flow graph is a directed graph.
- Edges in CFG portray control flow paths and the nodes in CFG portray basic blocks.

There exist 2 designated blocks in Control Flow Graph:

1. **Entry Block**

   Entry block allows the control to enter into the control flow graph.

2. **Exit Block**

   Control flow leaves through the exit block.

Hence, the control flow graph is comprised of all the building blocks involved in a flow diagram such as the start node, end node and flows between the nodes.

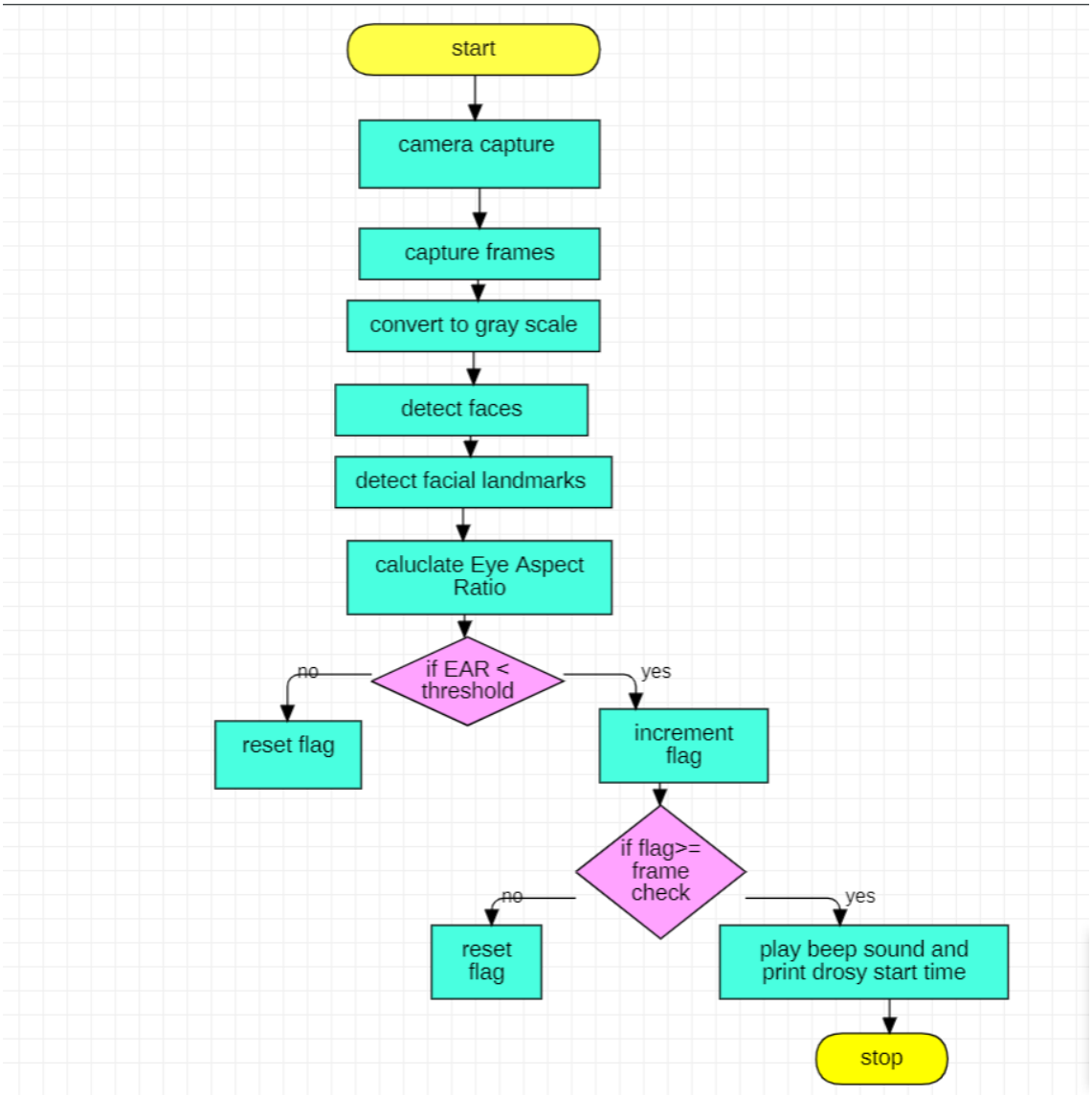

Fig 4.3: control flow diagram

5. IMPLEMENTATION

The implementation can be broken down into the following steps:

- **Frame capture**

  The system continuously captures video frames from the webcam in real-time. Each frame is a snapshot of the driver's face

- **Grayscale Conversion**

  To simplify processing, the captured frames are converted from color to grayscale. Gray scale images contain only intensity information and are easier to work with.

- **Facial landmark detection**

  Once the face is detected, a facial landmarks predictor (such as dlib's shape predictor) is used to locate key facial features, including the eyes. The predictor identifies specific points on the face that are crucial for calculating the Eye Aspect Ratio (EAR)

- **Eye Aspect Ratio (EAR) calculation**

  The EAR is a metric that quantifies the openness of the eyes. It's computed based on the distance between specific landmarks on the eyes.

- **Drowsiness Detection**

  The EAR values of both eyes are calculated, and their average is computed. If the average EAR falls below a preset threshold, it indicates that the driver's eyes are closing or partially closed. A low EAR over multiple consecutive frames signifies potential drowsiness.

- **Alert Mechanism**

  To alert the driver when drowsiness is detected, an alert mechanism is implemented. This could involve playing an alarm sound.

- **Termination:**

  The implementation runs within a loop that captures frames until the user decides to exit. After use, make sure to release the resources, such as closing the webcam feed and any open windows.

To implement the driver drowsiness detection using open cv technology in python, we need to initialize the following modules and libraries

- **D-lib:** D-lib is a toolkit for growing actual-world device gaining knowledge of and records analysis packages. It is used to become aware of the face and to mark the facial landmarks which may be used for a wide array of functions. The frontal face detector on the d-lib works properly.

- **Open CV**: Opencv is a library the usage of which we will broaden actual-time computer vision programs. It in particular used for photo processing, video seize and assessment in conjunction with capabilities like face detection and object detection. Opencv is the big open-supply library Used for laptop imaginative and prescient and also system gaining knowledge of and image processing and now it plays a prime function in actual-time operation which can be very critical in current systems. Through the use of it, you could process photographs, videos, real-time motion pictures to understand gadgets, faces, or even handwriting of a human which can be used for a big selection of functions

- **Numpy:** Numpy is a library that deals with massive multi-dimensional arrays and matrices. It does high-stage mathematical paintings on these mathematical systems. The numpy array is a widespread data shape for handling images, filter kernels and function factors.

- **Scipy:** Scipy is an open-source python library that is used to clear up scientific and mathematical problems. It is constructed at the numpy extension and allows the user to control and visualize statistics with a large range of excessive-level commands.

- **Imutils:** Imutils are a sequence of functions used to make fundamental photo processing functions which incorporates translation, rotation of images, resizing images, skeletonization, and displaying matplotlib pictures much less complex with openCV and each python 2.7 and python .3. It also helps in sorting contours and detecting edges.

- **Winsound:** It can utilize python's winsound module to provide an audio alert.

  You have to install these modules using following commands:

  1. Pip install open cv-python
  2. Pip install dlib
  3. Pip install scipy
  4. Pip install imutils

6. CODING

6.1 code for graphical user interface

```
 import tkinter as tk

 from tkinter import filedialog

from PIL import Image, ImageTk

import subprocess

def open_project_file():

    subprocess.run(["python","usingcsv.py"])

root = tk.Tk()

root.title("driver drowsiness detection")

root.geometry("4000x1000")

bg_image = Image.open("background.png")  # Replace with your image file path

bg_photo = ImageTk.PhotoImage(bg_image)

bg_label = tk.Label(root, image=bg_photo)

bg_label.place(x=0, y=0, relwidth=1, relheight=1)

button_style = {"font": ("Helvetica", 14, "bold"), "width": 20, "height": 2, "bg": "purple",
"fg": "white"}

open_button = tk.Button(root, text="Start detection", command=open_project_file,
**button_style)
```

```
open_button.pack(pady=300)
```

6.2 Code for Driver Drowsiness Detection using open cv:

```
import cv2

import dlib

import numpy as np

from scipy.spatial import distance as dist

import winsound

import time

from imutils import face_utils

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

thresh = 0.25

frame_check = 20

detect = dlib.get_frontal_face_detector()
```

```
predict = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

(lStart, lEnd) = (42, 48)

(rStart, rEnd) = (36, 42)

cap = cv2.VideoCapture(0)

flag = 0

drowsy_start_time = None

drowsy_detected = False

while True:

    ret, frame = cap.read()

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

    subjects = detect(gray, 0)

    for subject in subjects:

        shape = predict(gray, subject)

        shape = face_utils.shape_to_np(shape)

        leftEye = shape[lStart:lEnd]

        rightEye = shape[rStart:rEnd]

        leftEAR = eye_aspect_ratio(leftEye)

        rightEAR = eye_aspect_ratio(rightEye)

ear = (leftEAR + rightEAR) / 2.0
```

```
        if ear < thresh:
            flag += 1
            if flag >= frame_check:
                if not drowsy_detected:
                    drowsy_start_time = time.strftime("%Y-%m-%d %H:%M:%S")
                    drowsy_detected = True
                    winsound.Beep(500, 5000)
                    print(drowsy_start_time)
                    # Play beep sound
        else:
            flag = 0
            if drowsy_detected:
                drowsy_end_time = time.strftime("%Y-%m-%d %H:%M:%S")
                drowsy_detected = False
    cv2.imshow("Drowsiness Detection", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

7.TESTING

7.1 Test cases and test results

Test id	Test case title	Test condition	System behaviour	Expected results
T01	NSGY	Straight face, good light, with glasses	Non drowsy	Non drowsy
T02	YTGN	Tilted face, good light, no glasses	Drowsy	Drowsy
T03	YTGY	Tilted face, good light, with glasses	Drowsy	Drowsy

Note: Testing is performed manually

Table 1: Test cases

TEST RESULTS:

1.UNIT TESTING:

Unit testing focuses on testing individual units or functions in isolation to ensure they work as expected.

Unit Test 1: EYE ASPECT RATIO FUNCTION

Input: Pass known eye landmark coordinates to the `eye aspect ratio` function.

Expected Output: Verify that the function correctly calculates the eye aspect ratio.

Unit Test 2: DLIB FACE DETECTION

Input: Provide a sample grayscale image containing a face.

Expected Output: Ensure that the DLIB face detection (`detect`) function correctly identifies the face in the Image.

Unit Test 3: DLIB LANDMARK PREDICTION

Input: A detected face region (rectangle) and a grayscale image.

Expected Output: Confirm that the DLIB landmark prediction (`predict`) function correctly predicts facial landmarks.

Unit Test 4: DROWSINESS DETECTION LOGIC

Input: Mock eye aspect ratios and frame check values.

Expected Output: Test the drowsiness detection logic by simulating various scenarios of eye openness and closure. Ensure that it correctly detects drowsiness based on the threshold and frame check.

2. INTEGRATION TESTING:

Integration testing checks how different parts of the system work together.

Integration Test 1: **CAMERA VIDEO CAPTURE**

Input: Run the code with a camera.

Expected Output: Ensure that the code interfaces with the camera correctly and captures video frames without errors.

Integration Test 2: DLIB FACE DETECTION AND LANDMARK PREDICTION

Input: Feed the code with video frames containing faces.

Expected Output: Verify that the face detection and landmark prediction work together seamlessly to locate and predict landmarks on faces in the video feed.

Integration Test 3: DROWSY DETECTION AND

Input: Simulate drowsiness situations by manipulating eye aspect ratios.

Expected Output: Confirm that the code detects drowsiness, triggers the alert (beep sound), and records the start and end times of drowsiness when applicable.

3. System Testing:

System testing evaluates the overall behavior and functionality of the entire system.

System Test 1: Full System Execution

Input: Execute the entire code with a camera feed, making sure the "shape_predictor_68_face_landmarks.dat" file is available.

Expected Output: Verify that the code runs smoothly, detects drowsiness, and triggers alerts as needed. Ensure that the code handles potential exceptions and user interactions (e.g., quitting with 'q') correctly.

System Test 2: Handling Exceptions

Input: Introduce exceptions by removing the shape predictor file, denying camera access, or using an invalid threshold value.

Expected Output: Confirm that the code handles exceptions gracefully without crashing and provides informative error messages.

System Test 3: User Interaction

Input: Interact with the code by pressing 'q' to quit or adjusting the frame check parameter during runtime.

Expected Output: Ensure that the code responds appropriately to user input and exits cleanly when required.

System Test 4: Real-world Scenarios

Input: Run the code in various real-world scenarios, such as different lighting conditions and with individuals exhibiting drowsiness.

Expected Output: Evaluate the code's performance and accuracy in real-world conditions, ensuring that it effectively detects drowsiness.

8.OUTPUT SCREENS

In this system we have an graphical user interface (GUI) with a button “start detection”, by clicking that button the system starts detecting the eyes of the driver whether he is drowsy or not?

Fig 8.1: graphical user interface

The output of drowsiness detection will be an alarm sound if driver closes eyes for more than 5 seconds

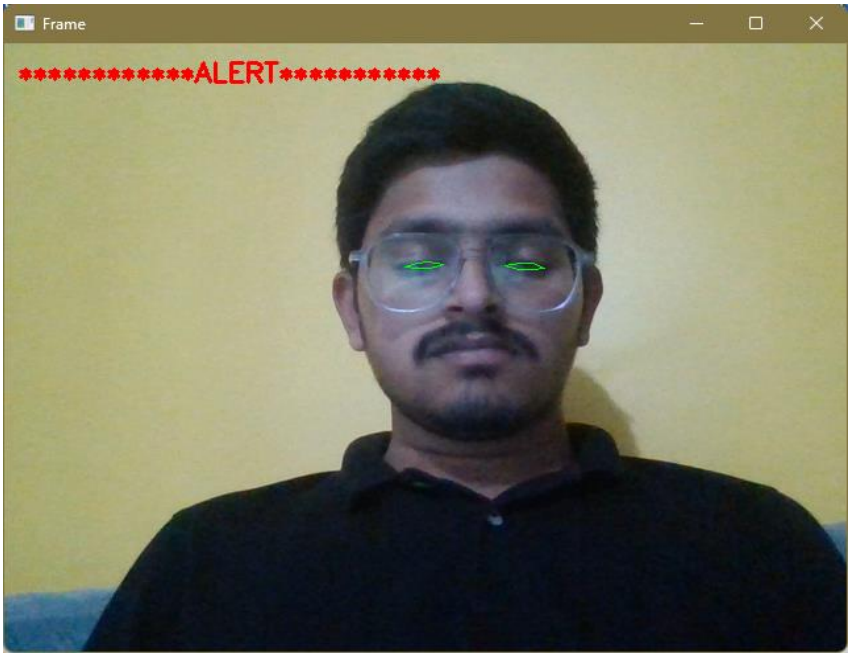

Fig 8.2: CV detecting the EYE with Spectacles

Fig 8.3: CV detecting EYE without spectacles

The output of drowsiness detection will be an alarm sound if driver closes eyes for more than 5 seconds

9.CONCLUSION

We have measured the driver's safety parameters, firstly in drowsiness detection model we made an alert system which can alert the driver whenever he feels drowsy for more than 3-4 seconds he'll be alarmed and can stay awake or take a break. Whereas in the second model, face recognition provides security to our vehicle by detecting the driver's face and providing access. The drowsiness detection system can be implemented in every vehicle such that we can prevent road accidents and decrease the death ratio which are caused due to drowsiness and the face recognition system is very helpful to maintain the security of the vehicle preventing vehicle thefts.

As AI techniques are growing vastly, we can make systems more intelligent to understand the requirements of the hour. We can introduce various models and use different types of algorithms to get the best results. Road accidents are common in countries like India. Due to small negligence there's a huge loss to the lives of the human. By adapting such systems, we can try to control the road accidents and also the security of the vehicle can be maintain by taking the alert and security systems into consideration

10.FURTHER ENHANCEMENT

Driver drowsiness detection using OpenCV is a great application of computer vision technology to improve road safety. Here are some further enhancements and recommendations to improve the accuracy and effectiveness of your drowsiness detection system:

1. Multi-modal Approach: Consider using multiple data sources to enhance accuracy. Combine facial features analysis with steering wheel movement, lane departure detection, and even physiological signals like heart rate or EEG readings for a more comprehensive drowsiness detection system.

2. **Deep Learning Models:** While OpenCV provides many useful tools, integrating deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can significantly improve the accuracy of facial feature extraction and analysis. You can use pre-trained models like OpenCV's DNN module or models from popular deep learning frameworks like TensorFlow and PyTorch.

3. **Dataset Diversity:** Train your models on diverse datasets that include a wide range of driver ethnicities, ages, and lighting conditions. This will improve the model's ability to generalize to various real-world scenarios.

4. **Real-time Eye Tracking:** Implement real-time eye tracking to detect changes in gaze direction and eyelid closure. This can provide more accurate cues for drowsiness detection.

5. **Attention Monitoring:** Consider implementing attention monitoring algorithms that track the driver's focus on the road. Sudden shifts in attention away from the road can indicate drowsiness or distraction.

6. **Data Augmentation:** Augment your training data by applying transformations like rotation, scaling, and color variations. This helps the model become more robust to different image conditions.

7. **Fusion of Features:** Combine features from different sources (e.g., facial landmarks, eye aspect ratio, yawning detection) using techniques like feature fusion or feature selection to improve the overall accuracy of your detection system.

8**. Real-world Testing:** Collect data in real-world driving scenarios to ensure that your system works effectively under various lighting conditions, weather conditions, and driving contexts.

9. **Online Learning:** Implement techniques for online learning or incremental learning. This allows your model to adapt to individual driver patterns over time, improving detection accuracy as it learns from the driver's behaviour.

10**. False Positive Reduction:** Implement mechanisms to reduce false positives. For example, use temporal analysis to distinguish between momentary eye closure and prolonged drowsiness.

11. **User Interface and Alerts:** Develop a user-friendly interface that displays the drowsiness level and provides timely alerts to the driver, such as audible alarms or vibrations in the steering wheel.

12. **Privacy Considerations**: Ensure that the system respects the driver's privacy. Use techniques like face blurring or anonymization if necessary, and make sure that no sensitive data is stored without consent.

13. **Energy Efficiency**: Optimize the algorithms for energy efficiency, especially if the system is intended for use in resource-constrained environments like mobile devices or embedded systems.

14. **Continuous Improvement**: Regularly update and fine-tune your model as you collect more real-world data and gain insights into its performance. This iterative process can lead to continuous improvements in accuracy and reliability.

In the future we can enhance these models by adding zoom in features, where the system can predict at a high rate accuracy and it will be more efficient. With the update of technology and AI techniques we can completely upgrade our system such that cameras will be only able to capture the eyes of the driver instead of complete facial expression.. Apart from this we can add up more features which are related to driver’s safety such as seat belt verification, security related purposes and also vehicle maintenance system can be added to it. In future we can take many parameters and make our system very more accurate and provide safety measures to our drivers. We can also detect the mouth region of the driver and use it to calculate the Mouth Aspect Ratio(MAR) a use it to detect frequent yawning and alert the driver based on a threshold value of the mouth aspect ratio. This would add a new dimension to the project and help us detect drowsiness with a higher accuracy. We also can upgrade the cameras and use night vision cameras to detect accurately in low light conditions

11.REFERENCES

[1] M.Omidyeganeh, A.Javadtalab,S . Shirmohammadi , “Intelligent Driver Drowsiness Detection through Fusion of Yawning and Eye Closure”,IEEE International Conference on Virtual Environmnets , Human-Computer Interfaces, And Measurements Systems (2011)

[2] Menchie Miranda , Alonica Villanueva , Mark Jomar Buo , Reynald Merabite , Sergio Paulo Perez , John Michael Rodriguez , “Portable Prevention and Monitoring of Driver’s Drowsiness Focuses to Eyelid Movement Using Internet of Things”,IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (2018)

[3] Yashika Katyall , Suhas Alur , Shipra Dwivedi , “Safe Driving By Detecting Lane

[4] Discipline and Driver Drowsiness”, IEEE International Conference on Advanced Communications, Control and Computing Technologies (2014)

[5] Bappaditya Mandal , Liyuan Li , Gang Sam Wang and Jie Lin, “Towards Detection of Bus Driver Fatigue Based on Robust Visual Analysis of Eye State, IEEE Transactions on Intelligent Transportation Systems(2016)

[6] Vidhu Valsan A , Paul P Mathai , Lerin Babu “Monitering Driver’s Drowsiness Status

[7] Night Based on Computer Vision” International Conference on Computing, Communication, and Intelligent Systems ICCCIS(2021)