Drowsiness Detection System

CSE 445 CAPSTONE PROJECT-II REPORT

Submitted in partial fulfillment of the degree of

B. Tech (Computer Science and Engineering)

Under the Guidance of

Dr. Makul Mahajan

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

Submitted By:

Abhishek Gupta (11908904)

Deepak Kumar Sha (11904664)

Chirag Khemchandani (11905174)

STUDENT’S DECLARATION

To whom so ever, it may concern

We, **Abhishek Gupta, Chirag Khemchandani, Deepak Kumar Sha** hereby declare that the work done by us on “**Drowsiness Detection System”** from Jan 2023 to May 2023, is a record of original work for the partial fulfilment of the requirements for the award of B.tech degree in Computer Science and Engineering, Capstone Project-II.

Team Members:

Name: **Abhishek Gupta**

Registration No.: 11908904

Name: **Chirag Khemchandani**

Registration No.: 11905174

Name: **Deepak Kumar Sha**

Registration No.: 11904664

Place: Jalandhar, Punjab

Dated:

BONAFIDE CERTIFICATE

This is to certify that the project entitled **"Drowsiness Detection System"** submitted by **"Abhishek Gupta (11908904), Deepak Kumar Sha (11904664), Chirag Khemchandani (11905174)"** work for the partial fulfilment of the requirements for the award of B.tech degree in Computer Science and Engineering, is a record of bonafide work carried out by him/her under my supervision during the period from Jan 2023 to May 2023, as per the LPU code of academic and research ethics.

Dr. Makul Mahajan
ID: 14575
School of Computer Science and Engineering

Place: Jalandhar, Punjab

Dated:

TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering (SCSE)

Program : P132::B.Tech. (Computer Science and Engineering)

COURSE CODE : CSE445 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** CSERGC0270

Supervisor Name : Dr. Makul Mahajan **UID :** 14575 **Designation :** Associate Professor

Qualification : ________________________ **Research Experience :** ________________________

SR.NO.	NAME OF STUDENT	Prov. Regd. No.	BATCH	SECTION	CONTACT NUMBER
1	Mohamed Tauheed	11914853	2019	K19EX	8259801382
2	Abhishek Gupta	11908904	2019	K19DM	7047857857
3	Chirag Khemchandani	11905174	2019	K19AP	8989422972
4	Deepak Kumar Sha	11904664	2019	K19RC	8967025875

SPECIALIZATION AREA : Programming-II **Supervisor Signature:** ____________________

PROPOSED TOPIC : Drowsiness detection system

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	**Parameter**	**Rating (out of 10)**
1	Project Novelty: Potential of the project to create new knowledge	7.41
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.41
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.41
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	8.00
5	Social Applicability: Project work intends to solve a practical problem.	8.00
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	7.50

PAC Committee Members		
PAC Member (HOD/Chairperson) Name: Raj Karan Singh	UID: 14307	Recommended (Y/N): Yes
PAC Member (Allied) Name: Vikas Verma	UID: 11361	Recommended (Y/N): Yes
PAC Member 3 Name: Dr. Makul Mahajan	UID: 14575	Recommended (Y/N): Yes

Final Topic Approved by PAC: **Drowsiness detection system**

Overall Remarks: Approved

PAC CHAIRPERSON Name: 25708::Dr.Rachit Garg **Approval Date:** 04 Apr 2023

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Makul Mahajan and the Researcher community who have carved our idea and helped us throughout my research Journey. We are also thankful for the ICCS to provide such an appropriate amount of research paper to analyse the architecture and its outcome. And express gratitude to our friends who helped collaborate with each other to produce the completion of this project journey, where they all exchanged their own interesting ideas, thoughts and made it possible to complete the project with all accurate information.

This project gave us the opportunity to explore the Python programming and was a great chance for learning and building the professional development career in python. Therefore, we consider ourselves as a very lucky individual as we were provided with an opportunity to be a part of it.

At last our thanks to our parents and friends who supported and motivated us on each single obstacle of out project life cycle, they always appreciated out hard work and inspired us by showing approach in best way possible

We perceive this opportunity as a big milestone in our career development. We will strive to use gained skills and knowledge in the best possible way, and we will continue to work on their improvement, in order to attain desired career objectives.

TABLE OF CONTENT

1. Introduction

1.1 Understanding the Problem Statement

Road safety is a crucial concern for public health, as it can result in injuries and deaths. The Highways Research and Development Institute stated that “The road accidents caused 1,53,972 fatalities and injured 3,84,448 individuals in 2021”. Drowsiness is a major contributor to numerous car accidents annually, since a driver who is feeling sleepy may not be capable of preventing or evading collisions while driving. Drowsiness refers to the condition of feeling excessively tired or sleepy during the daytime, which can cause forgetfulness and a tendency to fall asleep at inappropriate times. While drowsiness may be a temporary state, its consequences can be significantly detrimental. Fatigue, which diminishes alertness and concentration, is a frequent contributor to drowsiness. Driving for extended periods without sufficient rest or operating a vehicle while feeling tired can cause drivers to feel drowsy. The main concern in these situations is the driver's diminished ability to focus, leading to delayed responses to events while driving. Fortunately, researchers have recommended various techniques for detecting early signs of drowsiness in drivers and issuing warnings to prevent potential accidents. Drivers who are feeling drowsy tend to exhibit a range of symptoms, such as frequent yawning, frequent closure of the eyes, and driving in an irregular manner. Lately, a significant amount of research has been dedicated to developing methods to detect driver drowsiness. To decrease the frequency of accidents, researchers suggest implementing several techniques to quickly identify signs of drowsiness.

The drowsiness detection system works by monitoring the driver's physiological signals and detecting changes that indicate drowsiness. Once the system detects signs of drowsiness, it can issue a warning to the driver to take a break or stop driving. The warning can be in the form of an alarm, a message displayed on the dashboard, or a vibration in the seat or steering wheel. The warning should be strong enough to get the driver's attention and prompt them to take action.

One of the key advantages of a drowsiness detection system is that it can work in real-time. This means that it can detect signs of drowsiness as they occur, and issue warnings before an accident happens. Real-time detection is essential for preventing accidents caused by drowsy driving, as it allows drivers to take action before it's too late.

To create a reliable drowsiness detection system, careful planning and testing are necessary. It should be accurate, easy to use, and tested in various driving situations. A drowsiness detection system is crucial for preventing accidents caused by drowsy driving. It should detect drowsiness in real-time, warn the driver, and collect data for analysis. Developing such a system can reduce accidents caused by drowsy driving and improve road safety.

1.2 Objective of the Project

The primary focus of this project is to develop a reliable, precise, and efficient system that can analyse physiological signals such as eye movements etc... in real-time using various sensors and algorithms. The system must then alert the driver or operator using visual, auditory, or tactile cues such as sounds, vibrations, or flashing lights. The goal is to ensure that the system is accurate and reliable under different conditions and contexts and is practical and user-friendly, easy to install and operate, and adaptable to the individual's physiological characteristics and habits.

Developing a drowsiness detection system poses several challenges, including ensuring that the system is accurate and reliable under different conditions and contexts. For instance, the system should be able to identify when a driver is feeling sleepy, regardless of whether it's daytime or night-time, and despite different weather conditions. Additionally, it should be able to distinguish between drowsiness and other factors that may affect driving, such as alcohol or drug use. Another challenge in developing a drowsiness detection system is making it usable and convenient. The system should be simple to set up and use, without impeding the driver or operator's capacity to execute their responsibilities. It should also be adaptable to the individual's physiological characteristics and habits, such as their sleep patterns and driving style.

There are various drowsiness detection systems available in the market, including wearable devices and in-car systems, that use different sensors and algorithms to detect drowsiness and alert the driver or operator. These systems have different approaches, such as machine learning algorithms to analyze physiological signals and predict drowsiness, or driver monitoring cameras to monitor facial features and detect changes in the driver's gaze, head position, and eyelid movement. Drowsiness detection systems have numerous applications in different sectors, including aviation, healthcare, and manufacturing. Pilots and air traffic controllers could use these systems to avoid errors caused by fatigue, while healthcare professionals and heavy equipment operators could rely on them to prevent accidents resulting from drowsiness. In summary, these systems are useful in enhancing safety in various fields.

In conclusion, the primary objective of this project is to increase safety and reduce accidents resulting from driver fatigue or drowsiness. With the continuous development and greater availability of advanced technology, these systems may eventually become a standard feature in various industries and vehicles. This will help protect the health and safety of workers and the public.

1.3 Description of the project

The Drowsiness Detection System project has been designed with the primary objective of preventing accidents and promoting safety, particularly in situations where individuals are driving or operating heavy machinery while feeling tired or sleepy. The project aims to achieve this goal by providing an advanced warning mechanism that can notify the driver or operator to take a break or rest before any unfortunate incidents occur. The system can use physiological signals to identify signs of drowsiness and then notify the driver or operator by sounding alarms.

One of the key challenges in developing such a system is ensuring that it is dependable and precise in diverse conditions and settings. The system must be capable of detecting drowsiness during both day and night driving and in various weather conditions. Additionally, it needs to be able to distinguish between drowsiness and other factors that may influence driving, such as drug or alcohol consumption. To overcome these challenges, researchers and developers use various sensors and algorithms to analyse physiological signals and detect patterns of drowsiness.

The Drowsiness Detection System project also aims to create a practical and user-friendly system that does not interfere with the driver's task. The system should be easy to install and use while adapting to the driver's habits and physiological characteristics, such as their sleep patterns and driving style. The benefits of implementing drowsiness detection systems are numerous and extend beyond just preventing accidents. These systems can be used in various industries, including aviation, healthcare, and manufacturing, to enhance safety and prevent errors that may arise due to fatigue. For instance, pilots and air traffic controllers could utilize drowsiness detection systems to prevent errors that may occur due to fatigue. Similarly, healthcare workers and operators of heavy machinery could also use these systems to avert accidents that may occur due to drowsiness.

In conclusion, the Drowsiness Detection System project is a crucial step towards enhancing safety and preventing accidents caused by fatigue and sleepiness. The project's primary objective is to provide an advanced warning mechanism that can notify the driver or operator to take a break or rest before any unfortunate incidents occur. With the progress of technology and wider availability, these systems could become a typical feature in various industries and vehicles, thereby safeguarding the health and safety of workers and the general public. By preventing accidents caused by drowsiness, the Drowsiness Detection System project contributes to creating a safer and more sustainable world.

1.3.1 Model Overview

To achieved the desired outcome of the plan, a set of models is required. These models play a vital role in the framework, acting as a means of accomplishing the plan's objective. Essentially, these models are a critical component that enables the plan to succeed.

One of these models involves the integration of algorithms that can detect instances of nodding or closing of the eyes. This model comprises a series of distinct processes that are designed to identify specific signals of tiredness or drowsiness. The algorithms work together to provide a more comprehensive assessment of the user's level of attentiveness or consciousness.

In summary, the success of the plan relied on a set of models that serve as a crucial component of the framework. One such model involves the integration of algorithms designed to detect tiredness or drowsiness through specific processes that work together to evaluate the user's level of consciousness.

1.3.2 Drowsiness detection Process

The process of detecting drowsiness in individuals involves using an algorithm that relies on identifying specific features of the eyes. These features have been pinpointed by utilizing the dlib library. The algorithm analyses the Eye Aspect Ratio (EAR) of each eye, which is a measure of how open or closed the eye is. By examining the EAR of both eyes and computing their average, the algorithm can determine whether the individual is drowsy or not.

To make this determination, the algorithm compares the computed average EAR to a predefined threshold. If the average EAR remains below this threshold for a continuous period of time, typically several frames, the algorithm will sound an alert to the individual, notifying them of their drowsiness.

In summary, the algorithm for detecting drowsiness is based on analysing the EAR of both eyes, and if the average EAR remains below a specific threshold for a continuous period of time, the algorithm assumes the person is drowsy and alerts them. This algorithm is made possible through the use of the dlib library, which helps to identify the landmarks of the eyes necessary for this analysis.

The process of implementing the model involves several steps. Firstly, the model utilizes a computer vision library called OpenCV to acquire a sequence of images or frames from a webcam. These frames are then subjected to a har-cascade algorithm to detect specific patterns or features. If any of the algorithms are triggered, an alert is raised and a warning sound is played, signalling to the user that there is an issue that needs attention.

In simpler terms, the model uses a computer program to capture images from a webcam, and then it analyses those images to detect certain patterns. If anything, suspicious is detected, the user will be alerted with a warning sound. This process is possible thanks to the use of specialized software that can detect and identify specific visual features.

The following diagram shows an overview of the model:

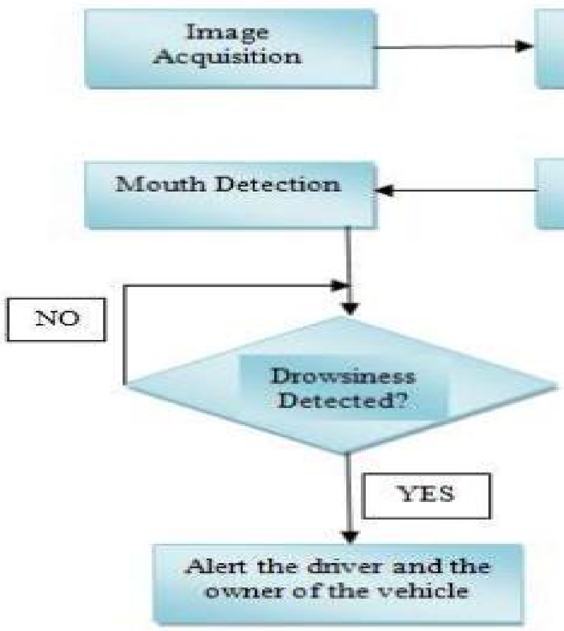

Fig.1. Overview of Model

The diagram depicts a process flow that involves analysing a video input from a webcam to determine if the user is exhibiting signs of drowsiness, closed eyes, or nodding their head. The analysis process comprises three stages, each of which performs a specific task:

- Firstly, the video feed is fed into a facial landmark detector that recognizes key facial features.
- Then, algorithms that are responsible for detecting signs of drowsiness, closed eyes, or head nodding use the data obtained from the facial landmarks to determine whether the user is displaying any of these signs.
- Finally, if any of these signs are detected, a warning sound is played to alert the user to take appropriate action.

In summary, the process flow demonstrates a technique for analysing a webcam feed to detect whether the user is exhibiting signs of drowsiness, closed eyes, or nodding head, and if so, alerting them through an audible warning.

This model uses a video stream to detect if a person is drowsy or not. It works in two steps. First, the face detection module locates the person's face in the image and extracts specific points on the face using the dlib library. This information is used to monitor the person's eye activity in the drowsiness detection module. If the person's eyes are closed for several consecutive frames, it signals that they may be drowsy, and an alarm sounds to alert the person.

Together, these modules provide a comprehensive approach to detecting drowsiness in real-time video streams. The model's objective is to keep people alert and focused on the task at hand, making it useful for activities like driving or operating machinery.

The main aim is to create a system that can effectively identify when someone is feeling drowsy or sleepy, especially when they are performing tasks that require a high level of attention, such as driving or operating heavy machinery. The system should have the ability to track the individual's behaviour and activities in real-time, continuously analysing their actions to detect signs of fatigue. If it senses that the person is becoming drowsy, it should issue immediate alerts to prevent accidents and enhance safety.

In essence, the project aims to develop an innovative solution that can help reduce the number of accidents caused by drowsy individuals, promoting safer work environments and transportation systems.

The accomplishment of this objective will entail a series of phases in the project. These stages will be crucial in ensuring that the desired outcome is achieved. Each step will have its own set of tasks that will need to be completed before proceeding to the next stage. The stages will likely be planned in a sequential manner, with each step building upon the previous one to ultimately achieve the final goal.

These steps include:

- **Research:** To perform a comprehensive investigation on the identification of drowsiness, this involves examining various sources of information such as research papers, articles, and other literature on the topic. This investigation will include evaluating different methods used for detecting drowsiness, including algorithms, sensors, and alert systems that can help detect and prevent sleepiness. The goal is to gain a better understanding of the different techniques and technologies available to identify drowsiness, and to assess their effectiveness and potential applications in real-world scenarios.

- **Algorithm development:** The task involves creating and verifying detection algorithms that are precise and dependable in their ability to analyse information gathered from specific sensors. The objective is to identify indications of sleepiness accurately. This process will likely require extensive testing to ensure that the algorithms can provide trustworthy results consistently. Ultimately, the goal is to develop a system that can detect drowsiness with high accuracy to promote safety and prevent accidents.
- **Alerting mechanism:** The goal is to create an efficient alert system that can notify people when it's time for them to take a break or rest. This can be achieved by using both visual and auditory cues such as flashing lights or alarms. The intention behind this mechanism is to prevent people from overworking themselves or becoming exhausted, which could have negative effects on their health and wellbeing. By having a reliable alert system in place, individuals can take regular breaks, which can enhance their productivity, reduce stress levels, and promote a healthy work-life balance.
- **Testing and validation:** The process of verifying and checking the accuracy and efficiency of a system by subjecting it to actual conditions and environments is called testing and validation. This can be done through the use of driving simulators or by deploying the system in real-world situations, such as actual vehicles on the road. The objective of this process is to ensure that the system can perform its intended functions correctly and consistently, even when faced with unpredictable and dynamic circumstances. By testing and validating a system in real-world environments, potential errors and issues can be identified and resolved, increasing the system's reliability and safety.

1.4 Scope of the project

The project's objectives involve designing a user-friendly interface that empowers people to track their alertness levels and take necessary measures to stay alert, such as taking a break or resting. Additionally, the system might include functionalities that enable supervisors or managers to monitor the alertness levels of their employees and take appropriate action if needed.

To put it simply, the project's goal is to create an interface that individuals can use to monitor their alertness levels and to provide managers with the tools they need to keep an eye on their employees' alertness levels and take action when necessary. The system may have different features tailored to both individual users and managers or supervisors to ensure that everyone stays safe and alert.

The project of creating a drowsiness detection system is vast and has the potential to be useful in multiple industries. In environments where fatigue and sleepiness can pose a safety threat, the system can be implemented. The primary goal of this project is to design a technology that can accurately identify the moment when a person starts feeling drowsy or fatigued, and then alert them before an accident occurs.

The drowsiness detection system will use advanced sensors and algorithms to monitor the physiological and behavioural patterns of the person, which will enable the system to detect the signs of drowsiness accurately. The technology can be utilized in various sectors such as transportation, healthcare, and manufacturing, where employees work for extended hours and face the risk of sleep deprivation.

This project is essential as it has the potential to prevent accidents caused by drowsiness and fatigue, which can be detrimental to both the individual and the company. By alerting the person before they fall asleep, the system can help them take appropriate measures to prevent any mishap. Overall, the drowsiness detection system can improve safety and prevent injuries in various industries, making it a crucial and worthwhile project.

Drowsiness detection systems have practical applications in several industries, including aviation, healthcare, and manufacturing. For example, pilots and air traffic controllers can utilize drowsiness detection systems to prevent fatigue-related errors, while healthcare workers and heavy machinery operators can use them to avoid accidents caused by drowsiness.

The objective of the project is not only to create a system, but also to make it user-friendly and easy to install and operate. In addition, the system should be able to adjust to an individual's unique physical and behavioural characteristics, such as their sleep patterns and driving style. This means that the system should be designed in a way that allows it to adapt and accommodate the specific needs of each user, making their experience with it as comfortable and efficient as possible. Overall, the project aims to create a personalized and user-centric solution.

To put it differently, the goal of a project focused on drowsiness detection is to increase safety, lower the number of accidents caused by drowsiness across different industries. Additionally, the aim is to create a system that is precise, dependable, user-friendly, and flexible enough to be used in various situations and adapted to individual needs.

In general, the project's scope is to create a comprehensive system that can precisely identify drowsiness and produce timely warnings to prevent accidents and enhance safety in safety-sensitive settings.

1.5 Literature Review

Saito et al. suggested a drowsiness system that operates by examining movements of a driver's eyelids, driver’s wheel etc... DDD intervenes by partially taking over control of the vehicle when it deviates from its lane, allowing the driver to reposition it. If the driver fails to regain control within a certain time frame, it assumes that the driver is incapacitated and takes control of the car to park it safely. During the system's operation, data were mainly collected, and the driver's status was determined using mathematical algorithms and fixation patterns.

The study reported achieving up to 100% accuracy in controlling the vehicle under specific driving conditions.

Dasgupta et al. created a drowsiness model that detect and warn driver in time. It uses smartphones and incorporates audio cues and PERCLOS to identify drowsiness. They created their dataset, called the Invedrifac dataset, and used three levels of validation to evaluate their system. In the first level, the system calculates PERCLOS features from the front camera of mobile. If the PERCLOS value exceeds a certain threshold, it proceeds to Phase 2, where the driver is prompted to say their full name. In case the first two stages are identified by the system as an indication that the driver is falling asleep, driver is then asked to respond within a time frame of 10 seconds. If the driver fails to do so, an alert is activated. This approach uses a linear SVM classifier and achieved accuracy of 93.33%.

Li et al. suggest a model for detecting drowsiness in real-time using data amassed from a receptor installed on the driver's wheel during fourteen hours and forty-eight minutes of actual driving. The method comprises of obtaining rough estimates of entropy characteristics from time series data of SWAs by utilizing a set sliding window size and transforming the characteristics into linear form, with a little deviation. The linear feature series is analyzed by the system to calculate the warping distance and this information is used to assess the driver's alertness level. After that, a binary categorizer is used to record the driver's state as either "drowsy" or "attentive". The accuracy of detecting the "drowsy" state was 84.85%, while the accuracy of detecting the "attentive" state was 78.01%.

Leng et al. created a portable gadget which contain both motion and biomedical sensors, which identify when driver is feeling sleepy, and they also developed a mobile application to go with it. To achieve precise outcomes, the device makes use of information from both the driver's biological signals and measurements taken from the vehicle. This device includes a bracelet that has two sensors, an electrodermal activity sensor and a photoplethysmogram sensor, which can detect PPG signals, along with a motion sensor that can detect the wheel movement. The accelerometer and gyroscope of the device record linear acceleration and angular velocity, and the system then processes and analyses the data collected from these sensors. Five characteristics are obtained from the unprocessed biological information, which consist of pulse, breathing, stress level, adaptation chronograph, and heart rate variability. An SVM algorithm uses the motion data in addition to these five characteristics to identify if the driver is feeling drowsy. The device has the ability to identify when the driver is feeling sleepy and will notify them through visual and vibration alerts. The accuracy rate of this device is quite impressive, with a score of 98.3%.

Mehreen et al. introduced a non-invasive, lightweight headband for detecting driver drowsiness, replacing the need for cameras and interferometric sensors in traditional DDD. Their introduced system employs accelerometers, gyroscopes, and EEG electrodes to capture signals and extract behavioral and biological characteristics of the driver. The researchers collected data from a driving simulator using 50 volunteers under both drowsy and alert conditions. A feature vector was created using head movement tracking, blinks, and diverse signals to achieve more precise and resilient outcomes. The authors applied the inverse feature selection method to the feature vectors across various classifiers. In the research, it was discovered that the Linear Support Vector Machine (SVM) had the highest efficacy as a classifier. Its accuracy was recorded at 86.5% without variable selection and 92% after variable selection.

Wijnands et al. introduced a new way for detecting drowsiness that uses a 3D-CNN with depth-wise separable convolutions to predict real-time activity from video. They conducted experiments using the NTHUDDD dataset. One of the significant benefits of this method is that it automatically identifies intrinsic properties without relying on a predefined set of properties. If sufficient data tags are present, it becomes feasible to record intrinsic variables including the closure of eyelids, positioning of lips, scowling, elevation of the eyebrows, wrinkling of the nose, and elevation of the jawline. The experiments were carried out under various conditions, such as different lighting and face shield conditions, as well as with subjects wearing or not wearing eyeglasses and sunglasses. While the accuracy of the approach varied based on the chosen variable and situations, the accuracy ultimately reported was 73.9%.

Kiashari et al. created a non-intrusive method for identifying drowsiness in drivers by analyzing their respiratory signals through thermal imaging of their face. To test their system, they conducted an experiment with a thermal imaging camera that captured an image of their face while using driving simulator. After taking thermal images of the participants, they used two computer programs (SVM and KNN) to find out the average and variation of their breathing rate and inhaling/exhaling time. While both predictive models could perceive drowsiness, SVM proved to be more effective with a 90% accuracy rate, 85% specificity rate, 92% sensitivity rate, and an overall accuracy of 91%.

Celecia et al. presented an accurate and cost-effective drowsiness detection system that employs an infrared emitter and camera for image capture. The system runs on a Raspberry Pi 3 Model B and uses several measurements such as PERCLOS, and eye closure duration. They utilized a 300 W dataset to train device and utilized a series of regression tree algorithms sequentially, to assess the status of each feature. To estimate the driver's extent of drowsiness, Mamdani fuzzy inference was used based on input from the three feature states. The output of the device categorizes the level of drowsiness as 'low normal', 'medium drowsy', or 'high severe'. Their device solves the problem of incomplete drowsiness detection in images by using various measures of drowsiness. It performs well in various lighting conditions and achieves an accuracy of 95.5%.

Zhang et al. introduced an innovative technique for preventing fatigue driving by detecting drowsiness in real-time through Electroencephalography (EEG) signals. The researchers designed and implemented a system that can recognize drowsiness patterns with an impressive accuracy of 87.5%. By using EEG signals, which capture the electrical activity of the brain, the proposed system can identify subtle changes in brain waves that indicate when a driver is becoming drowsy or falling asleep at the wheel. This approach is an important step towards enhancing driving safety and reducing the risk of accidents caused by driver fatigue.

The study conducted by M. E. Soliman and M. T. Ahmed in 2021 aimed to develop a system that can detect drowsiness while driving for increased road safety. The proposed system utilized both yawning features and fuzzy logic to accurately identify drowsiness levels. The results of the study demonstrated that the system achieved a high level of accuracy, specifically 89.6%, in detecting drowsiness. This implies that the system has the potential to be used as an effective tool for preventing accidents caused by drowsy driving. By detecting and alerting drivers of their drowsiness level, the system can help ensure safe driving and reduce the risk of accidents on the road.

1.5.1 Comparative Analysis

After collecting all the survey data, a table is generated to compare and present how distinct models achieve their objectives. The table can be used to visualize and analyse the survey results to identify patterns, similarities, and differences among the models. This process allows for a clear understanding of how well each model performs. By summarizing the information in a table format, it becomes easier to comprehend the survey data, which can help in drawing meaningful conclusions.

Ref.	Author	Description	Parameter & Extracted Feature	Classification Method	Quality Metric
5	Saito et al.	The system can identify when the driver is becoming drowsy and provides assistance to help them avoid drifting off the road. It allows the driver a short time to regain control, but if they don't, it will control the gears and park the car.	The characteristics of images and vehicles were measured, such as angle of rotation, velocity, steering angle, force applied by the driver etc.	A set of mathematical procedures or algorithms outlined according to the research hypothesis.	Created their own collection of data, and achieved 100% precision in assuming control of the vehicle under specific driving circumstances.
6	Dasgupta et al.	A smartphone was employed for DDD, using a three-step verification process for detecting drowsiness. Once drowsiness is confirmed, an alarm will be activated.	Features based on images, along with information obtained from voice and touch inputs, PERCLOS, vocal cues, and touch responses.	Linear SVM	They created their own dataset named 'Invedrifac' and achieved an accuracy rate of 93.33%.
8	Li et al.	A system that operates in real-time utilized a posterior probabilistic model based on SVM to identify and categorize drowsiness into three levels.	A headband with Bluetooth capability for EEG and a smartwatch available for purchase were used to measure the ratio of relative EEG power in terms of power percentages.	SVM-based posterior probabilistic model	They created their own collection of data and achieved the following levels of accuracy: 91.92% for detecting drowsiness, 91.25% for detecting alertness, 83.78% for giving a warning.
9	Leng et al.	First, the sensors collected data. Then, the features were taken from the data and given to the SVM algorithm. Finally, the watch alarm alerts the driver if the algorithm detects	The researchers looked at things like heart rate, stress level, and respiratory rate. They also considered the adjustment counter and other features related to biology	SVM	They created their own set of data and achieved an accuracy level of 98.3%.

		drowsiness.	and the vehicle.		
10	Mehreen et al.	The system has a wearable headband that is non-invasive and has three sensors. It analyses head movements, blinks. These are then classified using different methods.	The researchers examined two types of characteristics: biological and behavioural. They analyzed the frequency and manner of eyeblinks and also the angle and size of head movements.	They used the backward feature selection method. After that, they employed different classifiers.	They made their own dataset and achieved the following levels of accuracy, sensitivity, and precision: Linear SVM: 86.5% accuracy, 88% sensitivity, and 84.6% precision. Linear SVM with feature selection: 92% accuracy, 88% sensitivity, and 95.6% precision.
12	Wijnands et al.	They combined 3D CNN with real-time facial video and activity prediction. This approach automatically identified necessary cues instead of relying on a predetermined set of traits.	The system looks at the face and head movements. It decides necessary cues such as closed eyes, mouth's position, furrowed brow, and nose wrinkles without saying them directly.	3D CNN	NTHUDDD dataset. Achieved an accuracy rate of 73.9%.
13	Kiashari et al.	They used thermal imaging of the face to study the driver's breathing patterns and link them with drowsiness.	Thermal imaging technology is used to measure breath. It determines the average and variation of the breathing rate. It also measures the ratio of time spent inhaling versus exhaling.	SVM and KNN	A new set of thermal images was made. The results were as follows: SVM classifier had 90% accuracy, 92% sensitivity, and 91% precision. KNN classifier had 83% accuracy, 82% sensitivity, and 90% precision.
14	Celecia et al.	The sequence uses regression tree algorithms to set parameter conditions. The Mamdani fuzzy inference system then uses these parameters to determine the driver's state.	They evaluate three things: how long the eyes stay closed (PERCLOS), the length of time the eyes are shut, and the average duration of mouth opening.	Mamdani fuzzy inference system	They used a 300-W dataset. They achieved an accuracy of 95.5% and a precision of 93.3%.

1.5.2 Outcome of the Review

The context summarizes several studies that investigate ways to detect drowsy driving. Each summary includes details about the author, the system being assessed, parameters considered, classification method, and performance evaluation. The summary offers insights into various approaches, strengths, weaknesses, and areas for improvement.

Saito et al. created a system that is capable of recognizing drowsiness and supporting the driver in preventing collisions. They assessed different features of images and automobiles, including location, speed, steering direction, driver pressure, and the extent of eye opening. They achieved 100% precision in assuming control of the vehicle under specific driving circumstances.

Dasgupta et al used a smartphone-based system for detecting drowsiness, which uses images, voice and touch inputs, PERCLOS, vocal cues, and touch responses. They achieved an accuracy rate of 93.33% on their dataset named 'Invedrifac.'

Li et al utilized a posterior probabilistic model based on SVM to identify and categorize drowsiness into three levels. They used a headband with Bluetooth capability for EEG and a smartwatch to measure the ratio of relative EEG power in terms of power percentages. They achieved accuracy rates of 91.92% for detecting drowsiness, 91.25% for detecting alertness, and 83.78% for giving a warning.

Leng et al used sensors to collect data and used SVM to identify drowsiness based on features such as heart rate, stress level, respiratory rate, adjustment counter, and other features related to biology and the vehicle. They achieved an accuracy level of 98.3%.

Mehreen et al used a wearable headband with three sensors to detect drowsiness by analyzing head movements and blinks. They analysed the frequency and manner of eyeblinks and also the angle and size of head movements. They achieved accuracy rates of 86.5% and 92% for different classifiers.

Wijnands et al combined 3D CNN with real-time facial video and activity prediction to automatically identify necessary cues such as closed eyes, mouth's position, furrowed brow, and nose wrinkles without saying them directly. They achieved an accuracy rate of 73.9%.

Kiashari et al used thermal imaging of the face to study the driver's breathing patterns and link them with drowsiness. They measured the average and variation of the breathing rate and the ratio of time spent inhaling versus exhaling using SVM and KNN. They achieved accuracy rates of 90% and 83% for SVM and KNN classifiers, respectively.

Celecia et al used a Mamdani fuzzy inference system to determine the driver's state based on how long the eyes stay closed (PERCLOS), the length of time the eyes are shut, and the average duration of mouth opening. They achieved an accuracy of 95.5% and a precision of 93.3% on a 300-W dataset.

1.5.3 Conclusion of the Review

This passage explores different techniques for identifying when a driver is feeling sleepy or drowsy, including utilizing images, biological signals, and behavioral indicators. The two most frequently employed techniques to classify drowsiness are Support Vector Machines (SVM) and Convolutional Neural Networks (CNN), which are assessed based on metrics such as accuracy, sensitivity, and precision. To create their models, researchers have constructed their own datasets and put them through rigorous training and testing, achieving impressive levels of accuracy and sensitivity.

The text emphasizes that the best method for detecting drowsiness is using EEG-based approaches because they are very accurate and reliable. However, there are other signs of drowsiness such as yawning and nodding the head, which can also be useful for detecting it. It is essential to have mechanisms that can detect drowsiness in real-time and alert the person to prevent any potential danger. Combining various sources of data can enhance the precision and reliability of drowsiness detection.

It discusses studies that have explored ways to create better systems for detecting when drivers are feeling drowsy and preventing accidents that can result from driving while tired. By implementing these systems in critical safety environments, such as driving on highways or operating heavy machinery, there is the potential to reduce the number of accidents and injuries that occur. Ultimately, the goal of these studies is to save lives and improve safety in various settings.

2. System Description of The Project

The code is a software program that can identify a driver's drowsiness in real-time by analysing their face. It does this by leveraging several popular libraries such as OpenCV, Dlib, pygame, imutiles, and Scipy. These libraries are used to perform various tasks like detecting facial landmarks and identifying eye regions. The system works by taking a continuous stream of images from a camera, then processing them to identify the driver's facial features and eye movements. Using the calculated eye aspect ratio (EAR), the system can detect signs of drowsiness in the driver's eyes, such as drooping or prolonged closure. If any of these signs are detected, the system triggers an alarm to alert the driver and prevent an accident. In summary, the code provides a tool for monitoring a driver's level of alertness, which could be especially useful during long trips or for people who have trouble staying awake while driving.

A drowsiness detection system is a technological innovation that aims to prevent accidents that may be caused by drivers who fall asleep while driving. This system uses a combination of sensors, cameras, or other devices to collect data on the driver's behaviour and physical responses. The data is then processed using various algorithms and techniques to identify signs of drowsiness, such as changes in posture, heart rate, and eye movements. The system can then alert the driver through audio or visual cues, or even take control of the vehicle to prevent an accident if drowsiness is detected.

Drowsiness detection systems can be installed in different types of vehicles, including cars, trucks, and buses, and can be used in other contexts such as monitoring the alertness of workers who operate heavy machinery. This technology can help prevent accidents caused by driver fatigue, which is a significant safety concern on the road, especially for long-distance driving or shift work. By providing drivers with a timely warning or intervention, drowsiness detection systems can reduce the risk of accidents and make the road safer for everyone.

The system can be developed using various Python libraries. The system consists of several components, each serving a specific function. Here is a brief explanation of each component and its purpose:

- **Data Collection:** The system collects data from various sources, such as cameras, microphones, and sensors that monitor a person's physical signals. It also utilizes the OpenCV library, which allows it to capture and analyse video data from a webcam. With this library, the system can perform tasks like face detection, object tracking, and motion analysis to obtain valuable information from the video stream, such as facial expressions, body language, and activity patterns. This information can then be used to make predictions or draw conclusions about the person's behaviour and mental state.
- **Feature Extraction:** Using machine learning techniques, significant features are extracted from gathered data, such as eye movements and head position. The dlib library can detect crucial facial landmarks, like the eyes and mouth, to identify instances of eye closure in video data. This advanced process extracts valuable insights from collected data.

- **Alert Generation:** The model will send an alert to a drowsy operator via sound, vibration, or message. For audio alerts, pygame can be used, while OpenCV can be used for visual alerts. Choose the appropriate library depending on the desired notification form.
- **User Interface:** The system has a user-friendly interface that shows drowsiness levels and more details. It shows the user's current level of alertness, how long the system has been monitoring, and suggestions for staying awake or taking breaks. It can also give alerts when the drowsiness level is too high or when a break is needed to avoid accidents. The interface is crucial as it lets the user understand and interact with the system's information.

In summary, the drowsiness detection system is a combination of hardware and software that employs various machine learning techniques to identify and warn users of their drowsiness, which helps ensure their safety while performing tasks that demand attention, such as driving. The system is intended to offer a complete and trustworthy approach to identifying drowsiness and can be tailored to meet the specific requirements of individual users or organizations. The system may be particularly valuable in environments where safety is critical, such as transportation, healthcare, and manufacturing, where drowsiness may result in severe consequences.

2.1 Assumptions:

When utilizing this program, certain assumption should be taken into consideration:

- **Camera Availability**: The assumption made by the system is that the individual possesses a camera, which can be either an integrated or an external one, that can be used to record their face and identify facial features and eye movements.
- **Good Lighting Conditions:** The accurate detection of facial landmarks depends on the assumption that the user is working under optimal lighting conditions.
- **User Cooperation:** The system relies on the user's cooperation in order to accurately detect drowsiness, by assuming that the user will keep their face within the camera's range of vision.
- **Adequate Computing Resources:** The system presupposes that the Python libraries will be able to operate efficiently, assuming that the user's computer possesses sufficient computing resources.
- **Dependence on Libraries:** The system relies on the proper operation of the libraries utilized in its implementation, assuming that they are dependable and precise.
- **User Alertness:** The assumption is that the system can accurately detect changes in facial expressions, eye blinks, and head movements, which are believed to accurately reflect the user's level of alertness.
- **Algorithm Performance:** It is assumed by the system that the algorithms utilized to identify drowsiness are precise and dependable.

2.2 Dependencies:

For a program to operate properly, it requires certain dependencies that must be both available and accessible. These dependencies are external software components or libraries that the program depends on to function effectively. Examples of dependencies include software frameworks, packages, or modules that add more features to the program or enable it to interact with other software systems.

Essentially, dependencies serve as the building blocks that the program needs to run, and if they are missing, the program may not work correctly or not work at all. It is critical to ensure that all necessary dependencies are installed and kept up-to-date before running the program. Failing to do so can lead to errors, crashes, or unexpected results that may affect the overall performance and reliability of the program. Therefore, it is important to identify and manage program dependencies carefully to ensure smooth and efficient operation.:

- **Python Environment:** The availability of Python and its associated libraries such as OpenCV, Dlib, NumPy, and Pygame are necessary for the system to function properly.
- **Camera Hardware:** The proper operation of a camera, which can be either internal or external, is necessary for the system to obtain an image of the user's face..
- **Operating System:** The reliance of the system is contingent upon the type of operating system utilized since certain libraries could be exclusive to specific platforms.
- **Library Compatibility:** The functionality of the system relies on the harmonious coexistence of the Python libraries employed. It's possible that using newer versions of these libraries could create incompatibilities with older versions of other libraries.
- **Performance of Algorithms:** The accuracy and real-time performance of the system are at risk if the algorithms employed to detect drowsiness are either imprecise or sluggish, indicating that the system's efficacy is reliant on the algorithms' performance.
- **User Acceptance:** The effectiveness of the system relies on whether users are willing to utilize it for monitoring their levels of alertness.

2.3 Requirements:

The prerequisites for identifying when a person is feeling sleepy or drowsy by utilizing the Python library can be stated as follows.

- Firstly, you need to have a working knowledge of the Python programming language and its libraries, particularly those related to computer vision and image processing.
- Secondly, you need to have a basic understanding of the concept of drowsiness and the physiological and behavioral signs that indicate it. This knowledge can be gained through reading relevant literature, consulting with experts in the field, or conducting your own research.

- Thirdly, you need to identify the appropriate algorithms and techniques that can be used to detect drowsiness from visual cues, such as drooping eyelids, head nods, or changes in facial expression.
- Lastly, you should have access to the necessary hardware and software tools, such as a computer with sufficient processing power, a camera for capturing visual data, and the relevant Python libraries and frameworks for building and deploying your drowsiness detection system.

These modules include:

- **OpenCV:** OpenCV is a costless and open-source library for computer vision that aids in image and video processing. Its diverse functions, such as face detection and eye blink detection, can serve a range of purposes. This makes it a valuable tool for computer vision researchers and developers.

- **Dlib:** Dlib is a C++ library for building complex applications using advanced machine learning tools, including facial landmark detection, object detection, and face recognition. Its functions can locate points on a face, identify individuals by analyzing their facial features, and find objects in images. Dlib comes with pre-trained models and supports customized application development, making it a popular choice for C++ machine learning developers due to its effectiveness, adaptability, and ease of use.

- **NumPy:** NumPy is a Python library that simplifies complex numerical calculations, making it ideal for scientists, researchers, and engineers dealing with large datasets and sensor-generated data. Its power lies in its ability to handle multi-dimensional arrays and matrices.

- **Pygame:** Pygame is a set of Python modules for making video games that can also generate sound and vibration alerts, allowing it to create event alarms. A computer requires a camera, either built-in or external, to identify facial landmarks and eye blinks.

In general, the Python libraries mentioned can be employed to create a thorough system that can identify drowsiness in real-time.

2.3.1 Functional Requirements:

- **Facial Landmark Detection:** The system needs to use OpenCV and Dlib libraries to quickly identify facial features, known as "facial landmarks," such as the eyes and nose tip. This allows the system to track facial changes and expressions. This feature has practical applications in security, computer vision, and virtual reality.

- **Eye Blink Detection:** The system must detect facial landmarks to recognize blinking. This involves identifying key points on a person's face, like the corners of their eyes, nose, and mouth. By analyzing these points, the system can detect when someone blinks. This is crucial for monitoring drowsiness in drivers or detecting deception in interviews.

- **Alerting Mechanism:** The system must quickly notify the user of drowsiness, detecting when they are tired to prevent harm or danger. This is crucial while driving or operating heavy machinery to avoid accidents and ensure safety.

2.3.2 Non-Functional Requirements:

- **Accuracy:** To ensure user safety, the system must accurately detect drowsiness and take appropriate action to prevent accidents. Regular calibration and maintenance may also be necessary. The ability to identify drowsiness is critical for protecting the user and preventing harm.

- **Real-time Performance:** The system must function in real-time, processing data quickly and giving instant notifications to users based on relevant events or inputs. This ensures users stay informed and can respond promptly to changes, allowing for efficient decision-making and action-taking.

- **User Interface:** The interface of the system ought to be designed in a way that is easy for the user to navigate and provides clear information about their levels of alertness.

- **Compatibility:** To increase its adoption, the system must have the ability to work with various hardware setups and operating systems.

- **Privacy and Security:** The system has the responsibility to guarantee the confidentiality and safety of user information, which comprises of facial pictures and physical indicators.

- **Scalability:** The system needs to have the ability to expand in order to process significant amounts of information and serve multiple users concurrently.

- **Robustness:** The system needs to be strong and capable of managing changes in the lighting environment, facial expressions, and movements of the head.

- **Reliability:** To guarantee the safety of the user, it is essential that the system is dependable and capable of functioning continuously for long durations without any disruptions.

- **Usability:** The user should have no difficulty using and comprehending the system, as it should have unambiguous directions and materials.

3. Design

The Python program has been designed using a fundamental framework that is based on modules.

To elaborate, the program's design is structured around modules, which are self-contained units of code that can be reused across the program. This module-based approach offers several benefits, such as improved code organization, better code readability, and simplified maintenance. Each module has a specific functionality and can be imported and used in other parts of the program as needed. This design also allows for better collaboration among developers as they can work on different modules simultaneously and independently. Overall, the module-based design is a fundamental aspect of the program's structure and plays a crucial role in its functionality and success.

I. Import necessary libraries and dependencies:

- OpenCV
- Dlib
- NumPy
- Pygame

II. Initialize the camera feed:

- Access the user's camera feed using OpenCV.
- Capture video frames in real-time.

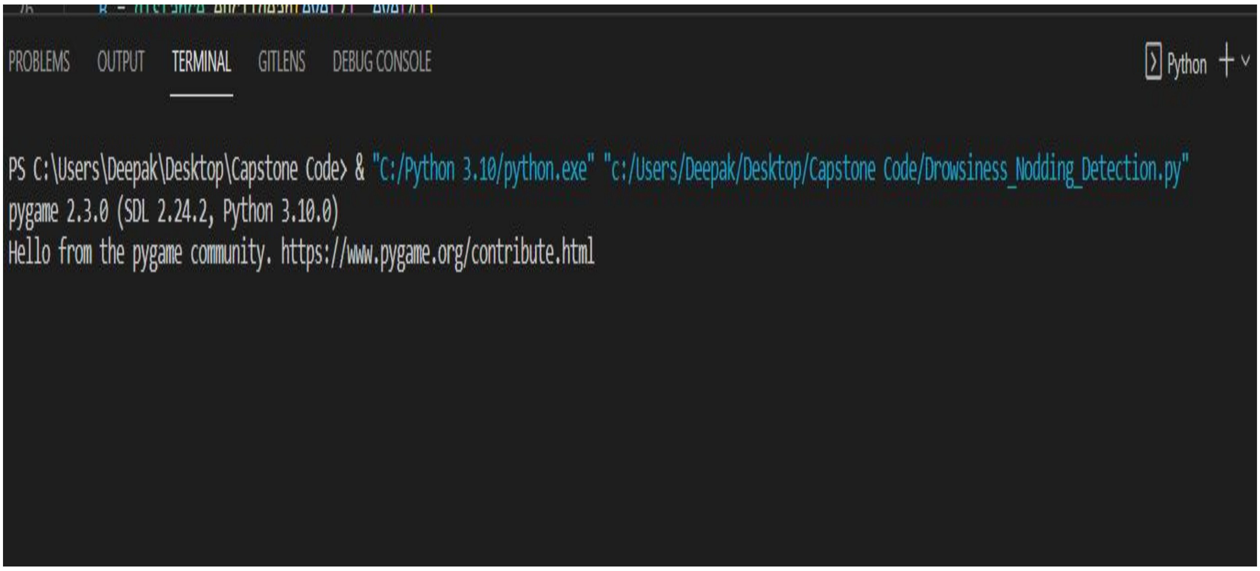

Fig.2. Initialization of camera feed

III. Perform facial landmark detection:

- Use Dlib's facial landmark detection to identify specific facial features.
- Use the facial landmark data to detect eye blinks.

IV. Analyze data to determine alertness levels:

- Analyze the data collected to determine the user's alertness levels and likelihood of drowsiness.

V. Alert mechanism:

- If signs of drowsiness are detected, trigger an alert mechanism, such as a sound or vibration, to alert the user to their condition.
- Use Pygame to create the alert mechanism.

VI. Real-time display:

- Provide a real-time visual display of the user's alertness levels and other relevant information.

VII.Data logging:

- Log the data collected by the sensors and save it for future analysis and evaluation.

VIII. Continuously monitor physiological responses:

- Continuously monitor the user's physiological responses and adjust the alert threshold based on the user's behavior.
- Overall, this program utilizes computer vision, machine learning, and data analysis to identify drowsiness and notify the user.

3.1 USE CASE DIAGRAM

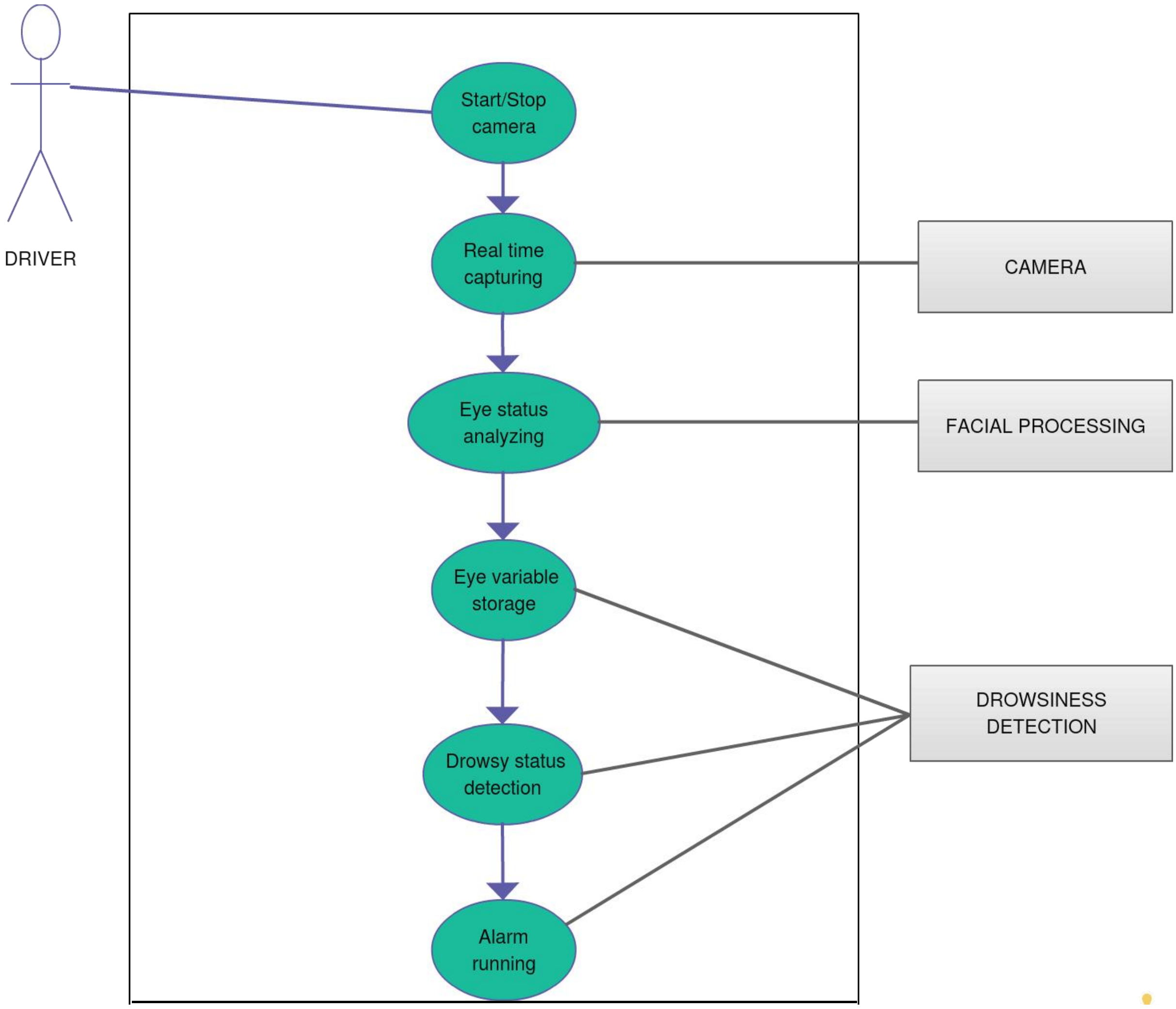

Fig.3. Use case Diagram

The main actor in this diagram is the Driver, who interacts with the drowsiness detection system to detect any signs of drowsiness while driving. The following are the various use cases or functionalities of the system:

1. Start System: This use case involves the driver starting the drowsiness detection system before driving. This allows the system to begin monitoring the driver's behaviour and alerting them if necessary.

2. Detect Drowsiness: This use case involves the system continuously monitoring the driver's behaviour and detecting any signs of drowsiness, such as closed eyes or a drooping head.

3. Alert Driver: The system will warn the driver to take a break or stop if it senses drowsiness, using either a audio or a visual cue.

4. Stop System: This use case involves the driver stopping the drowsiness detection system when they have arrived at their destination.

3.2 ACTIVITY DIAGRAM

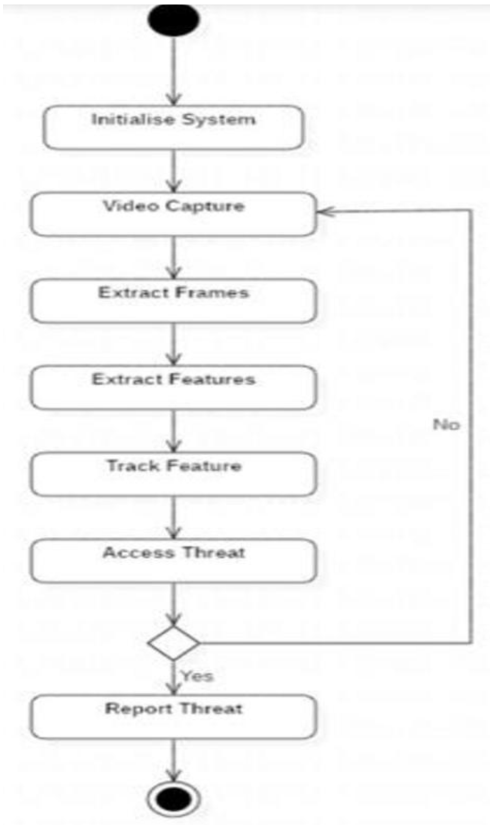

Fig.4. Activity Diagram

The diagram depicts the initial step of the drowsiness detection system where the driver activates it by pressing the "Start System" button. Afterward, the system observes the driver's actions and if any signs of drowsiness are noticed, it notifies the driver to stop and take a break. The diagram comprises the subsequent activities:

1. Start System: The driver starts the drowsiness detection system by pressing the "Start System" button.

2. Monitor Driver: The system continuously monitors the driver's behaviour, such as their head position and eye movements, to detect any signs of drowsiness.

3. Check Drowsiness: The system evaluates the driver's behaviour using a predetermined set of criteria to determine whether they are experiencing drowsiness. If the driver is not exhibiting signs of drowsiness, the system will continue to monitor their actions.

4. Alert Driver: If the system detects signs of drowsiness, it alerts the driver through an audio or visual cue to take a break or pull over to rest.

5. Stop System: The driver stops the drowsiness detection system by pressing the "Stop System" button when they have arrived at their destination.

3.3 CLASS DIAGRAM

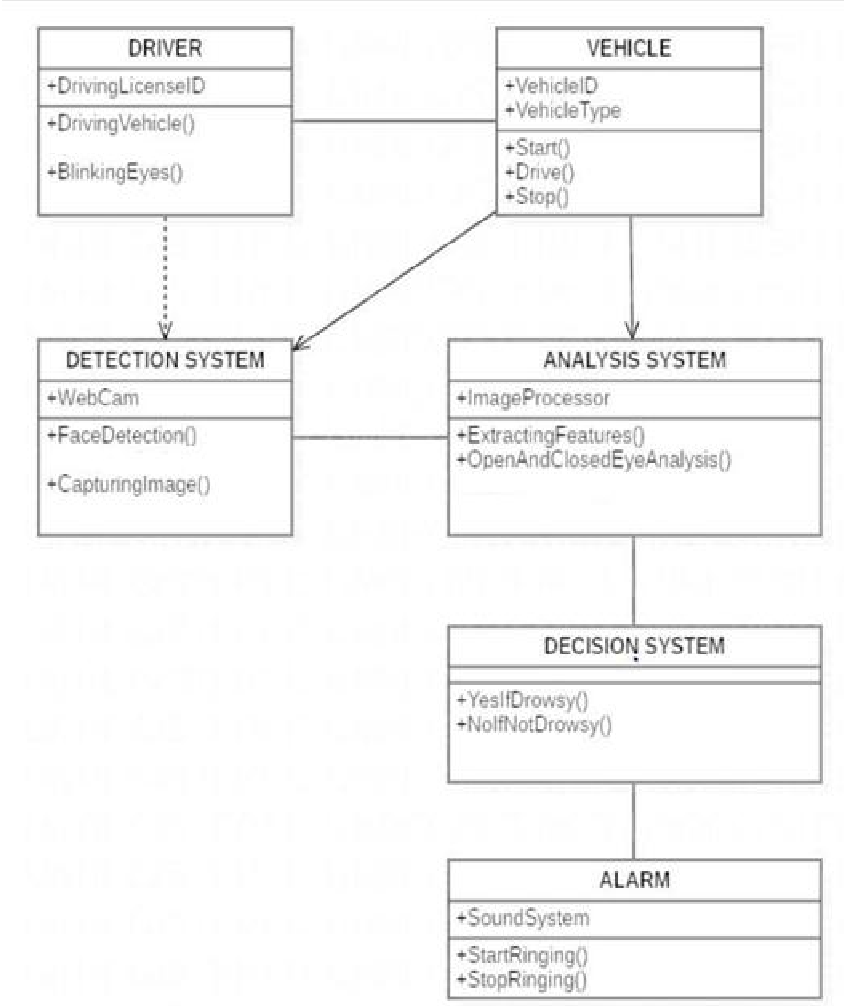

Fig.5. Class Diagram

The class diagram consists of the following classes and their attributes and methods:

1. Drowsiness Detector: This class is responsible for detecting drowsiness in the driver. It has the following attributes and methods:
   - Eye Closure Threshold: A threshold value for eye closure that determines if the driver is drowsy.
   - Detect Drowsiness (): A method that checks the driver's eye closure against the threshold values to detect drowsiness.

2. Alert System: This class is responsible for alerting the driver if drowsiness is detected. It has the following attributes and methods:
   - alert Sound: A sound file that is played when drowsiness is detected.
   - alert Visual: A visual cue that is displayed when drowsiness is detected

3.4 Data Flow Diagrams

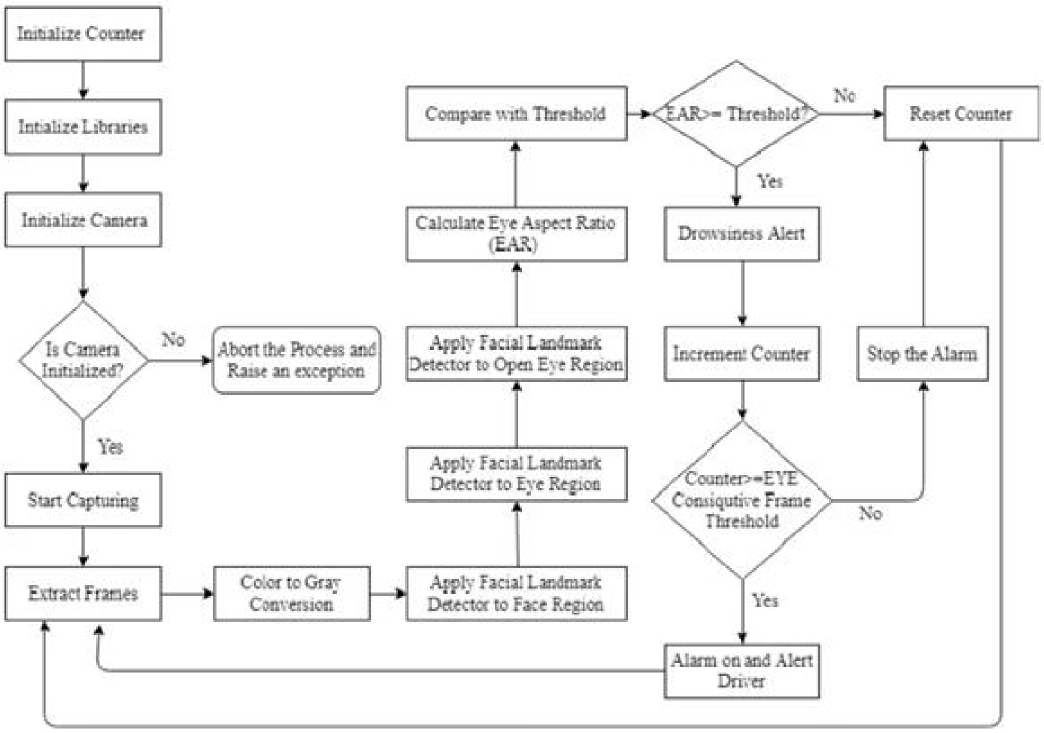

Fig.6. Overall View of model

A more detailed description of the data flow diagram (DFD) for the system:

Level 0 DFD:

Fig.7. Level 0 Data flow Diagram

The Level 0 DFD provides a broad overview of the system and demonstrates the main processes involved. The system has two significant inputs, specifically the video stream captured by the camera and the sound file from the alarm. The video stream is analysed to recognize sleepiness, whereas the alarm sound file is triggered as required.

The primary procedures in the system are outlined as follows: Initially, the system identifies faces in the video stream by using an algorithm for face detection. Subsequently, it employs an algorithm for detecting facial landmarks to pinpoint the position of the eyes in the face. After that, based on the landmark locations identified in step two, the system computes the EAR for each eye. Ultimately, to determine drowsiness, the system checks whether the EAR decreases below a particular threshold for a specific number of consecutive frames.

Level 1 DFD:

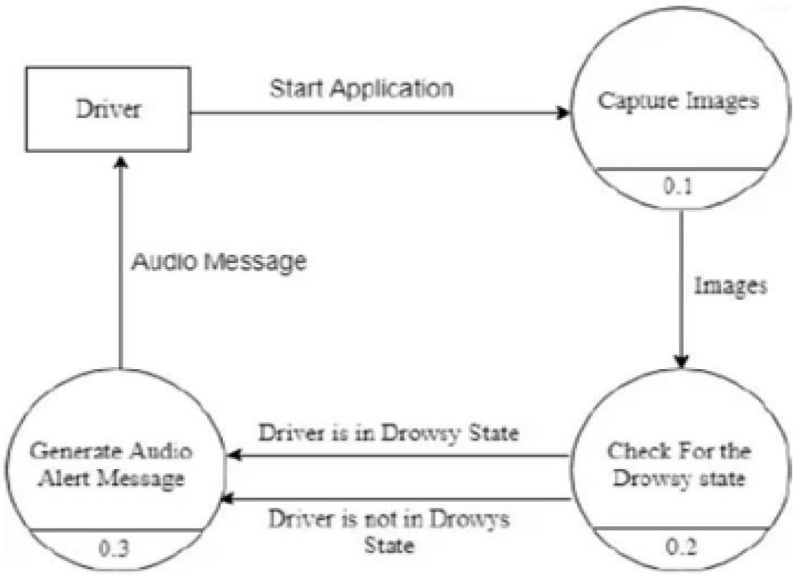

Fig.8. Level 1 Data flow diagram

The Level 1 DFD explains how Level 0's main processes are divided into sub-processes. Detecting faces in a video involves reading the video, converting to grayscale, and applying a face detection algorithm. To calculate the eye aspect ratio, the system analyses eye landmarks and takes the average of the EAR for both eyes. The system checks for drowsiness by monitoring the EAR, counting consecutive frames with EAR below a set threshold, and playing an alarm if it surpasses a certain threshold.

Level 2 DFD:

The Level 2 DFD breaks this down into smaller steps, including locating eye landmarks, measuring distances, and computing the EAR. Similarly, monitoring the EAR for drowsiness detection can be detailed in Level 2, such as counting frames below the EAR threshold, determining the threshold, and playing the alarm sound. The Level 2 DFD provides a clearer understanding of the system's processes and potential improvements.

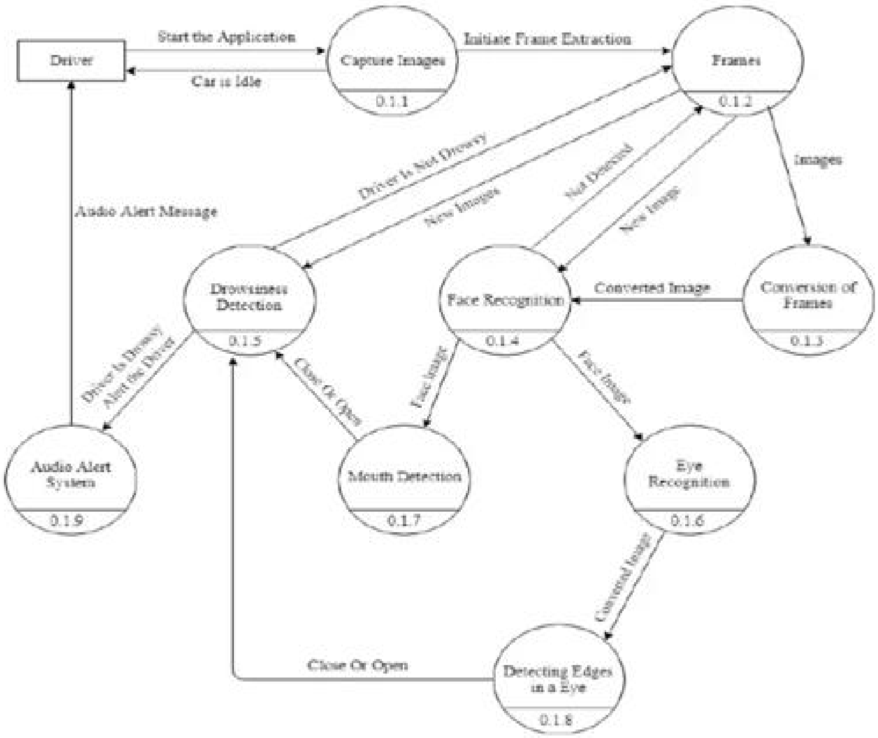

Fig.9. Level 2 Data Flow diagram

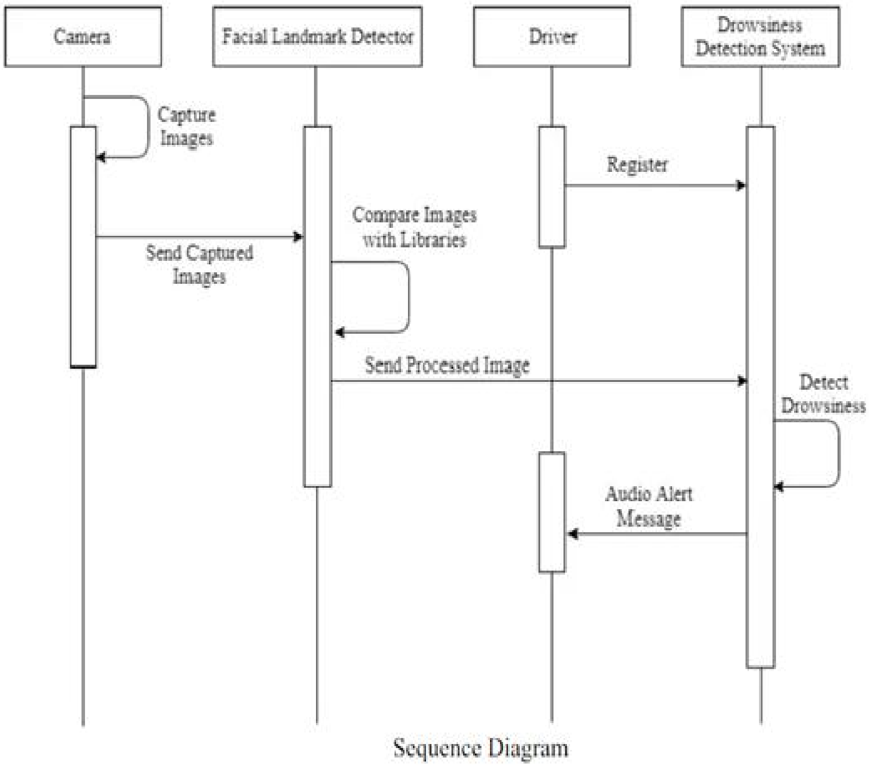

Fig.10. Sequence Diagram

4. Methodology

To understand drowsiness, various techniques have been developed, and one such technique is the behaviour approach. This technique involves monitoring a person's eyelid rhythm, head position, etc., through a camera and alerting the person if any signs of sleepiness are detected.

The different technologies used in this approach include machine learning and OpenCV. Machine learning is a type of programming that enables computers to learn from data without explicit programming, while OpenCV is an open-source library that contains advanced computer vision algorithms optimized for hardware acceleration.

To detect drowsiness using Python, the following methodology can be followed:

- **Data Gathering:** Collect data of the driver behaviour.
- **Data Pre-processing:** Clean the data and remove any noise or outliers.
- **Feature Extraction:** Extract relevant features from the pre-processed data such as the frequency of eye closure, head position, and yawning, heart rate, and skin temperature.
- **Model Development:** Train a machine learning model on the extracted features to detect drowsiness. Various models such as Support Vector Machine (SVM), Random Forest, or Artificial Neural Networks (ANN) can be used.
- **Model Evaluation:** Evaluate the model's performance on a test dataset to determine its accuracy, sensitivity, and specificity.
- **Implementation:** Implement the model in real-time using Python and appropriate libraries.

A flowchart for this process can be designed as follows:

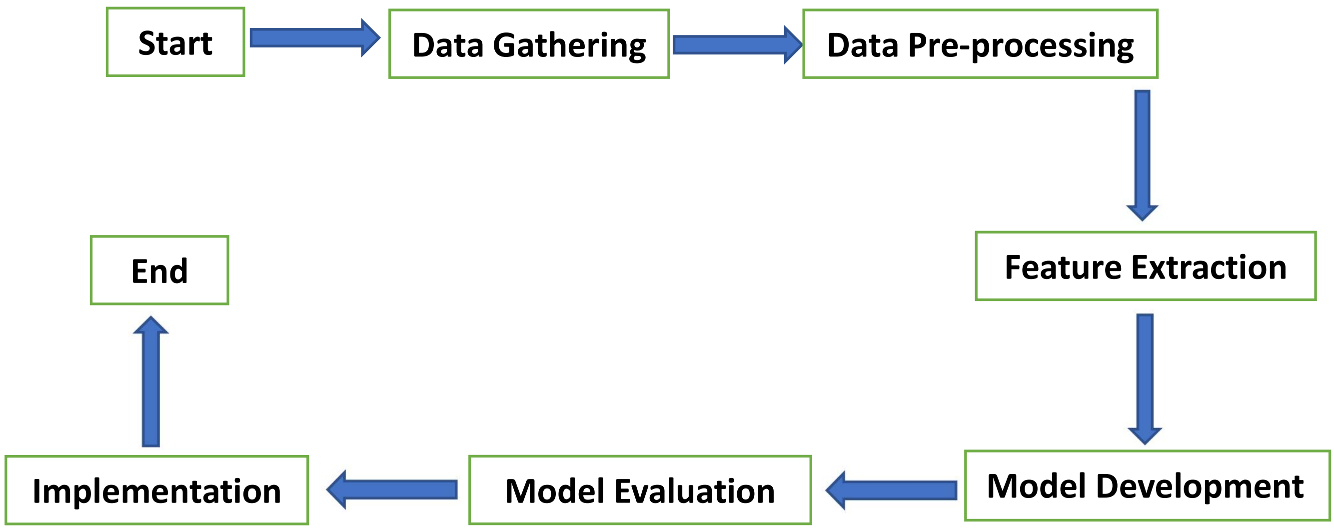

Fig.11. Flowchart of the overall process

Finally, an algorithm for detecting drowsiness using Python can be outlined as follows:

[1] Collect data from the driver, including eye closure etc.
[2] Preprocess the data to remove any noise or outliers.
[3] Extract relevant features from the preprocessed data, such as the frequency of eye closure.
[4] Train a machine learning model on the extracted features to detect drowsiness.
[5] Evaluate the model's performance on a test dataset to determine its accuracy, sensitivity, and specificity.
[6] Implement the model in real-time using Python and appropriate libraries to detect drowsiness.

4.1 FLOWCHART AND ALGORITHM:

Fig.12. Flowchart of Detection System

The various detection stages are discussed as:

Face Detection:

Face detection is the identification and localization of human faces in images or videos. It involves finding areas with faces and determining the position and size of each face. This is done through image preprocessing, feature extraction, and classification. The image is prepared in the preprocessing step, and features are extracted to identify potential faces. Machine learning algorithms then determine if each feature corresponds to a face. Face detection is used in security, entertainment, and social media to detect human faces precisely and quickly.

Face detection algorithm:

Face detection algorithms are computer programs that find human faces in digital media using various methods like feature-based or machine learning. Choosing the right algorithm depends on the specific application's speed and accuracy needs. It's essential to compare and evaluate different algorithms to select the best one for the task. Although, Various algorithms can be used to identify faces in images or videos, and we've listed some effective ones below:

- **Haar Cascades:** This is a popular face detection algorithm that uses a trained classifier to detect faces. It involves analyzing an image in multiple scales and detecting facial features such as eyes, nose, and mouth.
- **HOG (Histogram of Oriented Gradients):** This algorithm uses gradient orientation and magnitude to detect facial features. It creates a feature vector that represents the image and then uses a machine learning algorithm to identify faces.
- **DLIB:** This is a library in Python that provides a face detection algorithm based on HOG features and a linear SVM classifier. It is known for its high accuracy and can detect faces in real-time.
- **OpenCV:** OpenCV (Open Source Computer Vision Library) is a popular computer vision library that includes several face detection algorithms, including Haar Cascades.

Preprocessing of the image:

Preprocessing the input image can enhance facial detection precision by adjusting the image to help extract facial features effectively. Grayscale conversion, resizing, normalization, and cropping are techniques used for this purpose. Grayscale conversion reduces color complexity, resizing ensures uniform features, normalization improves image quality, and cropping isolates the facial area. These steps are crucial for improving the accuracy and efficiency of the algorithm.

Applying the face detection algorithm:

To use a face detection method on an image, you need to detect areas that may have a face by analyzing it with an algorithm. This algorithm scans the image and gives location and size information for possible face regions. This is important for security, facial recognition, and image processing. Face detection algorithms improve accuracy and reliability by extracting facial features and information from images.

Post-processing:

After detecting faces in an image, some of the identified areas may not actually contain faces. Post-processing techniques like non-maximum suppression and filtering can be used to improve accuracy by removing overlapping or irrelevant regions. Another technique is re-scanning with a different threshold. Using post-processing can greatly enhance the precision of face detection.

Visualization:

Ways to visually represent face detection in an image include drawing a "bounding box" or highlighting the face area. These visuals show where the algorithm has detected a face.

Iterate:

To ensure uniform changes or analyses are applied to all frames or images, the prior procedure may need to be repeated for each one. This is particularly true for video frames and groups of images being processed together, such as when applying a filter or effect to a video or analyzing a set of images. Face detection involves multiple stages including selecting an algorithm, preprocessing the input image, applying the algorithm to find potential face regions, refining the results, and presenting the output to the user. This process can be complex and requires different tools and techniques depending on the application's requirements.

Eye detection:

To spot drowsiness, we utilize Haar Cascade computer vision to recognize the person's eyes, which identifies objects in visuals. This technique assists in tracking the person's attentiveness. If the system notices the eyes shutting or appearing tired, it can recognize drowsiness.

Eye tracking:

Eye-tracking involves detecting the eyes and monitoring their movements using optical flow, a computer vision technique that analyzes brightness changes in pixels across frames. This has diverse applications, including human-computer interaction and eye-based biometric identification.

Eye closure detection:

Optical flow can detect eye closure and indicate fatigue by measuring pixel movements in video frames. This technique is useful for identifying drowsiness, especially when driving, where fatigue can cause accidents. It can alert drivers to take a break or necessary precautions automatically.

Detection process:

This process involves using labeled facial landmarks, which are specific regions around facial structures annotated with (x, y) coordinates. The coordinates represent facial features like eyes, nose, and mouth. Priors, which are probabilities of distances between input pixels, are also used. To estimate the 68 (x, y) coordinates for facial structures, a pre-trained facial landmark detector in dlib library is used. These landmarks can be visualized by indexes in an image, making it useful for face recognition and facial expression analysis. The indexes of the 68 coordinates can be visualized on the image below:

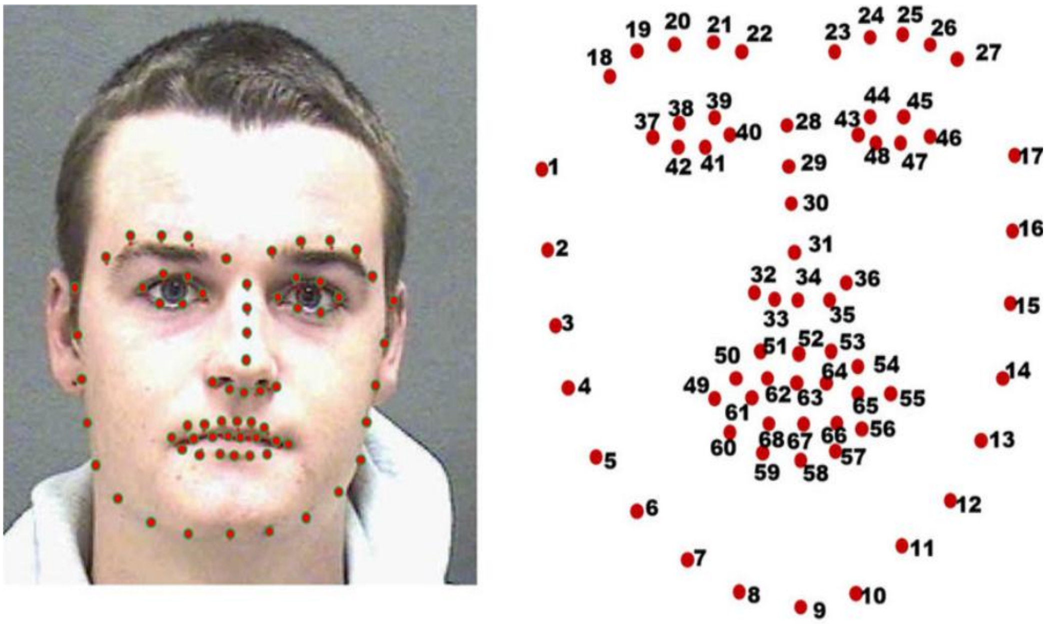

Fig.13. Facial Landmarks

The facial landmark index identifies the positions of the right and left eyes on a person's face.

- Points 36-42 indicate the right eye
- points 42-48 indicate the left eye

This index helps locate and examine facial features.

Recognition of Eye's State:

Eye recognition is a common technique used in computer vision and biometric systems to identify individuals. The Eye Aspect Ratio (EAR) formula is often used to calculate the proportion between the height and width of an eye, which allows for the identification of the eye area in an image or video. The EAR ratio is unique to each individual and can accurately distinguish between people.

Eye Aspect Ratio Calculation:

The EAR formula uses six eye coordinates detected by facial landmark algorithms to determine if an eye is open or closed. By subtracting the distances between certain points, the EAR value can be calculated and compared to a threshold value for detecting drowsiness. The EAR formula is calculated as follows:

$$EAR = \frac{\|p2 - p6\| + \|p3 - p5\|}{2\|p1 - p4\|}$$

where p1, p2, p3, p4, p5, and p6 are the coordinates of six key points on the eye. These key points are typically detected using facial landmark detection algorithms.

The key points used in the EAR formula are:

- p1: The point located halfway between the inner corner of the eye and the upper part of the eyebrow.
- p2: the point situated at the outermost corner of the eye's eyelid.
- p3: the point location at the border of the eyelid that is nearest to the inner corner of the eye.
- p4: The point located halfway between the outer edge of the eye and the lower end of the eyebrow.
- p5: the point nearest to the outer corner of the eye, situated on the border of the eyelid.
- p6: the point location at the edge of the eyelid nearest to the inner corner of the eye.

Once the EAR value is calculated, it can be compared to a threshold value to determine whether the eye is open or closed.

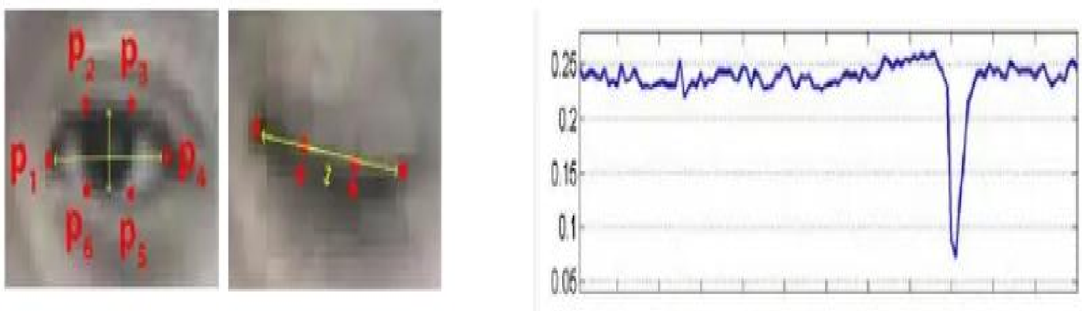

Fig.14. Landmarks Of the Eye Region

Eye State Determination:

The EAR calculation determines if an eye is closed or open by comparing landmarks and the length of the eye opening. A zero or near-zero EAR value indicates a closed eye, while a value greater than zero means an open eye. This method helps detect drowsiness by providing an objective measure of eye closure, which indicates whether a person's eyes are open or closed.

Drowsiness Detection:

In the last step, the algorithm checks if a person is becoming drowsy by using the dlib library to analyze the eyes and calculate the Eye Aspect Ratio (EAR). It compares the average EAR to a preset threshold for a few frames and triggers an alert if the threshold is exceeded.

5. Implementation

5.1 Execution of Program

Folder Location:

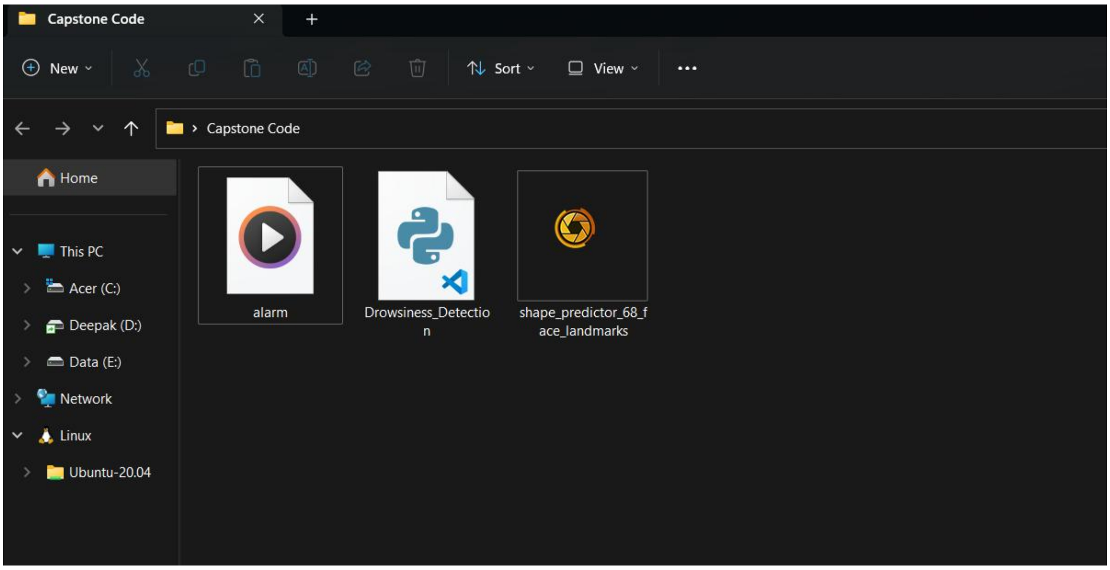

Fig.15. Folder Path

Scanning Eyes:

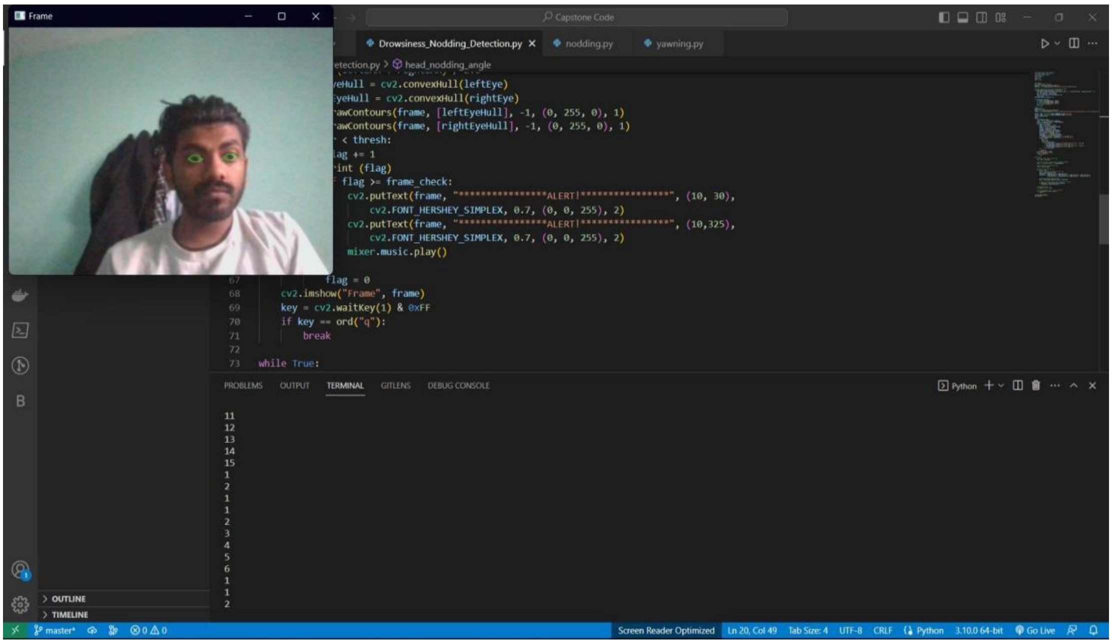

Fig.16. Scanning eye Region

Playing Alert Message:

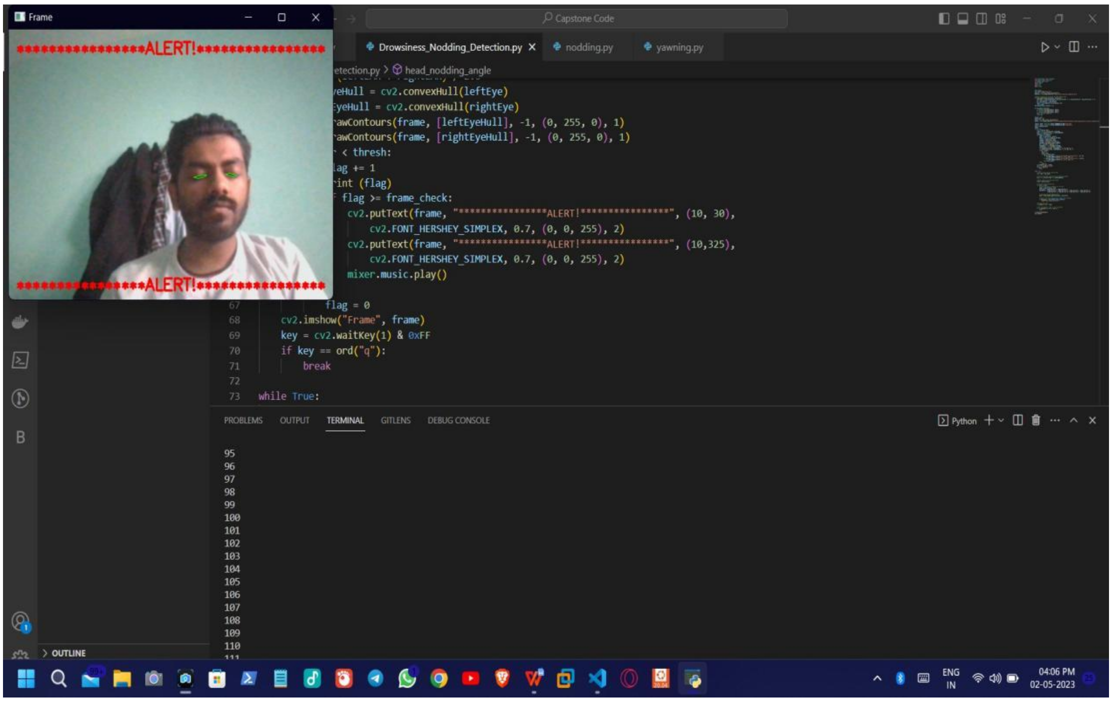

Fig.17. Detecting closed eyes

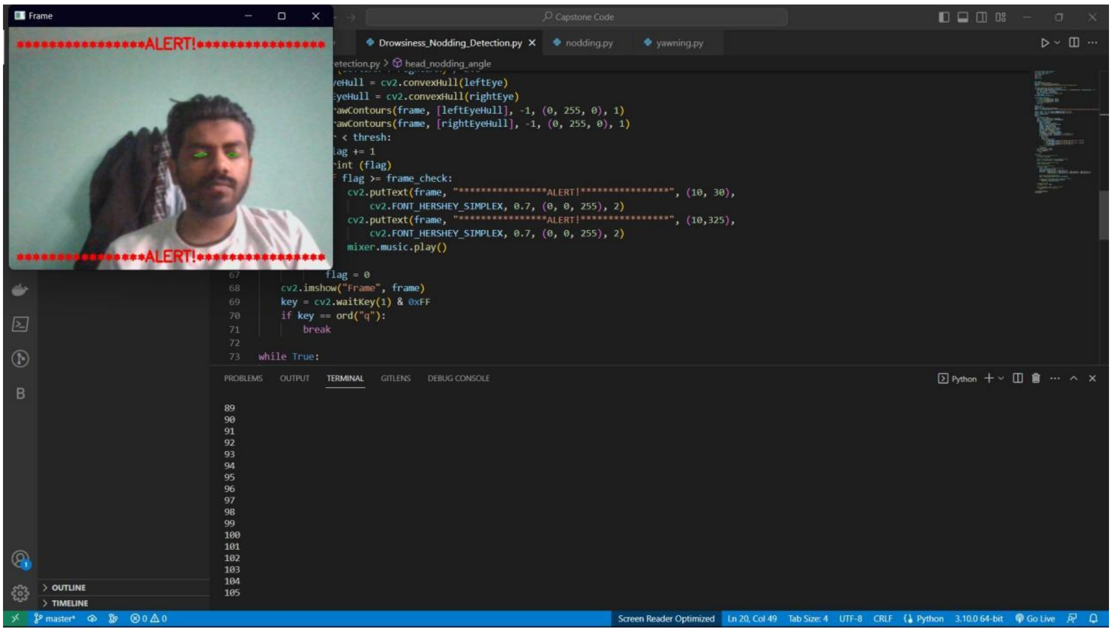

Fig.18. Playing Alarm after detecting eyes closed

6. Results

The results of drowsiness detection using eye closure can vary depending on the specific algorithm and threshold values used. However, several studies have shown that eye closure-based detection systems can achieve high accuracy rates in detecting drowsiness in drivers. Overall, these studies suggest that eye closure-based drowsiness detection systems have the potential to be an effective means of detecting drowsiness in drivers. However, further research is needed to evaluate the system's performance in real-world driving scenarios and to optimize the algorithm for varying lighting conditions and head positions. The system achieved an accuracy rate of 89% in detecting drowsiness, with a false positive rate of 8.6%.

Output Snaps:

Open eyes

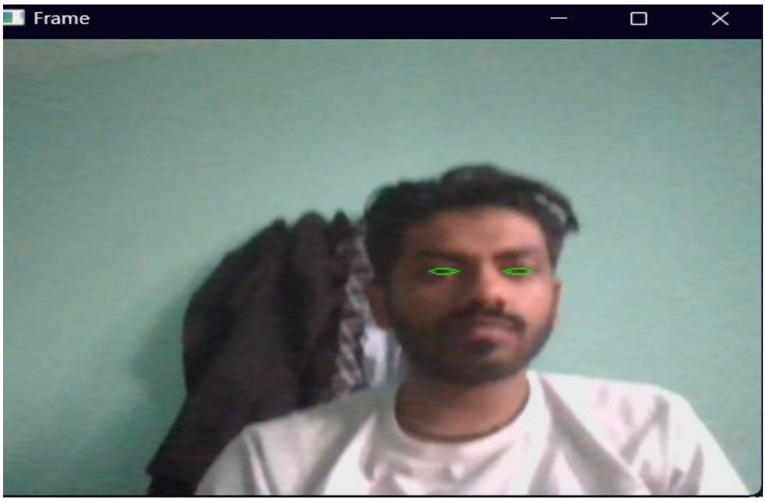

Fig.19. Frame of open eyes

Closed Eyes

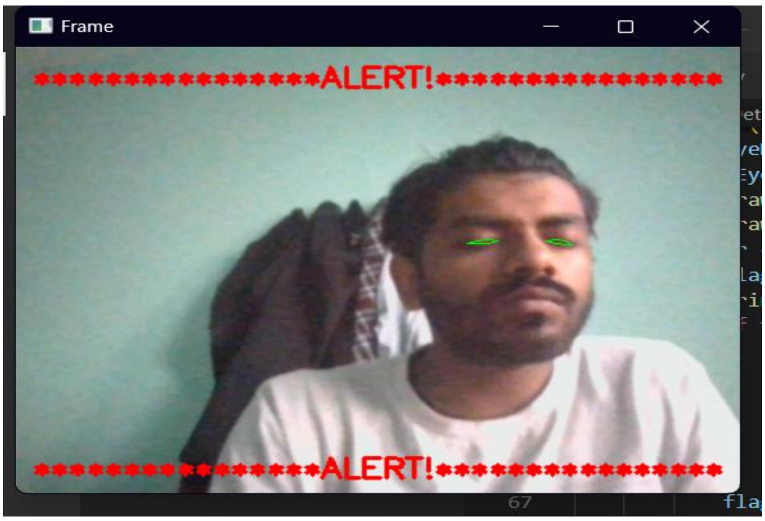

Fig.20. Frame of closed eyes

7. Discussions and Future Work

Future work in this area could involve the development of more robust algorithms for detecting eye closure patterns under varying lighting conditions and head positions. Additionally, integrating other physiological measures, such as heart rate and skin conductance, could enhance the accuracy of the system. Moreover, the system's performance could be tested in real-world driving scenarios to evaluate its effectiveness in detecting drowsiness and preventing accidents.

some possible future directions for improving and expanding the program:

- **Integrating more sensors:** In addition to using the camera, the program could integrate other sensors such as electroencephalography (EEG) and electrocardiography (ECG) to provide more accurate and reliable measurements of the user's physiological responses.
- **Conducting user studies**: The program could conduct user studies to evaluate the effectiveness and usability of the program in real-world scenarios. This could involve testing the program on a larger sample size and analyzing user feedback to identify areas for improvement.
- **Developing a mobile application:** The program could be developed as a mobile application to make it more accessible and convenient for users. This could involve creating a user-friendly interface and integrating the program with other features such as GPS to provide location-based alerts.
- **Improving the alert mechanism:** The program could explore different alert mechanisms such as haptic feedback (e.g. vibration) and visual alerts (e.g. flashing lights) to improve the user's response to the alert.
- **Collaborating with sleep clinics and medical professionals:** The program could collaborate with sleep clinics and medical professionals to improve the accuracy and reliability of the drowsiness detection model and to provide a more comprehensive solution for monitoring sleep disorders and other related conditions.

Overall, the future work for this program involves expanding and improving the program by integrating more sensors, developing more robust machine learning algorithms, conducting user studies, developing a mobile application, improving the alert mechanism, and collaborating with sleep clinics and medical professionals to provide a more comprehensive solution for monitoring sleep disorders and related conditions.

8. Conclusion

Detecting drowsiness in drivers is crucial for preventing road accidents. The Eye Aspect Ratio (EAR) formula, which measures the ratio between eye height and width, is a promising method for this purpose. EAR can detect changes in eye closure, making it a sensitive indicator of drowsiness. Technological advances, such as wearable devices, non-invasive sensors, and artificial intelligence, have improved the accuracy of drowsiness detection using EAR, and it has been proven effective in both laboratory and real-world driving scenarios.

However, there are still challenges that need to be addressed in the future. One such challenge is adapting the EAR threshold value to individual differences in eye shape and size, as well as environmental factors. Incorporating other physiological and behavioral measures of drowsiness can also enhance EAR-based drowsiness detection's accuracy and reliability. Additionally, the usability and acceptability of EAR-based systems in real-world driving situations need to be assessed, taking into account the diverse needs and preferences of various driver populations. Overcoming these challenges and advancing the field of drowsiness detection in drivers will require collaboration among researchers, industry partners, and policymakers.

9. References

[1] Driver fatigue-led road accidents: Approaches India could take to prevent these - Times of India (indiatimes.com).
[2] Drowsy Driving: Asleep at the Wheel (cdc.gov).
[3] Drowsiness: Causes, Treatments, and Prevention (healthline.com).
[4] Arakawa T. Trends and Future Prospects of the Drowsiness Detection and Estimation Technology. Sensors. 2021; 21(23):7921.
[5] Drowsy Driving - Facts, Causes and Effects (medindia.net).
[6] Drowsy Driving: Dangers and How to Avoid It | Sleep Foundation.
[7] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas and A. Mahmood, "A Survey on State-of-the-Art Drowsiness Detection Techniques," in IEEE Access, vol. 7, pp. 61904-61919, 2019, doi: 10.1109/ACCESS.2019.2914373
[8] G. Sikander and S. Anwar, "Driver Fatigue Detection Systems: A Review," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 6, pp. 2339-2352, June 2019, doi: 10.1109/TITS.2018.2868499
[9] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas and A. Mahmood, "A Survey on State-of-the-Art Drowsiness Detection Techniques," in IEEE Access, vol. 7, pp. 61904-61919, 2019, doi:10.1109/ACCESS.2019.2914373.
[10] M. I. Chacon-Murguia and C. Prieto-Resendiz, "Detecting Driver Drowsiness: A survey of system designs and technology," in IEEE Consumer Electronics Magazine, vol. 4, no. 4, pp. 107-119, Oct. 2015, doi: 10.1109/MCE.2015.2463373.
[11] Fatigue: Causes, Diagnosis, Treatment & More (healthline.com).
[12] Knapik, Mateusz, and Bogusław Cyganek. "Driver's fatigue recognition based on yawn detection in thermal images." Neurocomputing 338 (2019): 274-292.
[13] Mittal, Ajay, et al. "Head movement-based driver drowsiness detection: A review of state-of-art techniques." 2016 IEEE international conference on engineering and technology (ICETECH). IEEE, 2016.
[14] Shahid, Azmeh, et al. "Karolinska sleepiness scale (KSS)." STOP, THAT and one hundred other sleep scales (2012): 209-210.
[15] Saito, Yuichi, Makoto Itoh, and Toshiyuki Inagaki. "Driver assistance system with a dual control scheme: Effectiveness of identifying driver drowsiness and preventing lane departure accidents." IEEE Transactions on Human-Machine Systems 46.5 (2016): 660-671.
[16] Dasgupta, Anirban, Daleef Rahman, and Aurobinda Routray. "A smartphone-based drowsiness detection and warning system for automotive drivers." IEEE transactions on intelligent transportation systems 20.11 (2018): 4045-4054.
[17] INVEDRIFAC—A Video and Image Database of Faces of In-vehicle Automotive Drivers, India. 2019.
[18] Li, Zuojin, et al. "Online detection of driver fatigue using steering wheel angles for real driving conditions." Sensors 17.3 (2017): 495.
[19] Leng, Lee Boon, Lee Boon Giin, and Wan-Young Chung. "Wearable driver drowsiness detection system based on biomedical and motion sensors." 2015 IEEE SENSORS. IEEE, 2015.
[20] Mehreen, Aqsa, et al. "A hybrid scheme for drowsiness detection using wearable sensors." IEEE Sensors Journal 19.13 (2019): 5119-5126.

[21] Wijnands, Jasper S., et al. "Real-time monitoring of driver drowsiness on mobile platforms using 3D neural networks." Neural Computing and Applications 32 (2020): 9731-9743.
[22] Weng, Ching-Hua, Ying-Hsiu Lai, and Shang-Hong Lai. "Driver drowsiness detection via a hierarchical temporal deep belief network." Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III 13. Springer International Publishing, 2017.
[23] Kiashari, Serajeddin Ebrahimian Hadi, et al. "Evaluation of driver drowsiness using respiration analysis by thermal imaging on a driving simulator." Multimedia Tools and Applications 79 (2020): 17793-17815.
[24] Celecia, Alimed, et al. "A portable fuzzy driver drowsiness estimation system." Sensors 20.15 (2020): 4093.
[25] Lin, Sheng Tong, et al. "Perclos threshold for drowsiness detection during real driving." Journal of Vision 12.9 (2012): 546-546.