A PROJECT REPORT ON

“ DRIVER DROWSINESS MONITORING SYSTEM USING VISUAL BEHAVIOUR AND MACHINE LEARNING”

Submitted for partial fulfillment of the requirements For the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted By

VAIDUGULA HAASINI REDDY (121107012)
GAJAVELLI HARSHITHA (121107024)
AJMERA RASMITHA (121107036)
NAGAVELLI SWATHI(121107049)

Under the guidance of

DR. G. MADHAVI REDDY
(Asst.professor)

DEPARTMENT OF COMPUTER SCIENCE

CHAITANYA DEEMED TO BE UNIVERSITY
Kishanpura, Hanamkonda,
Telangana-506001   2023-2024

CHAITANYA DEEMED TO BE UNIVERSITY
Kishanpura,Hanamkonda,Telangana-506001

DEPARTMENT OF COMPUTER SCIENCE (2021-2024)

CERTIFICATE

This is to certify that **Ms. VAIDUGULA HAASINI REDDY** bearing **HTNO: 121107012**, **Ms. GAJAVELLI HARSHITHA** bearing **HTNO: 121107024**, **Ms. AJMERA RASHMITHA** bearing **HTNO**: **121107036**, **Ms. NAGAVELLI SWATHI** bearing **HTNO: 121107049** at **Chaitanya (Deemed to be University)** has satisfactory completed the project entitled **DRIVER DROWSINESS MONITORING SYSTEM USING VISUAL BEHAVIOUR AND MACHINE LEARNING** in the fulfillment of the requirements for the award of Degree **Bachelor of Computer Applications** during the academic Year **2023-2024.**

GUIDE	**Head of the Department**	**Dean Administration**
DR.G.MADHAVI REDDY	**DR.A.RAMESH BABU**	**DR.S.KAVITHA**
ASSISTANT PROFESSOR	**PROFESSOR**	**PROFESSOR**

Internal Examiner **External Examiner**

DECLARATION

We hereby declare that project report titled **"DRIVER DROWSINESS MONITORING SYSTEM USING VISUAL BEHAVIOUR AND MACHINE LEARNING "** is an original work done at **CHAITANYA(DEEMED TO BE UNIVERSITY)** Hanamkonda, submitted in fulfilment for the Bachelor of Computer Applications. We assured you that this project has not been submitted by any degree anywhere in this college or university.

VAIDUGULA HAASINI REDDY (121107012)
GAJAVELLI HARSHITHA (121107024)
AJMERA RASHMITHA (121107036)
NAGAVELLI SWATHI (121107049)

ACKNOWLEDGEMENT

We express our sincere gratitude to **DR.CH.V. Purushotham Reddy Chancellor, Dr. S. Kavitha** Professor Dean Administration of **CHAITANYA (Deemed to be University)** for his unstained interest in providing me facilities to complete the project successfully.

We experience from this kind of work is great and will be useful to me in future .We thank **Dr. A. Ramesh Babu Professor** & Head of the Department of Computer Science, for encouraging us to do such project work and for providing al the facilities to carry out this project.

We extend our special thanks to my Guide **Dr. G. MADHAVI REDDY (Asst.Professor),** who helped immensely with her valuable suggestions during the completion of the project work and who allowed us to do the project work and give us a lot of support and constant encouragement throughout this project.

We thank all the faulty members of Department of Computer Science for sharing their valuable knowledge with us. We extend my thanks to the **Technical staff** of the department for their valuable suggestions to technical problems. We also thank our family, friends for their hand in bringing this project to this present shape.

VAIDUGULA HAASINI REDDY (121107012)
GAJAVELLI HARSHITHA (121107024)
AJMERA RASHMITHA (121107036)
NAGAVELLI SWATHI (121107049)

INDEX

Contents **page**

ABSTRACT

Drowsy driving is one of the major causes of road accidents and death. Hence, detection of driver's fatigue and its indication is an active research area. Most of the conventional methods are either vehicle based, or behavioral-based, or physiological-based. Few methods are intrusive and distract the driver, some require expensive sensors and data handling. Therefore, in this study, a low-cost, real-time driver's drowsiness detection system is developed with acceptable accuracy.

In the developed system, a webcam records the video, and the driver's face is detected in each frame employing image processing techniques. Facial landmarks on the detected face are pointed and subsequently, the eye aspect ratio and mouth opening ratio are computed, and depending on their values, drowsiness is detected based on developed adaptive thresholding. Machine learning algorithms have been implemented as well in an offline manner.

INTRODUCTION

Drowsy driving is one of the major causes of deaths occurring in road accidents. The truck drivers who drive for continuous long hours (especially at night), bus drivers of long distance route or overnight buses are more susceptible to this problem. Driver drowsiness is an overcast nightmare to passengers in every country. Every year, a large number of injuries and deaths occur due to fatigue related road accidents.

Hence, detection of driver's fatigue and its indication is an active area of research due to its immense practical applicability. The basic drowsiness detection system has three blocks/modules; acquisition system, processing system and warning system. Here, the video of the driver's frontal face is captured in acquisition system and transferred to the processing block where it is processed online to detect drowsiness. If drowsiness is detected, a warning or alarm is send to the driver from the warning system. Generally, the methods to detect drowsy drivers are classified in three types; vehicle-based, behavioural based and physiological based. In vehicle based method, a number of metrics like steering wheel movement, accelerator or brake pattern, vehicle speed, lateral acceleration, deviations from lane position etc. are monitored continuously.

Detection of any abnormal change in these values is considered as driver drowsiness. This is a nonintrusive measurement as the sensors are not attached on the driver. In behavioural based method [1- 7], the visual behavior of the driver i.e., eye blinking, eye closing, yawning, etc. are analyzed to detect drowsiness. This is also nonintrusive measurement as simple camera is used to detect these features. In physiological based method [8,9], the physiological signals like Electrocardiogram (ECG), Electrooculogram (EOG), Electroencephalogram (EEG), heartbeat, pulse rate etc. are monitored and from these metrics, drowsiness or fatigue level is detected. This is intrusive measurement as the sensors are attached on the driver which will distract the driver. Depending on the sensors used in the system, system cost as well as size will increase. However, inclusion of more parameters/features will increase the accuracy of the system to a certain extent. These factors motivate us to develop a low-cost, real time driver's drowsiness detection system with acceptable accuracy. Hence, we have proposed a webcam based system to detect driver's fatigue from the face image only using image processing and machine learning techniques to make the system low-cost as well as portable.

The Rise of Driver Drowsiness Monitoring Systems.

Drowsy driving presents a significant threat on roadways worldwide. Similar to driving under the influence of alcohol, fatigue impairs judgment, coordination, and reaction times, leading to countless accidents, injuries, and fatalities each year. Studies have shown that the dangers of drowsy driving can be as severe as driving with a Blood Alcohol Content (BAC) exceeding the legal limit.

In response to this growing concern, Driver Drowsiness Monitoring Systems (DDMS) have emerged as a technological solution. These systems act as vigilant in-vehicle guardians, continuously monitoring the driver for signs of fatigue and issuing timely warnings to prevent accidents.

Understanding the Mechanisms of Drowsiness Detection.

DDMS employ a variety of techniques to detect driver fatigue. Some of the most common methods include:

Eye Movement Monitoring: Cameras track eye movements, focusing on blinking frequency, eyelid closure duration, and overall gaze patterns. Excessive blinking, prolonged eye closures, and frequent glances away from the road can all be indicative of drowsiness.

Facial Recognition: Advanced systems analyze facial features to detect signs of fatigue. These features can include drooping eyelids, yawning, and changes in facial expressions.

Steering Wheel Patterns: Some DDMS monitor steering wheel movements. Erratic movements, lane drifting, and a lack of steering corrections can suggest a decline in driver alertness.

Different Approaches to Driver Monitoring.

The specific methods employed by a DDMS depend on the technology used. Here's a breakdown of some common approaches:

Camera-Based Systems: These are the most widely used DDMS. A camera mounted within the vehicle captures video of the driver's face. Advanced image processing algorithms then analyze the video to detect signs of drowsiness.

Wearable Device Systems: These systems utilize sensors in wristbands, headbands, or earpieces to monitor physiological data like heart rate, brainwaves, and blood oxygen levels. The collected data is then analyzed by algorithms to assess the driver's alertness.

Vehicle-Based Systems: Some vehicles come equipped with built-in features that can indirectly detect drowsiness. These might include steering wheel movement monitoring or lane departure warning systems.

The Advantages of Driver Drowsiness Monitoring

DDMS offer a range of potential benefits for both drivers and public safety in general. Here are some key advantages:

Reduced Accident Risk: By detecting drowsiness early and issuing warnings, DDMS can help prevent accidents caused by fatigued drivers.

Enhanced Driver Alertness: DDMS can act as a wake-up call for drivers, prompting them to take a break when needed, ensuring they remain alert and focused on the road.

Improved Safety Measures: DDMS can integrate with other driver-assistance features like lane departure warnings or adaptive cruise control, forming a comprehensive safety net on the road.

Data-Driven Insights: Data collected by DDMS can be used to identify patterns of fatigue and develop strategies to mitigate drowsy driving, such as improved rest area placement on highways.

Who Can Benefit Most From DDMS?

While DDMS are valuable for all drivers, some groups stand to benefit particularly:

Long-Haul Truck Drivers: These professionals often spend long hours on the road, making them highly susceptible to fatigue. DDMS can provide a crucial safety net for them.

Shift Workers: People who work irregular hours, including night shifts, are more prone to drowsiness during daytime driving. DDMS can offer an additional layer of protection.

Individuals with Sleep Disorders: Those with conditions like sleep apnea are at an increased risk of fatigue behind the wheel. DDMS can serve as an additional safeguard.

The Road Ahead: The Future of Driver Drowsiness Monitoring

As technology continues to evolve, DDMS are expected to become even more sophisticated and widely used. Future advancements might include:

Integration with Other Safety Features: DDMS could become more seamlessly integrated with other features like adaptive cruise control and lane departure warning systems, creating a more comprehensive safety suite.

Personalized Alerts: Systems might adjust alerts based on individual drivers' habits and preferences, tailoring warnings to optimize effectiveness.

Connection to Traffic Networks: DDMS could potentially communicate with traffic networks to warn other drivers or infrastructure of potential hazards caused by drowsy drivers.

By staying alert and vigilant, Driver Drowsiness Monitoring Systems are poised to play a vital role in reducing drowsy driving accidents and creating safer roads for everyone.

Machine Learning:

The effectiveness of Driver Drowsiness Monitoring Systems (DDMS) hinges on a powerful technology called Machine Learning (ML). In essence, machine learning equips computers with the ability to learn and improve without explicit programming.

Imagine a child learning to identify different animals. Initially, the child might need someone to point out a dog and say "that's a dog." Over time, with repeated exposure and observation, the child can recognize a dog on their own, even encountering different breeds or sizes. Machine learning works in a similar way.

The Core of Machine Learning: Data and Algorithms

Machine learning algorithms are the tools that enable computers to learn. These algorithms are fed vast amounts of data, allowing them to identify patterns and relationships within that data. The data used in DDMS can include:

Facial landmarks: The position of eyes, mouth, and other facial features captured by the camera.

Eye movement data: Blinking frequency, eyelid closure duration, and gaze patterns.

Steering wheel patterns: Movements, corrections, and lane deviations.

Physiological data (wearable devices): Heart rate, brainwaves, and blood oxygen levels (if applicable).

The Learning Process: Training the Machine

The data is then used to train the machine learning algorithms. During training, the algorithms analyze the data, identifying patterns that differentiate between a driver who is alert and one who is drowsy. These patterns might include:

- The average blink rate of an alert driver compared to a drowsy one.
- The typical duration of eye closures during normal driving versus fatigue.
- The frequency of steering corrections when alert versus drowsy.
- Physiological changes indicative of fatigue (wearable devices only).

Putting Knowledge to Use: Real-Time Detection

Once trained, the machine learning algorithms can analyze data from the driver in real-time. By comparing the incoming data to the established patterns, the system can assess the driver's alertness level. When signs of drowsiness are detected, the system triggers an alert, which could be:

- An audible warning sound.
- A visual notification on the dashboard.
- A vibration in the driver's seat.

The Ongoing Evolution: Continuous Learning

Machine learning algorithms don't stop learning after the initial training phase. As DDMS collect more data from real-world usage, the algorithms can be continuously refined. This ongoing process helps to improve the accuracy and effectiveness of drowsiness detection over time.

The Benefits of Machine Learning in DDMS

Machine learning plays a critical role in the success of DDMS by:

Automating Drowsiness Detection: ML algorithms remove the need for manual analysis, ensuring consistent and objective detection of fatigue signs.

Adapting to Individual Drivers: Over time, the system can learn individual driving patterns and adjust alerts accordingly, leading to more personalized detection.

Continuous Improvement: Machine learning allows for continuous refinement as more data is collected, enhancing the overall accuracy of the system.

In conclusion, machine learning is the backbone of Driver Drowsiness Monitoring Systems. By providing the ability to learn and adapt, machine learning empowers DDMS to effectively detect drowsiness and keep drivers safe on the road.

Background of the Driver Drowsiness Monitoring System Project

The growing concern over drowsy driving has fueled the development of innovative solutions like Driver Drowsiness Monitoring Systems (DDMS). Here's a breakdown of the key factors contributing to the emergence of this project:

Prevalence of Drowsy Driving: Studies reveal a significant threat posed by drowsy driving. It's estimated to be a contributing factor in a substantial number of road accidents, injuries, and fatalities each year. The dangers of drowsy driving can be as severe as driving under the influence of alcohol.

Impact of Fatigue on Drivers: Fatigue impairs various cognitive and physical abilities crucial for safe driving. These include:

Decreased reaction time: Drowsy drivers take longer to react to hazards on the road, increasing the risk of collisions.

Poor judgment: Fatigue can lead to risky driving decisions like speeding or improper lane changes.

Microse (shuimian - sleep) Episodes: Drowsy drivers might experience brief moments of unintentional sleep (microsleeps) lasting a few seconds, significantly compromising control of the vehicle.

Limitations of Traditional Methods: Traditional methods for detecting drowsy driving, such as observing driving patterns or relying on self-awareness, have proven to be insufficient. Drivers might not recognize their own fatigue until it's too late, and erratic driving patterns can have other causes.

Advancements in Technology: The project leverages advancements in computer vision and machine learning, making it possible to develop a more objective and reliable system for drowsiness detection.

Potential Benefits of DDMS:

The Driver Drowsiness Monitoring System project aims to address the critical issue of drowsy driving by offering several potential benefits:

Reduced Accident Risk: By detecting drowsiness early and issuing warnings, DDMS can help prevent accidents caused by fatigued drivers.

Enhanced Driver Alertness: Timely warnings can act as a wake-up call, prompting drivers to take breaks and maintain focus on the road.

Improved Safety Measures: DDMS can integrate with other driver-assistance features like lane departure warnings or adaptive cruise control, creating a comprehensive safety net for drivers.

Data-Driven Insights: Data collected by DDMS can be used to identify patterns of fatigue and develop strategies to mitigate drowsy driving, such as improved rest area placement on highways.

Target Users:

While valuable for all drivers, DDMS can be particularly beneficial for groups like:

Shift Workers: Individuals working irregular hours, including night shifts, are more prone to drowsiness during daytime driving. DDMS can offer an additional layer of protection.

People with Sleep Disorders: Those with conditions like sleep apnea experience increased risk of fatigue behind the wheel. DDMS can serve as an additional safeguard.

Overall, the Driver Drowsiness Monitoring System project addresses a critical road safety concern by leveraging technology to promote alertness and prevent accidents caused by drowsy driving.

1.1 Existing system:

- In existing system the driver drowsiness detection system involves controlling accident due to unconsciousness through Eye blink.
- Here one eye blink sensor is fixed in a vehicle where if the driver loses consciousness,
- Then it alerts the driver through a buzzer to prevent the vehicle from an accident.

Disadvantages of existing System:

- Not Reliable
- May damage retina
- Highly expensive
- Intrusive
- Not portable

1.2 Proposed System:

In Proposed System, a low-cost, real time driver's drowsiness detection system developed with acceptable accuracy.

A webcam based system is used to detect driver's fatigue from the face image using image processing and machine learning techniques.

- Access
- Processing
- Warning

2. LITERATURE SURVEY

Intelligent Video-Based Drowsy Driver Detection System under Various Illuminations and Embedded Software Implementation

Authors: Wei-Liang Ou,Ming-Ho Shih,Chien-Wei Chang,Xue-Han Yu,Chih-Peng Fan

Abstract: An intelligent video-based drowsy driver detection system, which is unaffected by various illuminations, is developed in this study. Even if a driver wears glasses, the proposed system detects the drowsy conditions effectively. By a near-infrared-ray (NIR) camera, the proposed system is divided into two cascaded computational procedures: the driver eyes detection and the drowsy driver detection. The average open/closed eyes detection rates without/with glasses are 94% and 78%, respectively, and the accuracy of the drowsy status detection is up to 91%. By implementing on the FPGA-based embedded platform, the processing speed with the 640×480 format video is up to 16 frames per second (fps) after software optimizations.

Driver Fatigue Detection based on Eye Tracking and Dynamic Template Matching

Authors: W. B. Horng, C. Y. Chen, Y. Chang, C. H. Fan

Abstract: A vision-based real-time driver fatigue detection system is proposed for driving safely. The driver's face is located, from color images captured in a car, by using the characteristic of skin colors. Then, edge detection is used to locate the regions of eyes. In addition to being used as the dynamic templates for eye tracking in the next frame, the obtained eyes' images are also used for fatigue detection in order to generate some warning alarms for driving safety. The system is tested on a Pentium III 550 CPU with 128 MB RAM. The experiment results seem quite encouraging and promising. The system can reach 20 frames per second for eye tracking, and the average correct rate for eye location and tracking can achieve 99.1%

on four test videos. The correct rate for fatigue detection is l00%, but the average precision rate is 88.9% on the test videos.

Monitoring Driver Fatigue using Facial Analysis Techniques Authors:

S. Singh, N. P. Papanikolopoulos

Abstract: In this paper, we describe a non-intrusive vision-based system for the detection of driver fatigue. The system uses a color video camera that points directly rewards the driver's face and monitors the driver's eyes in order to detect micro-sleeps (short periods of sleep). The system deals with skin-color information in order to search for the face in the input space. After segmenting the pixels with skin like color, we perform blob processing in order to determine the exact position of the face. We reduce the search space by analyzing the horizontal gradient map of the face, taking into account the knowledge that eye regions in the face present a great change in the horizontal intensity gradient. In order to find and track the location of the pupil, we use gray scale model matching. We also use the same pattern recognition technique to determine whether the eye is open or closed. If the eyes remain closed for an abnormal period of time (5-6 sec), the system draws the conclusion that the person is falling asleep and issues a warning signal..

3. SYSTEM REQUIREMENTS SPECIFICATION

Hardware Requirements:

System	: Pentium Dual core.
Processor	: Intel-Core 15
Hard Disk	: 120 GB.
RAM	: 8 GB.

Software Requirements:

Operating system	: Windows 10.
Frontend	: Tkinter
Backend	: Python
Coding Language	: Python Version 3.8

4. DESIGN

4.1 UML Diagrams Introduction

The unified modeling language (UML) is general purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. The unified modeling language (UML) offers a way to visualize a system's architecture blueprints in a diagram (see image), including elements such as any activities individual components of the system and how they can interact with other software components. How the system will run, how entities interact with others components and interfaces) external user interface.

Although originally intended solely for object-oriented design documentation, the unified modeling language (UML) has been extended to cover a larger set of desi documents and been found useful in many contexts.

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmaticrules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by asset of diagram, which is as follows • User Model View

i. This view represents the system from the user's perspective

ii. The analysis representation describes a usage scenario from the end-user's perspective

• Structural Model View

i. In this model the data and functionality are arrived from inside the system
ii. This model view models the static structures

- Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view

- Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are • Environmental Model

In these the structural and behavioral aspects of the environment in which UML is specifically constructed through two different domains they are:

The system is to be implemented are represented UML analyses modeling this focuses on the user model and structural model views of the system.

UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views. Use case Diagram represent the functionality of the system from a user's point of view Use case are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view. There are four types of UML diagrams for this project namely

- Class diagram
- Use case diagram
- Sequence diagram
- Activity diagram
- Data Flow diagram

UML Diagrams

4.2.1 Class Diagram :

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes interfaces associations and collaborations. Class diagram basically represent the object reigned view of a system which is static in nature. of a system which is static in nature.

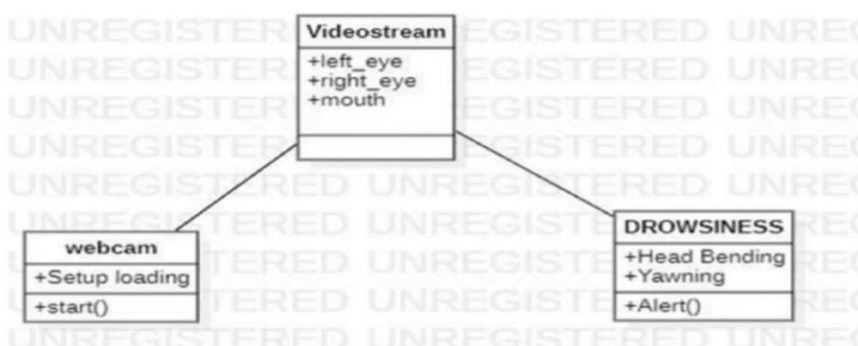


Fig 4.1: Class diagram

4.2.2. Use case diagram :

Use case diagrams are a set of use cases, actors and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. So, use case diagram is used to describe the relationship among the functionalities and their internal external controllers. These controllers are known as actors.

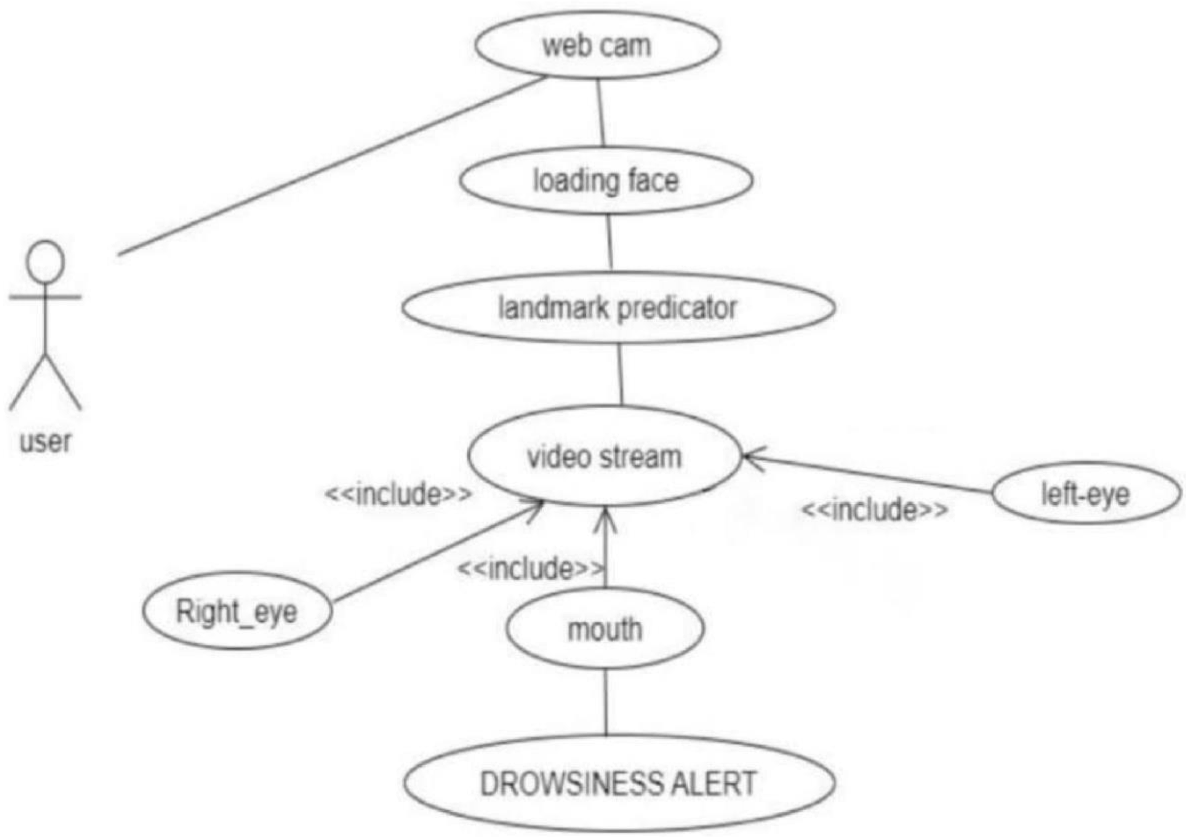


Fig 4.2: Use Case diagram

4.2.3. Sequence diagram :

A sequence diagram is an interaction diagram. From the name it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the components of a system is very important from implementation and execution perspective. So sequence diagram is used to visualize the sequence of calls in a system to perform aspecific functionality.

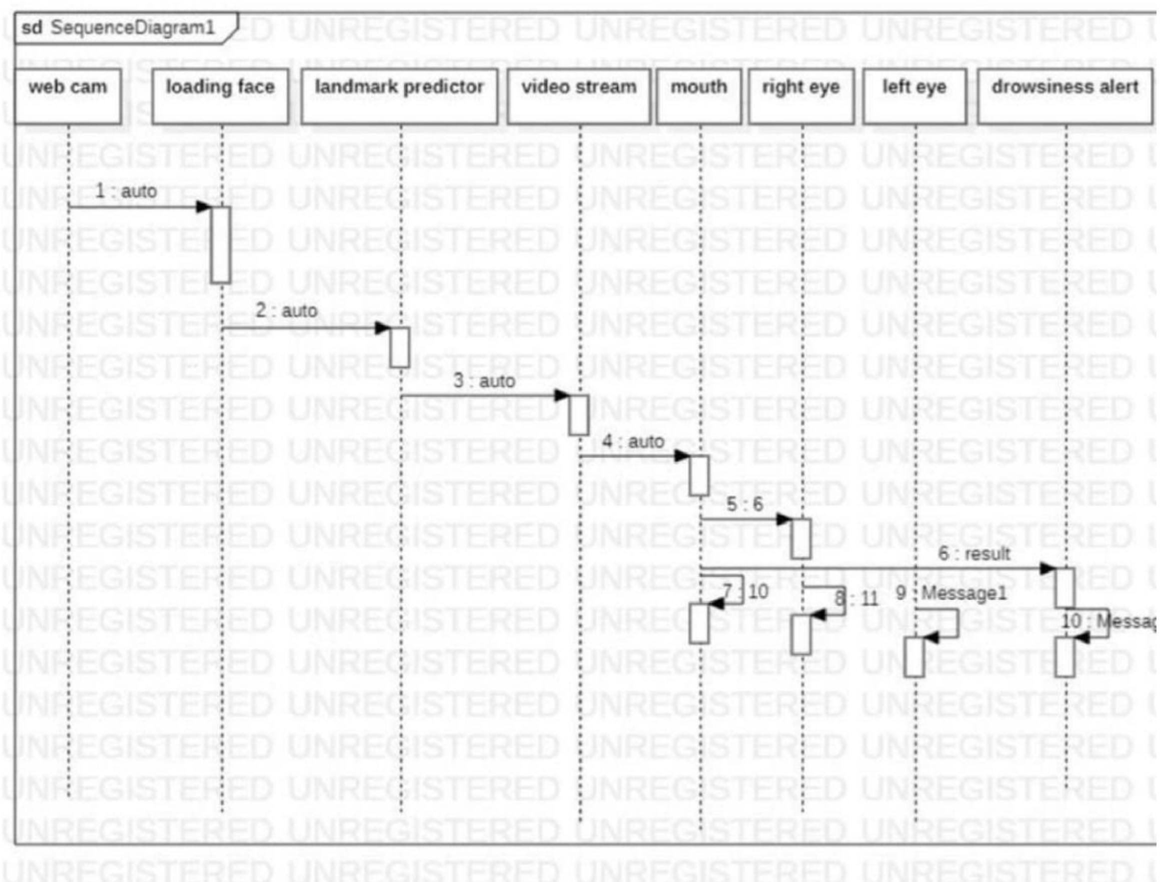


Fig 4.3: Sequence diagram

4.2.4. Activity diagram of System:

Activity diagram describes the flow of control in a system. So it consists of activities and links. The flow can be sequential, concurrent or branched activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system. Activity diagrams are used to visualize the flow of control in a system. This is prepared to have an idea of how the system will work when executed in this project the activity diagram shows the flow functions/activities.

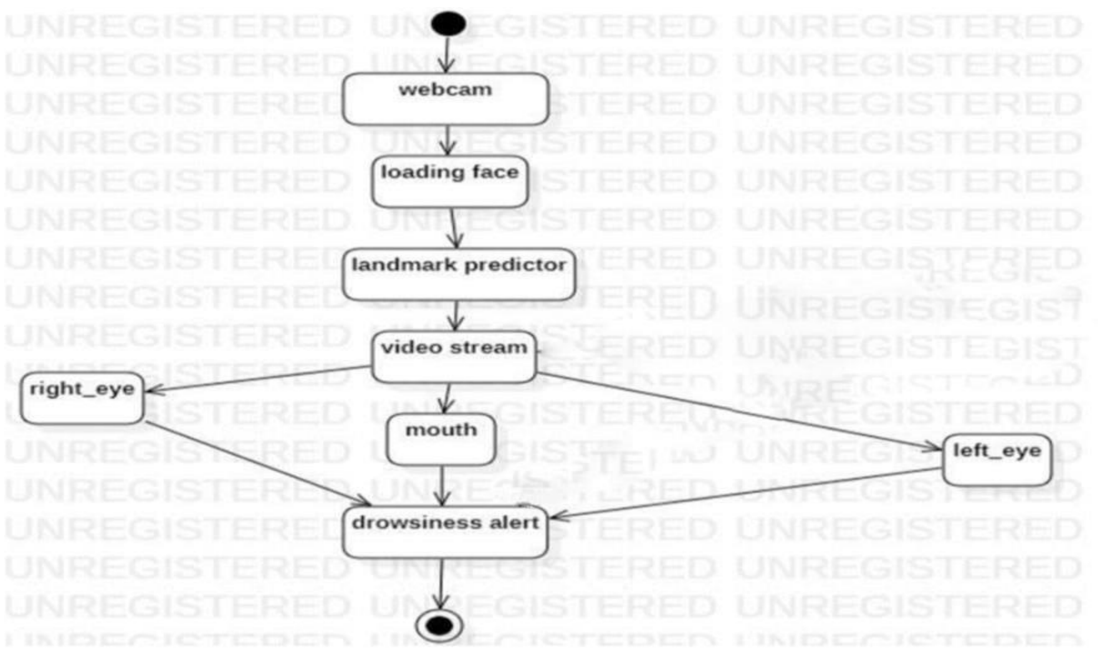


Fig 4.4: Activity diagram

4.2.5. Data Flow Diagram of System:

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships. Data flow diagrams visually represent systems and processes that would be hard to describe in a chunk of text. You can use these diagrams to map out an existing system and make it better or to plan out a new system for implementation. Visualizing each element makes it easy to identify inefficiencies and produce the best possible system.

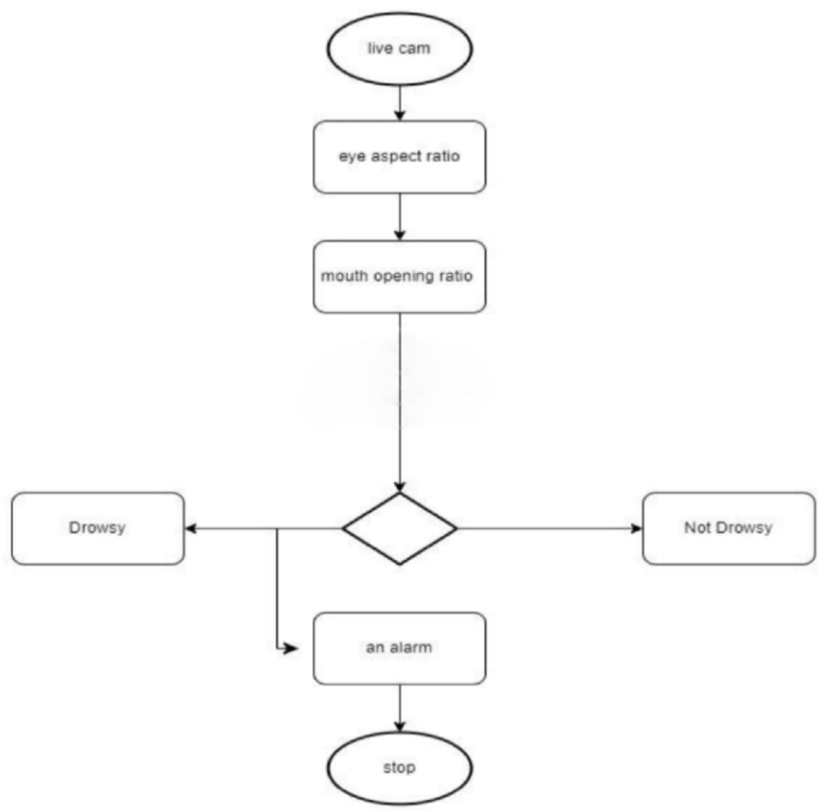


Fig 4.5: Data Flow diagram of system

5. IMPLEMENTATION

5.1 MODULES

Data Acquisition:

The video is recorded using webcam (Sony CMU-BR300) and the frames are extracted and processed in a laptop. After extracting the frames, image processing techniques are applied on these 2D images. Presently, synthetic driver data has been generated. The volunteers are asked to look at the webcam with intermittent eye blinking, eye closing, and yawning. The video is captured for 30 minutes duration.

Face Detection :

After extracting the frames, first the human faces are detected. Numerous online face detection algorithms are there. In this study, histogram of oriented gradients (HOG) and linear SVM method [10] is used. In this method, positive samples of descriptors are computed on them. Subsequently, negative samples (samples that do not contain the required object to be detected i.e., human face here) of same size are taken and HOG descriptors are calculated. Usually the number of negative samples is very greater than number of positive samples. After obtaining the features for both the classes, a linear SVM is trained for the classification task. To improve the accuracy of VM, hard negative mining is used. In this method, after training, the classifier is tested on the labeled data and the false positive sample feature values are used again for training purpose. For the test image, the fixed size window is translated over the image and the classifier computes the output for each window location. Finally, the maximum value output is considered as the detected face and a bounding box is drawn around the face. This non-maximum suppression step removes the redundant and overlapping bounding boxes.

Facial Landmark marking :

After detecting the face, the next task is to find the locations of different facial features like the corners of the eyes and mouth. Prior to that, the face images should be normalized in order to reduce the effect of distance from the camera, non-uniform illumination and varying image resolution. Therefore, the face image is resized to a width of 500 pixels and converted to gray scale image. After image normalization, ensemble of regression trees [11] is used to estimate the landmark positions on face from a sparse subset of pixel intensities. In this method, the sum of square error loss is optimized using gradient boosting learning. Different priors are used to find different structures. Using this method, the boundary points of eyes, and mouth are marked and the number of points for eyes mouth are given in Table I. The facial landmarks are shown in Fig 2. The red points are the detected landmarks for further

Table I: Facial landmark points

Parts	Landmark Points
Mouth	[13-24]
Right eye	[1-6]
Left eye	[7-12]
Nose	[25-28]

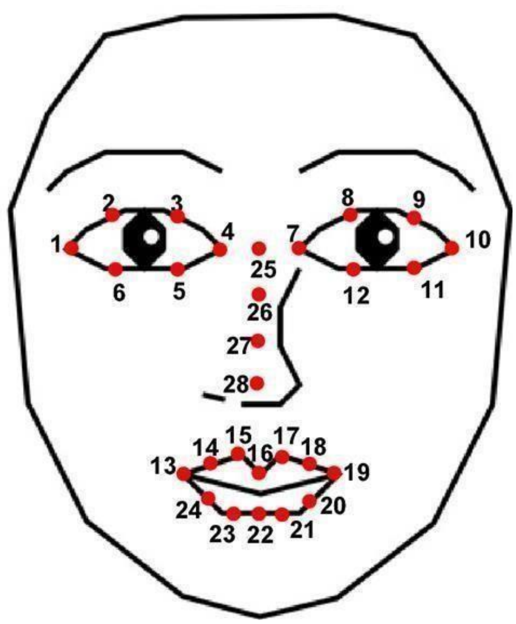


Fig : Facial landmarking points

Feature Extraction:

After detecting the facal land marks,the features are computed as described below. **Eye Aspect Ratio(EAR):** From the eye corner points, the eye aspect ratio is calculated as the ratio of different structures. Using this method, the boundary points of eyes, mouth and the central line of the nose are marked and the number of points for eye, mouth and nose are given in Table I. The facial landmarks are shown in Fig 2. The red points are the detected landmarks for further

Table I: Facial landmark points

Parts	Landmark Points
Mouth	[13-24]
Right eye	[1-6]
Left eye	[7-12]
Nose	[25-28]

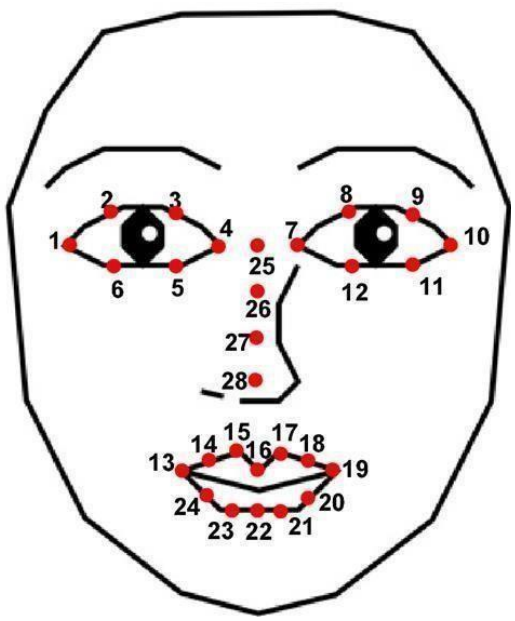


Fig : Facial landmarking points

Feature Extraction:

After detecting the facial landmarks, the features are computed as described below.

Eye aspect ratio (EAR): From the eye corner points, the eye aspect ratio is calculated as the ratio of height and width of the eye as given by

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

where represents point marked as i in facial landmark and is the distance between points marked as i and j. Therefore, when the eyes are fully open, EAR is high value and as the eyes are closed, EAR value goes towards zero. Thus, monotonically decreasing EAR values indicate gradually closing eyes and it's almost zero for completely closed eyes (eye blink). Consequently, EAR values indicate the drowsiness of the driver as eye blinks occur due to drowsiness.

Mouth opening ratio (MOR): Mouth opening ratio is a parameter to detect yawning during drowsiness. Similar to EAR, it is calculated as

$$\mathbf{MOR} = \frac{(P15-P23)+(P16-P22)+(P17-P21)}{3(P10-P12)}$$

As defined, it increases rapidly when mouth opens due to yawning and remains at that high value for a while due to yawn (indicating that the mouth is open) and again decreases rapidly threshold value. The average of EAR values is computed as the

towards zero. As yawn is one of the characteristics of drowsiness, MOR gives a measure regarding driver drowsiness.

Classification:

After computing all the features, the next task is to detect drowsiness in the extracted frames. In the beginning, adaptive thresholding is considered for classification. Later, machine learning algorithms are used to classify the data.

For computing the threshold values for each feature, it is assumed that initially the driver is in complete awake state. This is called setup phase. In the setup phase, the EAR values for first three hundred (for 10s at 30 fps) frames are recorded. Out of these three hundred initial frames containing face, average of 150 maximum values is considered as the hard threshold for EAR. The higher values are considered so that no eye closing instances will be present. If the test value is less than this threshold, then eye closing (i.e., drowsiness) is detected. As the size of eye can vary from person to person, this initial setup for each person will reduce this effect. Similarly, for calculating threshold of MOR, since the mouth may not be open to its maximum in initial frames (setup phase) so the threshold is taken experimentally from the observations. If the test value is greater than this threshold then yawn (i.e., drowsiness) is detected. After computing the threshold values, the system is used for testing. The system detects the drowsiness if in a test frame drowsiness is detected for at least one feature. To make this thresholding more realistic, the decision for each frame depends on the last 75 frames

If at least 70 frames (out of those 75) satisfy drowsiness conditions for at least one feature, then the system gives drowsiness detection indication and the alarm. To make this thresholding adaptive, another single threshold value is computed which initially depends on EAR average of 150 maximum values out of 300 frames in the setup phase. Then offset is determined heuristically and the threshold is obtained as offset subtracted from the average value. Driver safety is at risk when EAR is below this threshold. This EAR threshold value increases slightly with yawning upto a certain limit. As yawning is distributed over multiple frames, so yawning of consecutive frames are considered as single yawn and added once in the adaptive threshold. In a test frame, if EAR value is less than this adaptive threshold value, then drowsiness is detected and an alarm is given to the driver.

Table II illustrates this calculation for determining the adaptive threshold.

Table II: Threshold for the computed parameters

EAR from setup phase (average of 150 maximum values out of 300 frames)	0.34
Threshold=EAR- offset	0.34-.045=0.295
At Yawning,(MOR> 0.6)	Threshold=Threshold +0.002 *Max bound exist

Apart from using thresholding, the machine learning algorithms are used to detect drowsiness as well. The EAR and MOR, values are stored for the synthetic test data along with actual drowsiness annotation. Prior to classification, statistical analysis of the features has been done. At first, principal component analysis [12] is used to transform the feature space into an independent one. After transforming the feature values, the student's t-test is used to test whether the features are statistically significant for the two classes. As all these features are statistically significant at a 5% level of significance, all the features are used for classification using a Bayesian classifier [12], Fisher's linear discriminant analysis [12] and Support vector Machine [12].

5.2 Code:

```
from tkinter import * import
tkinter
from scipy.spatial import distance as dist
from  imutils  import  face_utils  import
numpy as np import imutils import dlib
import cv2 import winsound main =
tkinter.Tk()
main.title("Driver Drowsiness Monitoring")
main.geometry("500x400")

def EAR(drivereye):
   point1 = dist.euclidean(drivereye[1], drivereye[5])
point2 = dist.euclidean(drivereye[2], drivereye[4])
distance = dist.euclidean(drivereye[0], drivereye[3])
ear_aspect_ratio = (point1 + point2)/(2.0*distance)
return ear_aspect_ratio

def MOR(drivermouth):
   point = dist.euclidean(drivermouth[0], drivermouth[6])
point1 = dist.euclidean(drivermouth[2], drivermouth[10])
point2 = dist.euclidean(drivermouth[4], drivermouth[8])
Ypoint = (point1 + point2)/2.0     mouth_aspect_ratio =
Ypoint/point return
   mouth_aspect_ratio

def startMonitoring():
   pathlabel.config(text="Webcam Connected Successfully")
cap = cv2.VideoCapture(0) svm_predictor_path =
   'shape_predictor_68_face_landmarks.dat'
   EYE_AR_THRESH = 0.25
   EYE_AR_CONSEC_FRAMES = 10
    MOU_AR_THRESH = 0.75
COUNTER = 0     yawnStatus
= False yawns = 0 svm_detector =
   dlib.get_frontal_face_detector()
   svm_predictor = dlib.shape_predictor(svm_predictor_path)     (lStart,
lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
   (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"] (mStart, mEnd)
   = face_utils.FACIAL_LANDMARKS_IDXS["mouth"]
```

```
while True:        ret,
frame = cap.read() frame = imutils.resize(frame,
    width=640)        gray =
cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) prev_yawn_status
    = yawnStatus
rects = svm_detector(gray, 0)

    for rect in rects:
        shape = svm_predictor(gray, rect)
shape = face_utils.shape_to_np(shape)
leftEye = shape[lStart:lEnd] rightEye =
shape[rStart:rEnd]     mouth =
shape[mStart:mEnd]  leftEAR =
EAR(leftEye)          rightEAR=
EAR(rightEye) mouEAR=
        MOR(mouth)

        ear =(leftEAR + rightEAR)/2.0
leftEyeHull=cv2.convexHull(leftEye)
rightEyeHull=cv2.convexHull(rightEye)
mouthHull=cv2.convexHull(mouth)

        cv2.drawContours(frame,[leftEyeHull],-1,(0,255,255),1)
cv2.drawContours(frame,[rightEyeHull],-1,(0,255,255),1)
cv2.drawContours(frame,[mouthHull],-1,(0,255,255),1)

        if ear<EYE_AR_THRESH:
COUNTER += 1
cv2.putText(frame,"Eyes
Closed",(10,30),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
if COUNTER >= EYE_AR_CONSEC_FRAMES:
cv2.putText(frame,"DROWSINESS
ALERT!",(10,50),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
              winsound.Beep(200,2000)
else:
          COUNTER = 0
          cv2.putText(frame,"Eyes
          Open",(10,30),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,255,0),2)
          cv2.putText(frame,"EAR:
{:.2f}".format(ear),(480,30),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
```

```
        if mouEAR>MOU_AR_THRESH:
            cv2.putText(frame,"Yawning DROWSINESS
ALERT!",(10,70),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0 , 51 , 102) , 3 )
yawnStatus = True output_text = "Yawn Count: "+str(yawns+1)
            cv2.putText(frame,output_text,(10 ,130),cv2.FONT_HERSHEY_SIMPLEX,1 ,(51 ,
153 , 102) , 3 )
else:
            yawnStatus = False

        if prev_yawn_status == True and yawnStatus == False:
yawns+=1 cv2.putText(frame,"Yawn Count: "+str(yawns),(10
,130),cv2.FONT_HERSHEY_SIMPLEX,1 ,(51 , 153 , 102) , 3 )

      cv2.putText(frame,"Visual Behaviour & Machine Learning Drowsiness Detection @
Drowsiness",(370,470),cv2.FONT_HERSHEY_COMPLEX,0.6,(153,51,102),1)
cv2.imshow("Frame",frame)        key = cv2.waitKey(1)&0xFF     if key == ord("q"):
        break

   cap.release()
   cv2.destroyAllWindows()

font = ("times",16,"bold") title = Label(main, text="Driver Drowsiness Monitoring System
using Visual \n and Machine
Learning",    anchor=W,    justify=LEFT)
title.config(bg='black',             fg='white')
title.config(font=font)
title.config(height=3,
width=120) title.place(x=0,y=5)

font1 = ('times',14,'bold') upload = Button(main, text="Start Behaviour
Monitoring Using Webcam",
command=startMonitoring) upload.place(x=50,y=200)
upload.config(font= font1)



pathlabel = Label(main)
pathlabel.config(bg='DarkOrange1',fg='white')
pathlabel.config(font=font1) pathlabel.place(x=50,y=250)
```

main.config(bg='chocolate1') main.mainloop()

6. OUTPUT

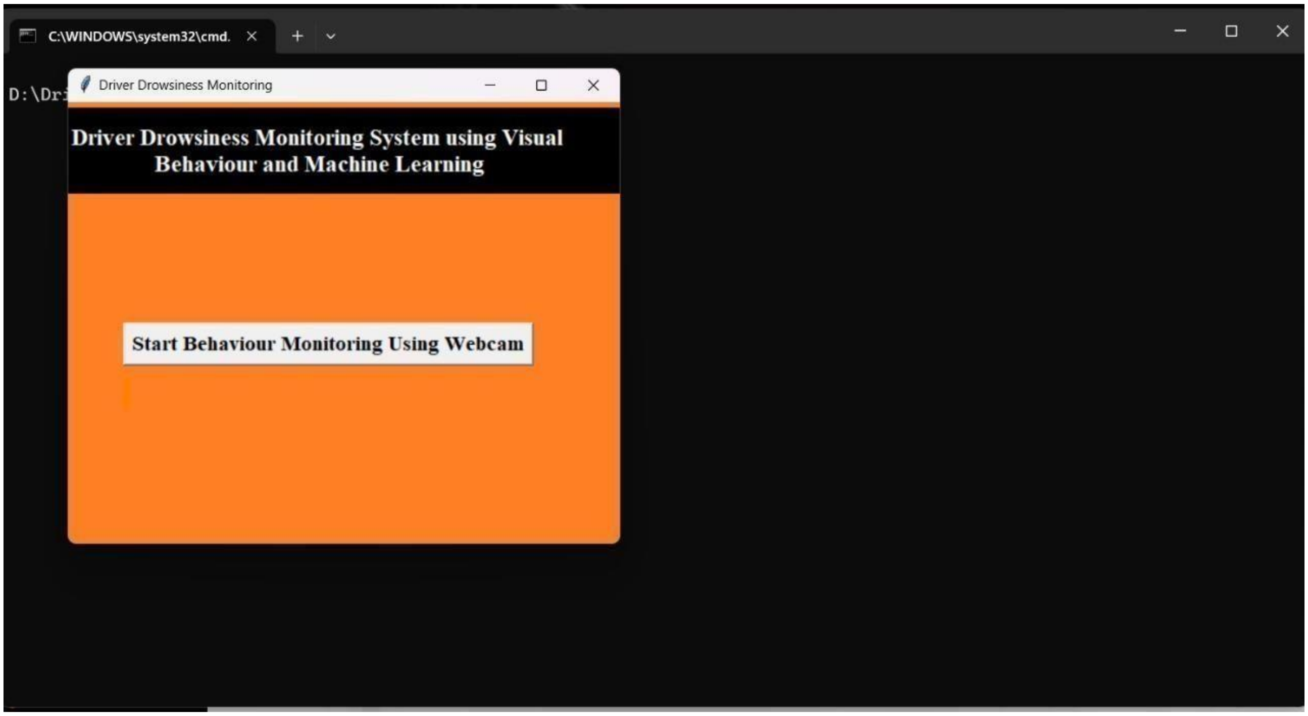


Fig 6.1 : start behaviour Monitoring Using Webcam

Fig 6.2 :Eyes Open

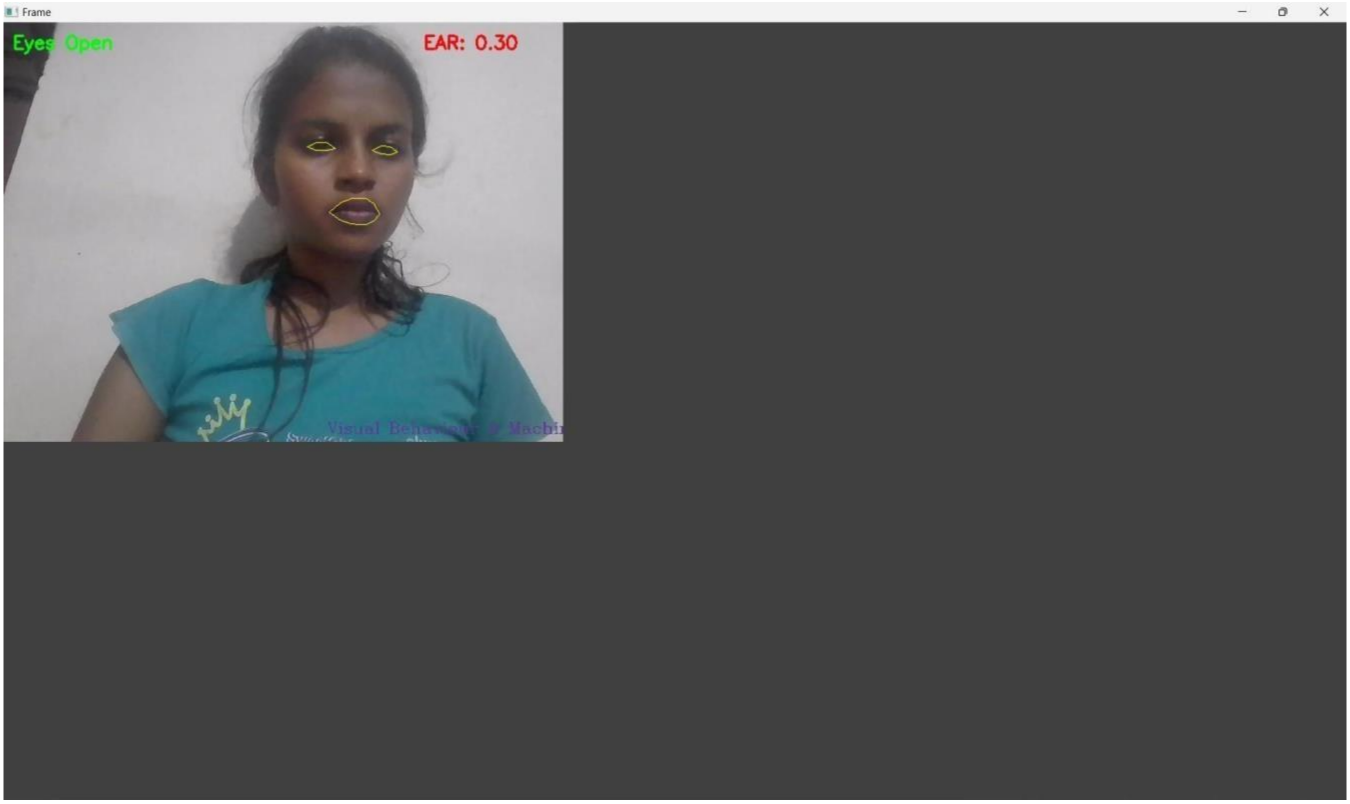
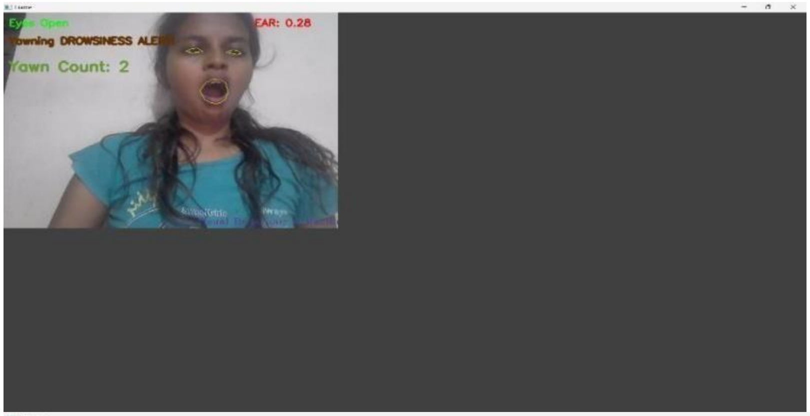


Fig 6.3:Yawning Drowsiness alert



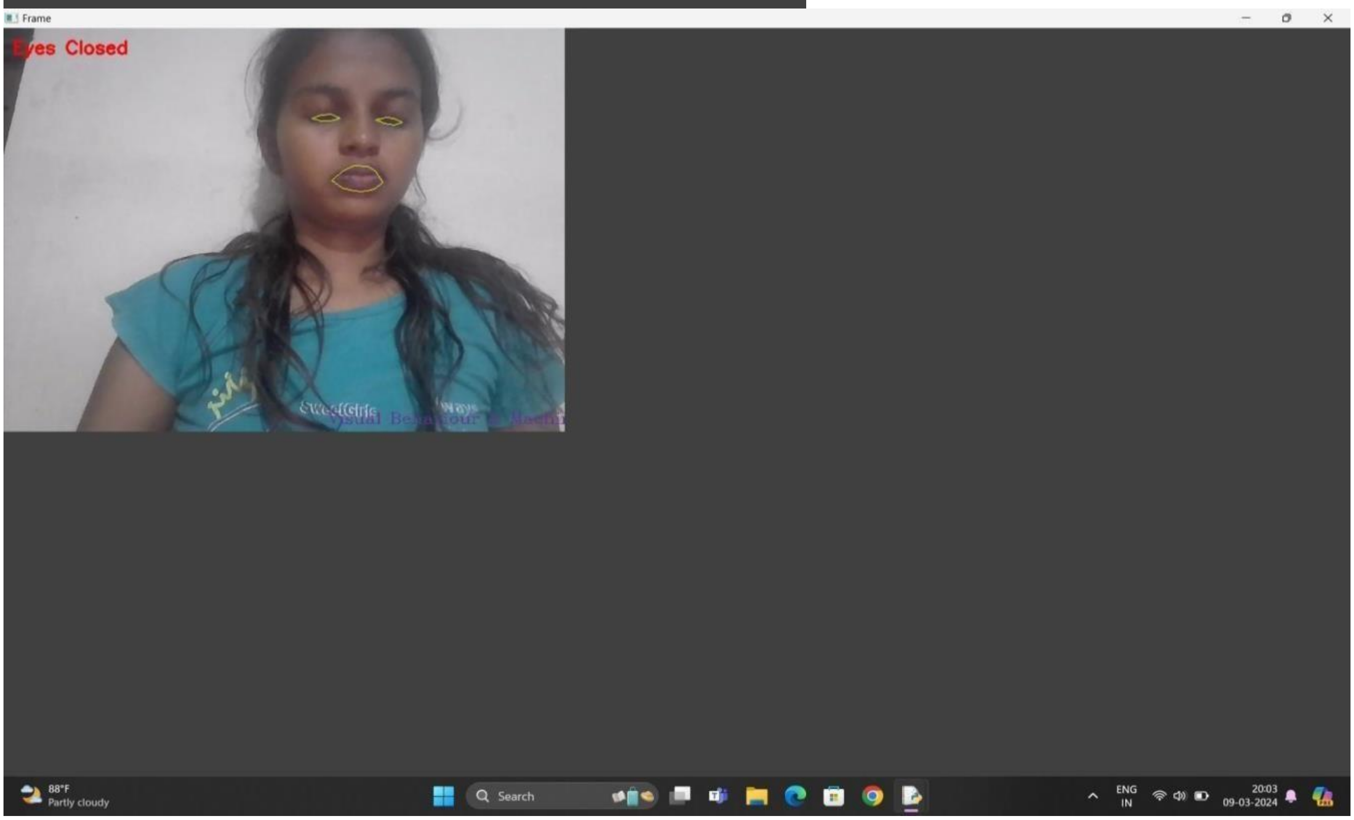


Fig 6.4 Eyes closed alert

TESTING

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

7.1 Types of Testing

Functional testing: It involves verifying the accuracy of the machine learning algorithms using various test scenarios

Performance testing: verifying that the visual behavior recognition system is working as intended**.**

user acceptance testing: it may also be conducted to ensure that the system is user-friendly and meets the needs of the target audience.

7.2 TEST CASES:

Test Description: Test cases for the alert for head bending

TEST ID	TEST CONDITION	TEST INPUT	TEST EXPECTED RESULTS	ACTUAL RESULTS	STATUS
1	To verify if our system can accurately detect yawning and issue an alert.	Facial movements of drivers	The system needs to be able to identify signs of driver fatigue, like their physical position, and respond by displaying an alert message and activating an audible warning signal.	Our system successfully identified driver drowsiness based on the detection of yawning and responded by displaying a "drowsiness alert" message and emitting a warning signal.	Pass

Table 7.2: Test Cases for yawning

Test Description: Test cases to Simulate various eye states

TE ST ID	TEST CONDITI ON	TES T INP UT	TEST EXPECT ED RESULTS	ACTUA L RESUL TS	STAT US
2	Confirm that the program correctly calculates the Eye Aspect Ratio (EAR).	open, closed, partially closed.	The EAR values should match the expected values for each eye state.	The EAR values matched the expected values for each eye state.	**Pass**

Table 7.2: Test Cases for eye status

8. LIMITATIONS

Identifying and addressing the limitations is an essential part of project development and research.

Here are some potential limitations of our project:

- **False Alarms:-** Some drowsiness monitoring systems may generate false alarms due to factors such as bright lights, sudden movements, or changes in the driver's posture.

- **Reliance on Technology:-** Drowsiness monitoring systems rely on technology, which can malfunction or fail. Drivers should always be aware of their own level of drowsiness and take action if necessary, even if the system is working properly.

- **Cost:-** Drowsiness monitoring systems can be expensive, especially High-end models with advance features.

9. FUTURE SCOPE

The Future Scope of this project includes the following

- If the user doesn’t respond for a longer time, emergency call or message will be sent automatically.
- The alcoholic sensor can be integrated for drunk drivers.
- This project, if integrated with car, automatic speed control should be imparted if the driver is found sleeping.

10. CONCLUSION

In this project, a low-cost, real-time driver drowsiness monitoring system has been proposed based on visual behavior and machine learning. Here, visual behavior features like eye aspect ratio and mouth opening ratio are computed from the streaming video, captured by a webcam. An adaptive thresholding technique has been developed to detect driver drowsiness in real-time. The developed system works accurately with the generated synthetic data. Subsequently, the feature values are stored and machine learning algorithms have been used for classification. Bayesian classifier, HOG, and SVM have been explored here. It has been. observed that HOG and SVM outperform the Bayesian classifier. The sensitivity of HOG and SVM is 0.896 and 0.956 respectively where the specificity is 1 for both. As HOG and SVM give better accuracy, work will be carried out to implement them in the developed system for the classification (i.e., drowsiness detection) online. Also, the system will be implemented to make it portable for car system.

11. REFERENCES

[1]. R. Kumaran, R. Deepak Kumar, M. Karthick Kumar, B. Rahul, 2020, Driver Drowsiness Monitoring System using Visual Behaviour and Machine Learning, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) RTICCT– 2020 (Volume 8 – Issue 12),

[2]. A. Sengupta, A. Dasgupta, A. Chaudhuri, A. George, A. Routray and R. Guha, "A Multimoda System for Assessing Alertness Levels Due to Cognitive Loading," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 25, no. 7, pp. 1037-1046, July 2017, doi:10.1109/TNSRE.2017.2672080.

[3]. R. Kumaran, R. Deepak Kumar, M. Karthick Kumar, B. Rahul, 2020, Driver Drowsiness Monitoring System using Visual Behavior and Machine Learning, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) RTICCT – 2020 (Volume 8 – Issue 12),

[4]. National Safety Council Drivers are Falling Asleep Behind the Wheel. [(accessed on 10 May 2021)]. Available online: https://www.nsc.org/road-safety/safety-topics/fatigued-driving

[5]. Beirness D.J., Simpson H.M., Desmond K., The Road Safety Monitor 2004: Drowsy Driving Drowsy Driving. 2005. [(accessed on 2 March 2022)]. Available online: http://worldcat.org/isbn/0920071473

[6]. Knapik M., Cyganek B. Driver’s fatigue recognition based on yawn detection in thermal images. *Neurocomputing.* 2019;**338**:274–292. doi: 10.1016/j.neucom.2019.02.014. [CrossRef] [Google Scholar]

[7]. Bamidele A., Kamardin K., Syazarin N., Mohd S., Shafi I., Azizan A., Aini N., Mad H. Nonintrusive driver drowsiness detection based on face and eye tracking. *Int J. Adv. Comput. Sci.Appl.* 2019;**10**:549doi: 10.14569/IJACSA.2019.0100775. [CrossRef] [Google Scholar]

[8]. Abbas Q., Alsheddy A. Driver fatigue detection systems using multi-sensors, smartphone, and cloud-based computing platforms: A comparative analysis. *Sensors.* 2021;**21**:56. doi: 10.3390/s21010056. [PMC free article] [PubMed] [CrossRef] [Google Scholar]