As per UGC guidelines an electronic bar code is provided to secure your paper

International Journal for Modern Trends in Science and Technology
Volume 12, Issue 07, pages 49-65.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue06.html
DOI: https://doi.org/10.5281/zenodo.20739595

Real Time Drowsiness Detection System Using Open Computer Vision

R. Uma Aruna Devi, Dodda Tilak Kumar

Department of Information Technology , Swarnandhra College of Engineering & Technology, Seetharamapuram, Narsapur, Andhra Pradesh, INDIA.

To Cite this Article
R. Uma Aruna Devi & Dodda Tilak Kumar (2026). Real Time Drowsiness Detection System Using Open Computer Vision. International Journal for Modern Trends in Science and Technology, 12(07), 49-65. https://doi.org/10.5281/zenodo.20739595

Article Info
Received: 06 June 2026; Revised: 29 June 2026; Accepted: 01 July 2026.

KEYWORDS

Computer Vision, Driver Safety, Drowsiness Detection, Eye Aspect Ratio (EAR), OpenCV, Dlib, Facial Landmarks.

ABSTRACT

Driver fatigue and drowsiness are among the leading causes of road accidents worldwide, often resulting in severe injuries and fatalities. This project presents the design and implementation of a Real-Time Drowsiness Detection System aimed at enhancing road safety by monitoring driver alertness. The system utilizes a non-intrusive, vision-based approach to detect fatigue symptoms through the analysis of facial expressions and eye movements. The core of the system is built using Python, OpenCV, and the Dlib library. It employs a webcam to capture high-resolution video of the driver, which is processed frame-by-frame. The detection process involves three primary stages: identifying the driver's face using Haar Cascade Classifiers, mapping 68 specific facial landmarks, and calculating the Eye Aspect Ratio (EAR). By monitoring the EAR, the system can distinguish between normal blinking and prolonged eye closure. If the EAR remains below a predefined threshold (0.3) for a specific number of consecutive frames (48 frames), the system triggers an immediate audible alarm and a visual alert to warn the driver. Experimental results indicate that the system operates with approximately 80% accuracy under standard conditions. The proposed solution is cost-effective, requires minimal hardware, and operates in real-time, making it a viable safety feature for both commercial and personal vehicles. Future enhancements may include the integration of infrared sensors for low-light performance and cloud-based notification systems for fleet management.

1. INTRODUCTION

The modern transportation landscape is a cornerstone of global economy and personal mobility, yet it remains one of the most hazardous environments for human life. Among the various factors contributing to vehicular accidents, human error is consistently identified as the primary catalyst. Within the spectrum of human error, driver fatigue and drowsiness represent a silent epidemic. Drowsiness is a complex physiological state that leads to a progressive decline in cognitive performance, vigilance, and motor coordination. Unlike many mechanical failures, fatigue is an internal process that often goes unnoticed by the driver until a critical failure in judgment occurs.

The World Health Organization (WHO) and various international transport agencies have highlighted that sleep-deprived driving can be as dangerous as driving under the influence of alcohol. The phenomenon of "microsleep"—a brief, involuntary episode of sleep lasting anywhere from a fraction of a second to thirty seconds—is particularly lethal at high speeds. At a speed of 100 km/h, a vehicle can travel over 80 meters during a mere three-second lapse in consciousness, often resulting in high-impact collisions, lane departures, and fatalities. As globalization increases the demand for long-haul logistics and 24-hour transport services, the risk of fatigue-related incidents continues to escalate, challenging the resilience of global public health and safety systems.

Historically, efforts to mitigate driver drowsiness have relied on two main approaches: manual surveillance and vehicle-based sensory systems. Manual surveillance, such as self-monitoring or co-driver observation, is highly subjective and prone to human error, as individuals often overestimate their level of alertness. On the other hand, traditional vehicle-based systems, such as Lane Departure Warning Systems (LDWS) or Steering Angle Sensors, monitor the *consequences* of drowsiness rather than the *cause*. These systems typically trigger an alert only after a vehicle has already begun to drift or exhibit erratic movement, which may be too late to prevent a disaster.

Furthermore, early physiological diagnostic methods involved wearable sensors that monitored heart rate variability (ECG) or brainwave patterns (EEG). While these medical-grade approaches are highly accurate, they are fundamentally impractical for daily automotive use. The intrusive nature of wearing electrodes or specialized headsets causes driver discomfort and distractions, leading to low adoption rates. There exists a significant technological gap for a system that is both "non-intrusive" and "proactive"—one that can detect the onset of fatigue by analyzing the driver's physiological state in real time without physical contract or reliance on vehicle telemetry.

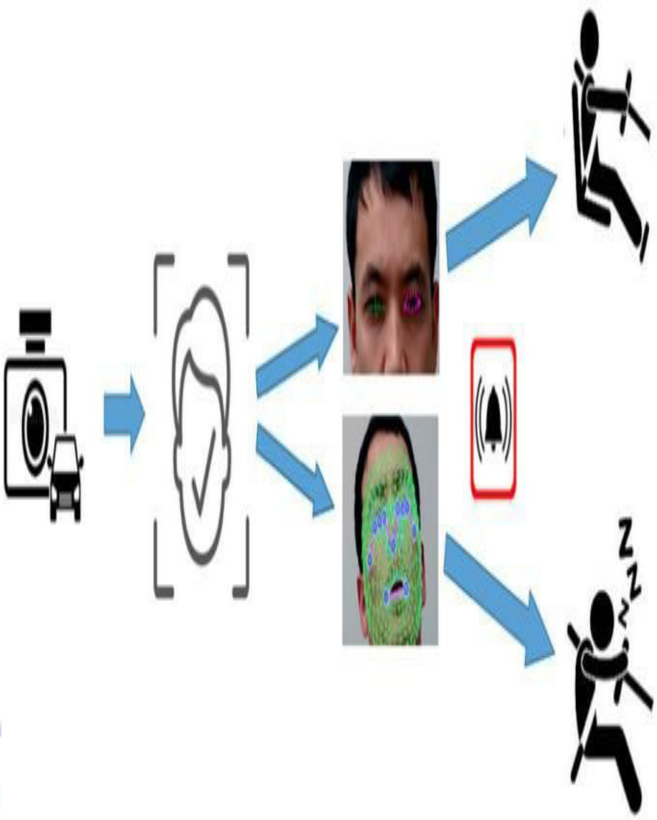

Fig 1.1: Drowsiness Detection System

Additionally, a risk classification mechanism categorizes predicted outbreaks into Low, Medium, and Highrisk levels, enhancing interpretability and aiding healthcare decision-making. By combining multiple deep learning techniques, the system addresses the limitations of traditional approaches and provides a scalable, efficient, and near real-time solution.

Overall, this project contributes to strengthening healthcare systems by enabling early detection, improving diagnostic accuracy, and supporting timely interventions. With ongoing advancements in AI and deep learning, such integrated solutions have the potential to significantly improve infectious disease monitoring and control.

1.1 OVERVIEW

The rapid advancement of technology has significantly improved human life, particularly in the field of transportation. However, this progress is shadowed by a persistent and growing concern: road safety. Statistics from global transport authorities consistently highlight that human error is the primary cause of road accidents. Among these errors, driver fatigue and drowsiness are the most critical yet preventable factors.

Drowsiness is a physiological state where a person's alertness and consciousness level decrease, often leading to involuntary sleep. For a driver, even a few seconds of "microsleep" can be fatal. Unlike mechanical failures, drowsiness is a gradual process, making it difficult for the driver to judge their own level of impairment accurately. This project, the Real-Time Drowsiness Detection System, is designed as a safety intervention tool that uses computer vision to monitor driver behavior and provide life-saving alerts before an accident occurs.

1.2 BACKGROUND OF THE PROBLEM

In the early days of automotive safety, the focus was primarily on "passive safety" features—mechanisms designed to protect occupants during a crash, such as seat belts, airbags, and reinforced vehicle chassis. As technology evolved, the industry shifted toward "active safety," which aims to prevent accidents from happening in the first place.

Traditional methods for detecting driver fatigue were often intrusive or unreliable. Some early systems used wearable sensors to monitor heart rate or brain activity (EEG), but these proved impractical for daily use as they caused discomfort to the driver. Other systems focused on vehicle telemetry, such as steering wheel patterns or lane deviation. However, these systems often trigger too late—only after the vehicle has already begun to drift dangerously.

With the emergence of high-speed processing and advanced image processing libraries like OpenCV and Dlib, it is now possible to implement "non-intrusive" monitoring. By using a standard camera to track facial landmarks and eye movements, we can detect the onset of fatigue based on physiological signs, providing a more reliable and comfortable solution for the driver.

1.3 PROBLEM STATEMENT

The core problem addressed by this project is the high rate of traffic accidents caused by driver exhaustion. Several factors contribute to this issue:

- Long-Distance Travel: Commercial truck drivers and long-distance travelers often drive for extended hours, leading to natural physical and mental exhaustion.
- Night-Time Driving: Circadian rhythms naturally lower alertness during late-night and early-morning hours.

Lack of Real-Time Feedback: Most drivers overestimate their ability to "push through" tiredness. Without an objective system to monitor their state, they may not realize they are at risk until it is too late

There is a clear need for a system that is:

- Real-time: Able to process video and alert the driver within milliseconds.
- Non-intrusive: Does not require the driver to wear any equipment.

Cost-effective: Can be implemented using standard hardware available in most modern vehicles or via a simple aftermarket camera and laptop/embedded system.

1.4 MOTIVATION

The primary motivation behind this project is the preservation of human life. While modern cars are equipped with airbags and anti-lock braking systems (ABS) to protect passengers *during* an accident, there is a growing need for "active safety" technology that prevents the accident from occurring in the first place. A system that can wake a driver up or alert them to pull over before they lose control of the vehicle can significantly reduce the number of injuries and fatalities on the road.

1.5 OBJECTIVES OF THE SYSTEM

The main goal of this project is to create a robust application that can accurately detect drowsiness in various environments. The specific objectives are:

- To detect the face of the driver in a video stream using Haar Cascade Classifiers.
- To map 68 facial landmarks to identify the precise location of the eyes.
- To calculate the Eye Aspect Ratio (EAR) to determine if the eyes are open or closed.
- To track the duration of eye closure and trigger a loud audible alarm and a visual "DROWSINESS ALERT" if the eyes remain closed for a predefined period (e.g., 48 consecutive frames).
- To ensure the system operates in real-time with minimal latency.

1.6 SCOPE OF THE PROJECT

This project covers the development of the software logic using Python. It utilizes a standard webcam for image acquisition and focuses on the facial region of the driver. The system is designed to work in a variety of indoor and outdoor lighting conditions. While the current scope is focused on detection and alerting, the framework is built to allow for future integrations, such as cloud-based notifications or vehicle speed control modules.

1.7 METHODOLOGY

The system follows a structured pipeline:

- **Image Capture:** Frames are pulled from a live video feed.
- **Preprocessing:** Each frame is converted to grayscale to reduce computational complexity.
- **Facial Landmark Detection:** The Dlib library is used to find 68 coordinates on the face.
- **Eye Aspect Ratio (EAR):** The vertical and horizontal distances of the eyes are compared.
- **Alerting:** If the EAR falls below a threshold (0.3) for a specific time, the system sounds an alarm.

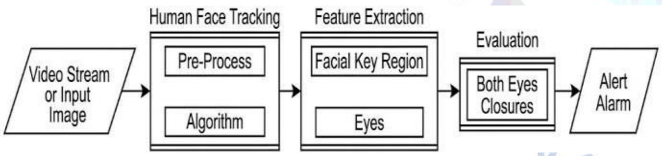

Fig 1.2: Block diagram for the purposed system

2. LITERATURE SURVEY

LITERATURE REVIEW

Drowsy driving is one of the common causes of fatalities in car accidents. Truck drivers that travel for lengthy periods of time (especially at night), long-distance bus drivers, and overnight bus drivers are more vulnerable to this condition. Passengers in every country face the nightmare of drowsy drivers. Fatigue-related traffic accidents result in a substantial number of injuries and deaths each year. As a result, due to its wide practical application, detecting and indicating driver drowsiness is a hot topic of research. In general, there are three sorts of approaches for detecting drowsy drivers: vehicle-based, behavioural-based, and physiological based. A number of parameters such as steering wheel movement, accelerator or brake pattern,6 vehicle speed, lateral acceleration, deviations from lane position, and so on are continuously monitored in the vehicle-based method. Driver drowsiness is defined as the detection of any abnormal change in these parameters.

2.1 INTRODUCTION TO DRIVER FATIGUE DETECTION

The study of driver fatigue has evolved over several decades. Early research focused primarily on medical and physiological markers. Researchers identified that the transition from alertness to sleep involves measurable changes in the body, such as brain wave patterns, heart rate variability, and muscle tone. However, applying these findings to a real-world driving environment posed significant engineering challenges. Literature in this field is generally categorized into three approaches: Physiological-based, Vehicle-based, and Behavioral-based methods.

2.2 PHYSIOLOGICAL-BASED METHODS

Physiological methods are considered the "Gold Standard" for detecting drowsiness because they measure internal biological signals.

- **Electroencephalogram (EEG):** Research by Sahayadhas et al. (2012) showed that alpha and theta wave activity increases significantly when a driver is drowsy. While highly accurate, the requirement for scalp electrodes makes it impractical for daily driving.
- **Electrocardiogram (ECG):** Monitoring heart rate variability (HRV) can indicate autonomic nervous system fatigue. Studies have shown that as a driver tires, the heart rate becomes more regular with less variability.
- **Drawbacks:** The primary limitation cited in literature is the "intrusive" nature of these sensors. Drivers find it uncomfortable to wear headsets or chest straps while operating a vehicle.

2.3 VEHICLE-BASED (SENSORY) METHODS

These methods monitor the behavior of the vehicle rather than the driver.

- **Steering Wheel Movement (SWM):** Researchers found that drowsy drivers make fewer but larger steering corrections. Systems developed by companies like Mercedes-Benz (Attention Assist) use steering sensors to detect these patterns.

- **Lane Departure Warning Systems (LDWS):** Using external cameras, these systems track the vehicle's position relative to lane markings. If the car drifts without a turn signal, an alert is triggered.
- **Drawbacks:** Literature indicates these systems are highly dependent on external factors such as road markings, weather conditions, and vehicle speed. Furthermore, they only detect drowsiness *after* a dangerous driving maneuver has already occurred.

2.4 BEHAVIORAL-BASED (VISION) METHODS

This is the domain of your project. Behavioral methods use computer vision to analyze the driver's physical actions.

- **Yawn Detection:** Monitoring the frequency and duration of mouth opening.
- **Head Pose Estimation:** Tracking "nodding" motions associated with falling asleep.
- **Eye Tracking:** This is the most reliable behavioral indicator. The "PERCLOS" (Percentage of Eye Closure) measure was established by Wierwille et al. (1994) as a valid metric for drowsiness.

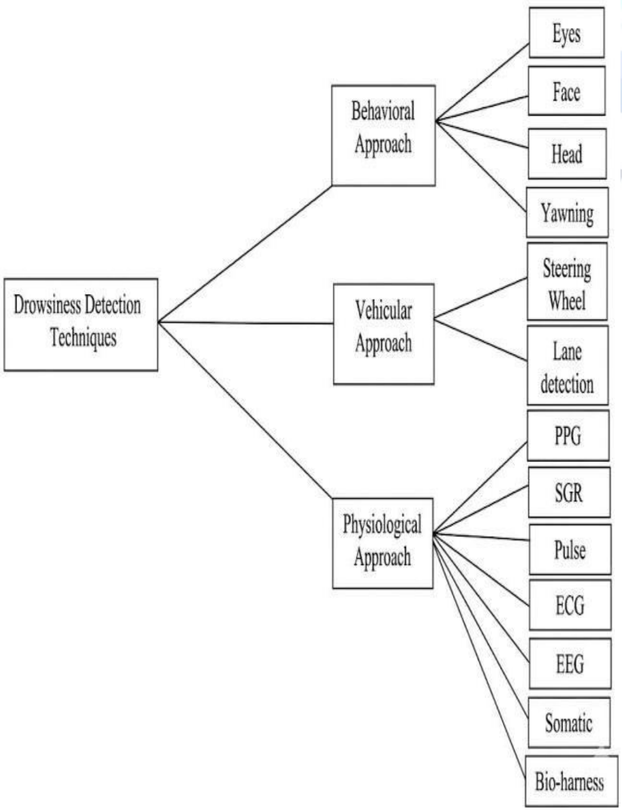

Fig 2.1: Drowsiness Detection Methods

2.5 EVOLUTION OF FACE DETECTION ALGORITHMS

To monitor eyes, the system must first find the face. The literature highlights two major milestones:

- **Viola-Jones Framework (2001):** Introduced the Haar Cascade Classifier. It revolutionized real-time detection by using "Integral Images" and "AdaBoost" to quickly identify facial features. This is the method utilized in your current OpenCV implementation.
- **HOG + Linear SVM:** Histogram of Oriented Gradients (HOG) combined with Support Vector Machines offered better accuracy in varying lighting, though it requires more computational power than Haar Cascades.

2.6 EXISTING SYSTEM / PURPOSED SYSTEM

1. Existing System

The existing systems for monitoring driver safety can generally be categorized into three main approaches:

- **Physiological-Based Methods:** These systems monitor internal body signals like brain activity (EEG), heart rate (ECG), or pulse.

Drawbacks: These are highly "intrusive," requiring drivers to wear uncomfortable sensors, headsets, or chest straps, which limits their daily practicality.

- **Vehicle-Based (Sensory) Methods:** These monitor the vehicle's behavior, such as steering wheel movements or lane departures.

Drawbacks: They are often unreliable because they depend on external factors like clear road markings or weather. More importantly, they only detect drowsiness *after* a dangerous driving maneuver has already happened.

- **Traditional Vision-Based Systems:** Some existing computer vision models used older algorithms like the
- **Viola-Jones Framework (Haar Cascades)** solely for simple bounding box detection.

Drawbacks: These often lacked the precision of modern landmark mapping and struggled with variations in head tilt or distance from the camera

2. Proposed System

Your project proposes a **non-intrusive, vision-based system** that uses high-speed landmark detection to monitor driver alertness in real-time.

- **Core Technology:** The system is built using **Python, OpenCV, and the Dlib library**.
- **Advanced Landmark Mapping:** Instead of just finding a "face box," it uses the **Dlib 68-point predictor** to map specific (x, y) coordinates on the face.
- **Mathematical Model (EAR):** The system's "brain" is the **Eye Aspect Ratio (EAR)** algorithm.
  - ✓ It calculates the ratio between the vertical and horizontal distances of the eyes.
  - ✓ This makes the system "**scale-invariant,**" meaning it remains accurate whether the driver is close to or far from the camera.
- **Temporal Thresholding:** To distinguish between a natural blink and actual drowsiness, the system uses a frame counter.

If the EAR stays below a threshold (e.g., 0.3) for a specific duration (e.g., 48 consecutive frames), it triggers an alert.

- **Multi-Threaded Alerting:** The proposed system uses a separate thread for the audible alarm to ensure the video feed never freezes while the alarm is sounding.

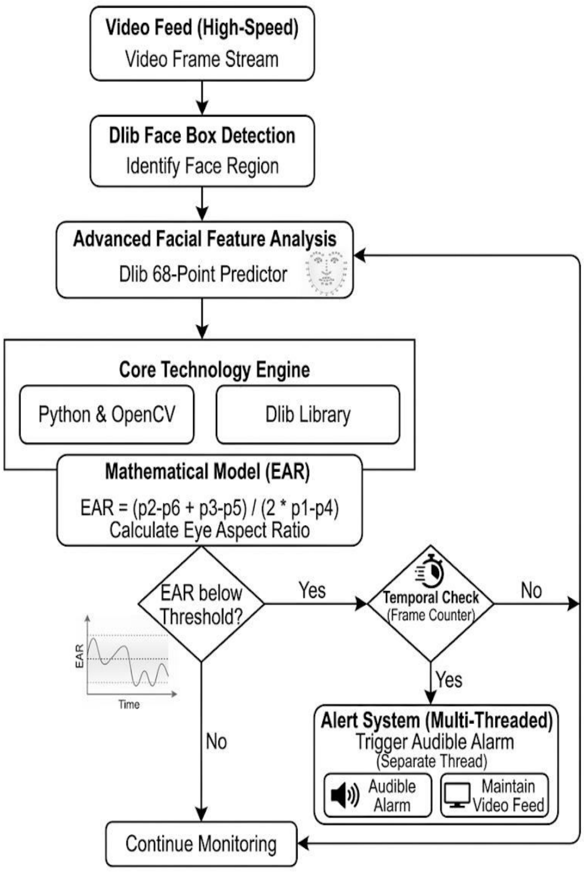

Fig 2.2: Purposed System

2.7 FACIAL LANDMARK LOCALIZATION (DLIB)

The shift from simple "bounding boxes" to "landmark points" allowed for much finer analysis.

- **Kazemi and Sullivan (2014):** They proposed the One Millisecond Face Alignment algorithm, which is the foundation of the **Dlib 68-point predictor**. This algorithm uses an ensemble of regression trees to estimate landmark positions in real-time.
- The literature emphasizes that by identifying specific points (36-47 for eyes), we can calculate geometric shapes regardless of the face's distance from the camera.

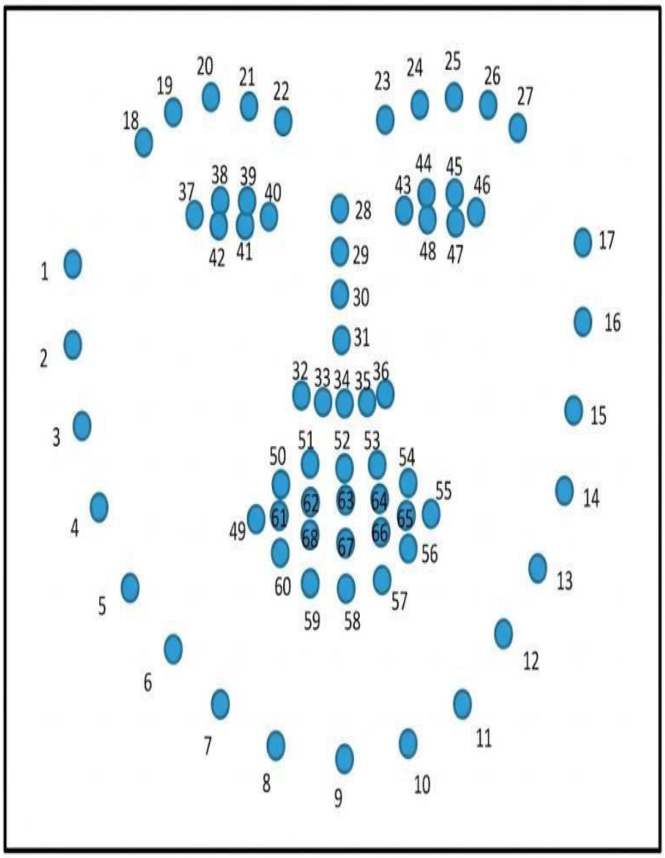

Fig 2.3: Dlib 68-Point Predictor

2.8 THE EYE ASPECT RATIO (EAR) ALGORITHM

The EAR was popularized by **Tereza Soukupová and Jan Čech (2016)** in their paper "Real-Tme Eye Blink Detection Using Facial landmarks."

- **The Concept:** Traditional blink detection relied on image templates or intensity changes. Soukupová proposed a simple ratio of the distances between vertical and horizontal eye landmarks.

- **Reliability:** The EAR is "aspect ratio-invariant," meaning it remains consistent even if the driver moves their head or changes their distance from the camera. This is a critical advantage noted in recent technical literature.

2.9 Comparative Analysis of Existing Systems

Method	Accuracy	Intrusiveness	Real-Time Capability	Cost
EEG (Brain Waves)	Very High	High (Wearable)	Low	High
Steering Patterns	Medium	None	High	Medium
Blink Rate (EAR)	High	None	Very High	Low
Heart Rate (ECG)	Medium	High (Wearable)	Medium	Medium

2.10 SUMMARY OF RESEARCH GAPS

While many high-end vehicles (e.g., Tesla, Volvo) have integrated these systems, they are often proprietary and tied to expensive hardware. The literature suggests a gap in "Aftermarket Solutions"—low-cost, software-based systems that can be installed on any vehicle using a smartphone or a basic laptop. Your project addresses this gap by using open-source tools (Python, OpenCV) to create an accessible safety solution.

3. PROPOSED METHODOLOGY

The proposed system is a real-time, non-intrusive driver monitoring solution designed to detect drowsiness by analyzing facial features and eye behavior from live webcam video streams. It operates without requiring any physical sensors attached to the driver, making it practical and comfortable for real-world driving conditions. By using computer vision techniques, the system continuously observes the driver's facial expressions to determine alertness levels.

The core idea of the system is that drowsiness produces observable visual symptoms, particularly in the eyes. When a driver becomes tired, changes such as slow blinking, longer eye closures, and reduced eyelid opening begin to appear. These visual cues can be captured through a standard camera and analyzed frame by frame to assess the driver's level of alertness.

To begin the detection process, the system first captures live video from a webcam positioned in front of the driver. Each video frame is processed individually to ensure continuous monitoring. This real-time processing allows the system to respond immediately when signs of drowsiness are detected.

The next step involves detecting the driver's face within each frame. Face detection algorithms identify the region of interest containing the face, which ensures that further analysis is focused only on relevant areas. Accurate face detection is essential because it directly affects the precision of subsequent landmark extraction and eye analysis.

After the face is detected, the system applies a 68-point facial landmark detection model, commonly implemented using the Dlib library. This model identifies specific key points on the face, including the eyes, eyebrows, nose, mouth, and jawline. Among these landmarks, particular attention is given to the points surrounding both eyes.

Using the eye landmarks, the system calculates the Eye Aspect Ratio (EAR), which serves as a numerical indicator of eye openness. The EAR is computed by comparing the vertical distances between specific eye landmark points to the horizontal width of the eye. This ratio provides a consistent and compact representation of whether the eye is open or closed.

When the eyes are open, the vertical distances are relatively large compared to the horizontal width, resulting in a stable EAR value. However, when the eyes close, the vertical distances decrease significantly while the horizontal width remains nearly constant. As a result, the EAR value drops sharply, making it an effective metric for detecting blinking and eye closure.

The system distinguishes between normal blinking and drowsiness by analyzing the duration for which the EAR remains below a predefined threshold. For example, if the EAR falls below a value such as 0.3 for a sustained period, such as 48 consecutive frames, the system interprets this as prolonged eye closure rather than a normal blink.

Once prolonged eye closure is detected, the system classifies the state as drowsiness. At this stage, an alert mechanism is activated to warn the driver. The alert may be in the form of an audible alarm, which is intended to immediately draw the driver's attention and encourage them to regain focus.

Overall, the proposed system combines face detection, facial landmark extraction, and EAR-based analysis to provide a reliable and low-cost method for real-time drowsiness detection. Because it relies solely on visual data from a camera and lightweight computations, it is suitable for practical deployment in vehicles and offers an effective solution to enhance road safety.

3.1 SYSTEM ARCHITECTURE

The system architecture is organized as a sequence of interconnected modules that work together to process live video data and make decisions in real time. Each module performs a specific task, and the output of one module becomes the input for the next. This structured flow ensures accurate and efficient detection of driver drowsiness.

The first module is Video Acquisition. In this stage, a webcam continuously captures the driver's video stream in real time. The camera is positioned to clearly capture the driver's face, especially the eye region, which is essential for accurate monitoring. The live video feed provides the raw input for the entire system.

Once the video stream is captured, it is divided into individual frames for processing. Each frame represents a single image extracted from the continuous video. By processing frames one by one, the system can analyze facial features dynamically and respond quickly to any signs of drowsiness.

The next module is Frame Preprocessing. In this step, each frame is resized to a standard dimension to reduce computational complexity. The frame is also converted from color to grayscale, which simplifies image data and improves processing speed without losing important structural details needed for detection.

After preprocessing, the Face Detection module is applied. A Haar Cascade Classifier is used to detect and isolate the face region from the frame. By focusing only on the detected face area, the system eliminates unnecessary background information and improves both accuracy and efficiency.

Following face detection, the Facial Landmark Detection module comes into operation. Using Dlib's 68-point facial landmark predictor, the system identifies key facial feature points. These landmarks include specific points around the left and right eyes, which are crucial for computing eye-related measurements.

With the eye landmarks identified, the system proceeds to the EAR Computation module. The Eye Aspect Ratio (EAR) is calculated using the coordinates of the eye landmark points. The EAR value represents the degree of eye openness by comparing vertical eye distances to the horizontal eye width.

The system calculates the EAR for both the left and right eyes and then determines an average value to represent the overall eye state. When the eyes are open, the EAR remains relatively stable. When the eyes close, the EAR drops significantly, providing a clear numerical indication of eye closure.

The next stage is the Drowsiness Decision Logic module. In this step, the system checks whether the EAR value falls below a predefined threshold, such as 0.3, and remains below that level for a specific number of consecutive frames, such as 48. This condition helps differentiate between a normal blink and prolonged eye closure associated with drowsiness.

Finally, the Alert Generation module is activated when the drowsiness condition is satisfied. The system immediately triggers a visual warning and an audible alarm to alert the driver. This completes the architectural flow, where the webcam sends frames to the processing module, the face is detected, landmarks are extracted, EAR is computed, and the decision logic determines whether to issue an alert based on sustained low EAR values.

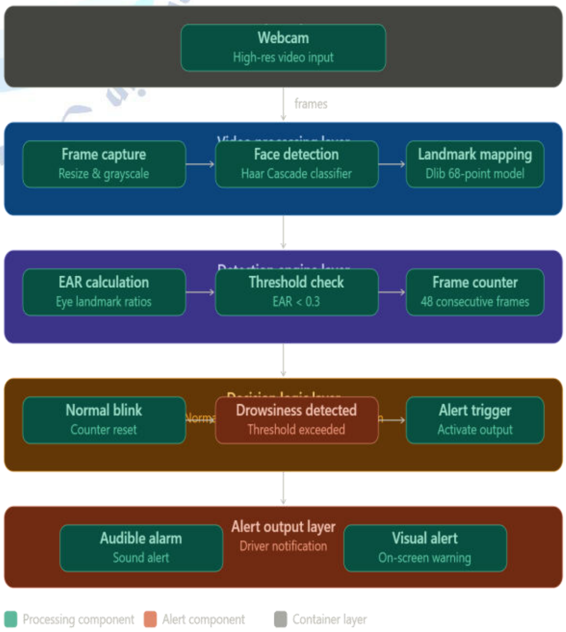

Fig 3.1 Proposed System Architecture

The proposed Driver Drowsiness Detection System follows a layered and modular processing architecture that transforms raw video input into real-time safety alerts. The system is designed to ensure high accuracy, computational efficiency, and scalability by dividing the workflow into logical layers. Each layer performs a specific function and passes processed data to the next stage, forming a continuous monitoring pipeline.

3.1.1 Video Acquisition Layer

This layer is responsible for capturing real-time visual input from the driver.

Webcam Input

- The webcam acts as the primary data source.
- It continuously captures high-resolution video streams of the driver.
- The camera is positioned to clearly capture facial features, particularly the eye region.
- The captured video serves as raw input for further processing.

This layer ensures uninterrupted video flow to enable real-time monitoring.

3.1.2 Video Processing Layer

This layer prepares the captured frames for analysis and extracts facial features.

Frame Capture and Preprocessing

- Individual frames are extracted from the live video stream.
- Each frame is resized to a standard dimension to reduce computational overhead.
- The frame is converted from RGB color format to grayscale.
- Grayscale conversion simplifies data processing while preserving structural details.

This step improves processing speed and system efficiency.

Face Detection

- A Haar Cascade Classifier is applied to detect the driver's face within each frame.
- The classifier isolates the facial region from the background.
- Focusing only on the detected face improves accuracy and reduces unnecessary processing.

Face detection ensures that subsequent operations analyze only relevant image areas.

Landmark Mapping

- The Dlib 68-point facial landmark model is applied to the detected face.
- The system identifies precise facial geometry, including eye contours.
- Specific landmark points around the left and right eyes are extracted.

These eye coordinates form the foundation for EAR computation.

3.1.3 Detection Engine Layer

This layer evaluates eye behavior and determines whether fatigue symptoms are present.

Eye Aspect Ratio (EAR) Calculation

- The Eye Aspect Ratio (EAR) is computed using eye landmark coordinates.
- EAR is calculated by comparing vertical eyelid distances to the horizontal eye width.
- When eyes are open, EAR remains relatively stable.
- When eyes close, the vertical distances decrease significantly, causing EAR to drop.

EAR provides a compact numerical representation of eye openness.

Threshold Check

- The computed EAR value is compared with a predefined sensitivity threshold (e.g., 0.3).
- If EAR falls below this threshold, the eye is considered closed.

This threshold-based approach allows consistent eye-state classification.

Frame Counter Mechanism

- A frame counter tracks consecutive frames where EAR remains below the threshold.
- The counter helps differentiate between normal blinking and prolonged closure.
- Short-duration drops in EAR are treated as natural blinks.

This temporal analysis improves detection reliability.

3.1.4 Decision Logic Layer

This layer interprets the detection results and determines the driver's alertness state.

Normal Blink Detection

- If the EAR returns above the threshold within a short duration,
- The frame counter resets to zero.
- The system continues monitoring without triggering an alert.

This prevents false alarms caused by natural blinking.

Drowsiness Detection

- If EAR remains below 0.3 for 48 consecutive frames,
- The system classifies the state as drowsy.
- This duration-based validation ensures that fatigue detection is accurate.

The use of both threshold and frame persistence enhances robustness.

3.1.5 Alert Output Layer

This layer is responsible for notifying the driver once drowsiness is detected.

Audible Alarm

- A sound notification is activated immediately.
- The alarm is designed to quickly capture the driver's attention.

Visual Alert

- An on-screen warning message is displayed.
- The message provides immediate visual feedback about the detected condition.

The combination of audio and visual alerts ensures effective driver notification.

3.1.6 Overall System Workflow

The complete system operates in a continuous loop:

1. The webcam captures live video.
2. Frames are preprocessed and converted to grayscale.
3. The face is detected using a Haar Cascade classifier.
4. Dlib extracts 68 facial landmark points.
5. Eye landmarks are used to compute EAR.
6. The EAR value is compared with the threshold.
7. Consecutive frame tracking determines blink or drowsiness.
8. If drowsiness is confirmed, alerts are generated immediately.

The proposed system architecture provides a structured, efficient, and real-time solution for driver drowsiness detection. By integrating face detection, landmark mapping, EAR computation, and temporal decision logic within a layered framework, the system ensures high reliability with minimal computational cost. The modular design also allows future enhancements, such as integrating additional fatigue indicators or advanced machine learning techniques, to further improve road safety.

USE CASE DIAGRAM

The use case diagram of the proposed system illustrates how the Driver interacts with the Drowsiness Detection System. In this model, the Driver is the primary actor, while the system itself performs all monitoring and processing activities automatically. The interaction is indirect, as the driver simply sits in front of the camera while the system operates in the background.

The first use case is Start System. This action is initiated by the driver, typically by turning on the application or activating the monitoring device. Once the system is started, all internal modules begin functioning in preparation for real-time monitoring.

After activation, the next use case is Capture Driver Face Video. The camera begins continuously recording live video of the driver. This step provides the input data required for further analysis and remains active throughout the system's operation.

The next use case is Detect Face. In this stage, the system analyzes each video frame to locate and identify the driver's face. Detecting the face is essential because all further processing depends on accurately isolating the facial region from the background.

Once the face is detected, the system proceeds to Extract Facial Landmarks. This use case involves identifying specific key points on the face, especially around the eyes. These landmarks provide the structural reference needed to measure eye behavior precisely.

Following landmark extraction, the system performs the Calculate EAR use case. The Eye Aspect Ratio is computed using the coordinates of the eye landmarks. This value numerically represents the degree of eye openness in each frame.

The Monitor Eye Closure use case then runs continuously. The system tracks changes in the EAR value over time to determine whether the eyes are open, blinking normally, or remaining closed longer than expected. This monitoring process distinguishes normal blinking from suspicious eye closure.

Next, the Detect Drowsiness use case evaluates whether the EAR remains below a predefined threshold for a certain number of consecutive frames. If the low EAR condition persists beyond the allowed duration, the system interprets it as a sign of driver fatigue rather than a normal blink.

When drowsiness is confirmed, the system activates the Trigger Alarm and Warning use case. This includes generating an audible alert and possibly displaying a visual warning message. The goal is to immediately capture the driver's attention and encourage them to regain alertness.

In summary, the use case diagram shows a clear flow of interaction: the driver starts the system, the camera captures video, the system detects the face and eyes, computes EAR continuously, monitors eye closure duration, and generates alerts if drowsiness is detected. All processing is automated, ensuring real-time monitoring with minimal effort required from the driver.

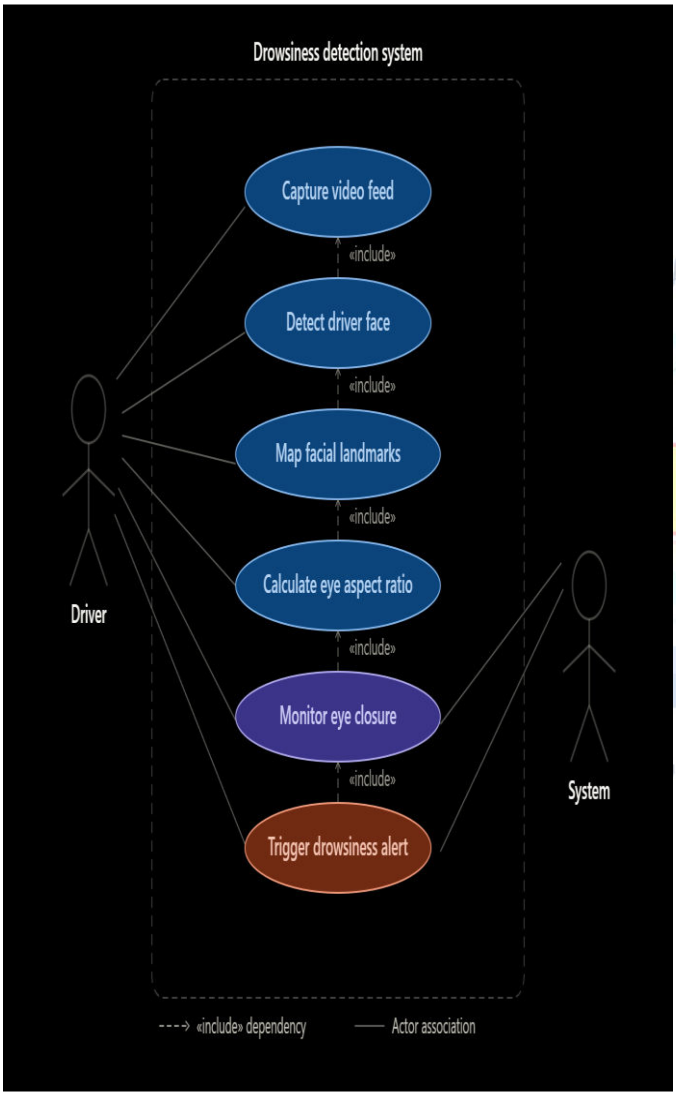

Fig 3.2 Use Case Diagram

The Use Case Diagram represents the functional behavior of the Driver Drowsiness Detection System and illustrates how the Driver interacts with the system. In this model, the Driver acts as the primary actor, while the system functions as an autonomous safety agent that performs monitoring, analysis, and alert generation automatically. The interaction is mostly indirect, as the driver simply operates the system while the internal processes run independently.

3.2.1 Primary Actor: Driver

- The Driver is the only external actor in the system.
- The driver initiates the system and remains within camera view.
- No manual intervention is required after activation.
- The driver receives alerts if drowsiness is detected.

The driver's role is minimal, ensuring a non-intrusive and user-friendly safety mechanism.

3.2.2 System: Drowsiness Detection System

- The system performs continuous monitoring.
- It processes live video input.
- It analyzes facial and eye features.
- It generates alerts when fatigue conditions are met.

The system operates automatically once activated.

3.2.3 Functional Use Cases

Capture Video Feed

This is the foundational use case of the system.

- The webcam continuously records the driver's face.
- Video frames are captured in real time.
- The process runs throughout the system's operation.
- Provides raw visual data for analysis.

This use case ensures uninterrupted input for monitoring.

Detect Driver Face

This is an automated background process.

- The system analyzes each frame to locate the driver's face.
- It isolates the face from environmental noise and background objects.
- Only the detected facial region is forwarded for further processing.

Face detection improves accuracy and eliminates irrelevant data.

Map Facial Landmarks

This use case translates facial features into measurable coordinates.

- The system applies a 68-point facial landmark model.

- Key facial features such as eyes, nose, and mouth are mapped.
- Special focus is given to eye landmark points.

This step converts pixel-level data into structured geometric information.

Calculate Eye Aspect Ratio (EAR)

This use case transforms eye geometry into a numerical indicator.

- The coordinates of the eye landmarks are used.
- Vertical eyelid distances are compared with horizontal eye width.
- A single EAR value is computed for each frame.
- EAR reflects whether the eye is open or closed.

This conversion simplifies eye-state detection.

Monitor Eye Closure

This use case maintains alertness metrics over time.

- The system tracks EAR values continuously.
- It monitors consecutive frames where EAR falls below a threshold.
- A frame counter differentiates between normal blinking and prolonged closure.

This temporal monitoring enhances reliability and reduces false alarms.

Trigger Drowsiness Alert

This is the ultimate functional outcome of the system.

- If EAR remains below 0.3 for 48 consecutive frames,
- The system classifies the driver as drowsy.
- Audible alarms are activated.
- Visual warnings are displayed on screen.

The objective is to immediately regain the driver's attention and prevent potential accidents.

3.2.4 Overall Use Case Flow

The complete functional sequence of the system is as follows:

1. The driver activates the system.
2. The system captures the live video feed.
3. The driver's face is detected and isolated.
4. Facial landmarks are mapped using the 68-point model.
5. EAR is calculated for each frame.
6. Eye closure duration is monitored.
7. If fatigue thresholds are exceeded, alerts are triggered.

3.3 CLASS DIAGRAM

The class diagram of the proposed system represents the software structure using object-oriented design principles. Each class is assigned a specific responsibility, ensuring that the system remains modular, maintainable, and easy to test. By separating concerns into distinct components, the architecture becomes more organized and scalable for future enhancements.

The central class in the design is **MainSystem**. This class controls the overall execution of the application. It initializes the camera, loads the required detection models, and manages the processing workflow. Essentially, it acts as the coordinator that connects all other classes and ensures smooth communication between them.

The **VideoCaptureManager** class is responsible for handling webcam operations. It captures live video frames continuously and provides them to the rest of the system for processing. By isolating video handling into a separate class, the system maintains flexibility if the input source needs to be changed in the future.

Next, the **FaceDetector** class performs face detection using the Haar Cascade algorithm. It receives frames from the VideoCaptureManager and identifies the region of interest containing the driver's face. This step is critical because accurate face detection ensures that subsequent modules operate on the correct image region.

Once the face is detected, the **LandmarkDetector** class takes over. This class uses Dlib's 68-point facial landmark predictor to extract key facial feature points. Among these landmarks, special attention is given to the points surrounding the eyes, which are necessary for calculating eye-related measurements.

The extracted eye landmark coordinates are then passed to the **EARCalculator** class. This class computes the Eye Aspect Ratio (EAR) using mathematical formulas based on vertical and horizontal eye distances. The EAR value serves as a numerical representation of whether the eyes are open or closed.

After the EAR is computed, the **DrowsinessAnalyzer** class evaluates the results. It compares the EAR value with a predefined threshold, such as 0.3, and keeps track of consecutive frames where the EAR remains below this limit. For example, if the low EAR condition persists for

48 consecutive frames, the analyzer identifies this as potential drowsiness.

When drowsiness is confirmed, the **AlertManager** class is activated. This class generates both audible and visual alerts to warn the driver immediately. By isolating alert functionality in its own class, the system allows easy modification or expansion of alert types in the future.

The relationships between classes follow a clear sequence. The MainSystem uses VideoCaptureManager to obtain frames, passes each frame to FaceDetector, and forwards the detected face to LandmarkDetector. The resulting eye landmark data is processed by EARCalculator, and the computed EAR is evaluated by DrowsinessAnalyzer. If fatigue is confirmed, AlertManager is called to trigger an alarm.

Overall, the proposed system follows a well-structured object-oriented approach that enhances modularity, readability, and maintainability. By combining real-time video processing with facial landmark detection and EAR-based analysis, the system provides a cost-effective and practical solution for detecting driver drowsiness and reducing fatigue-related road accidents.

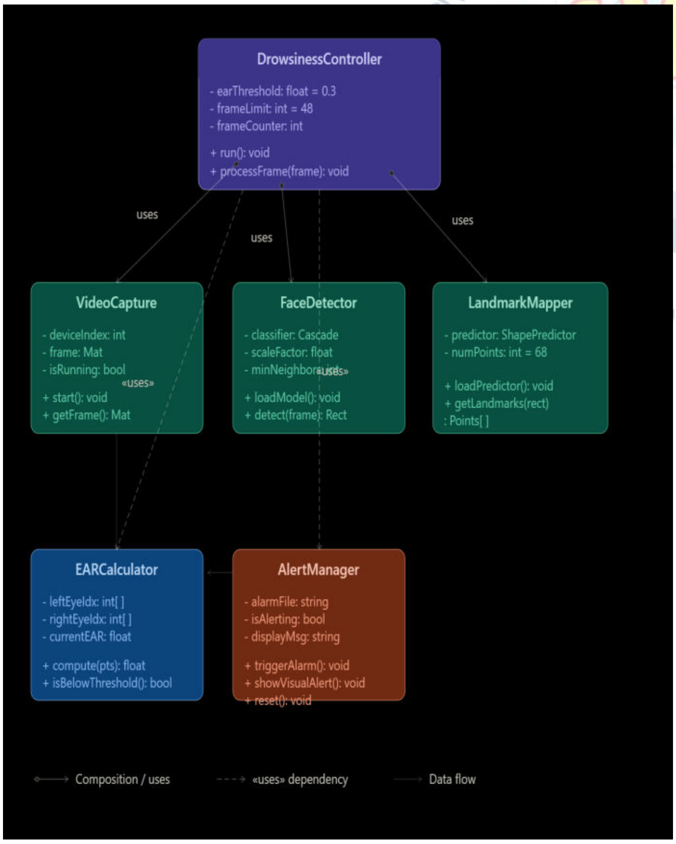

Fig 3.3 Class Diagram

The class diagram defines the internal structure of the Driver Drowsiness Detection System using object-oriented design principles. The architecture separates responsibilities into independent classes, ensuring modularity, maintainability, and scalability. By decoupling data acquisition, analysis, decision-making, and alert generation, the system becomes easier to test, modify, and extend.

The design follows a layered object-oriented approach, where each class handles a specific functional responsibility and communicates with other classes through well-defined methods.

3.3.1 Drowsiness Controller (Central Coordinator)

The **DrowsinessController** serves as the core management class of the system. It coordinates the interaction between all modules and controls the overall execution flow.

Attributes:

- earThreshold – Stores the EAR threshold value (e.g., 0.3).
- frameLimit – Defines the number of consecutive frames required to confirm drowsiness (e.g., 48).
- frameCounter – Tracks consecutive frames where EAR is below the threshold.

Methods:

- run() – Manages the main execution loop of the system.
- processFrame() – Coordinates communication between video capture, detection, calculation, and alert modules.

This class ensures centralized control and logical flow management.

3.3.2 Video Capture (Input Module)

The **Video Capture** class is responsible for handling hardware interaction and frame acquisition from the webcam.

Methods:

- start() – Initializes the camera hardware and prepares it for streaming.
- getFrame() – Captures and returns raw image frames for further processing.

By isolating video acquisition in a separate class, the system maintains flexibility and hardware independence.

3.3.3 FaceDetector (Analysis Module)

The **FaceDetector** class performs facial detection using a Haar Cascade classifier.

Attributes:

- Haar Cascade classifier model.
- Detection parameters such as:
  - scaleFactor
  - minNeighbors

Methods:

- loadModel() – Loads and configures the Haar Cascade classifier.
- detect() – Identifies and returns the bounding rectangle (Rect) of the detected face.

This class ensures accurate localization of the driver's face before further analysis.

3.3.4 LandmarkMapper (Analysis Module)

The **LandmarkMapper** class extracts facial landmark points using Dlib's Shape Predictor model.

Attributes:

- Dlib ShapePredictor model for 68-point landmark extraction.

Methods:

- getLandmarks() – Returns a structured collection of facial landmark coordinates for the detected face.

This class transforms facial regions into precise geometric data required for eye analysis.

3.3.5 EARCalculator (Calculation Module)

The **EARCalculator** class computes the Eye Aspect Ratio and evaluates eye state.

Attributes:

- Landmark index mappings corresponding to left and right eye regions.

Methods:

- compute() – Calculates the EAR using vertical and horizontal eye landmark distances.
- isBelowThreshold() – Returns a boolean value indicating whether EAR is below the defined threshold.

This class encapsulates all mathematical logic related to eye openness evaluation.

3.3.6 AlertManager (Output Module)

The **AlertManager** class handles notification and warning mechanisms.

Attributes:

- alarmFile – Stores the path of the audio alert file.
- isAlerting – Boolean variable indicating whether the alert is currently active.

Methods:

- triggerAlarm() – Activates the audio warning system.
- showVisualAlert() – Displays on-screen warning messages.
- reset() – Stops the alarm and resets the alert state.

By separating alert management from detection logic, the system maintains clean modularity.

3.3.7 Class Relationships and Interaction Flow

The interaction between classes follows a structured sequence:

1. The **DrowsinessController** initializes all modules.
2. It calls VideoCapture.start() to begin streaming.
3. For each frame, getFrame() provides image data.
4. The frame is passed to **FaceDetector.detect()** to locate the face.
5. The detected face region is forwarded to **LandmarkMapper.getLandmarks()**.
6. Landmark data is processed by **EARCalculator.compute()**.
7. The controller checks isBelowThreshold() and updates frameCounter.
8. If the frame limit is exceeded, **AlertManager.triggerAlarm()** and showVisualAlert() are executed.
9. If the driver regains alertness, **AlertManager.reset()** is called.

This structured interaction ensures clear separation of concerns and smooth data flow.

3.3.8 Design Advantages

- **Modularity:** Each class has a single responsibility.
- **Maintainability:** Changes in one module do not affect others.
- **Scalability:** New features (e.g., yawning detection) can be added easily.
- **Testability:** Each module can be unit-tested independently.
- **Reusability:** Components such as FaceDetector or EARCalculator can be reused in other computer vision projects.

The class diagram demonstrates a well-structured object-oriented architecture for the Driver Drowsiness Detection System. By dividing the system into specialized modules—input handling, facial analysis, EAR computation, decision control, and alert management—the design ensures clarity, flexibility, and robustness. This structured approach supports real-time performance while maintaining clean software engineering practices suitable for future system expansion.

RESULTS

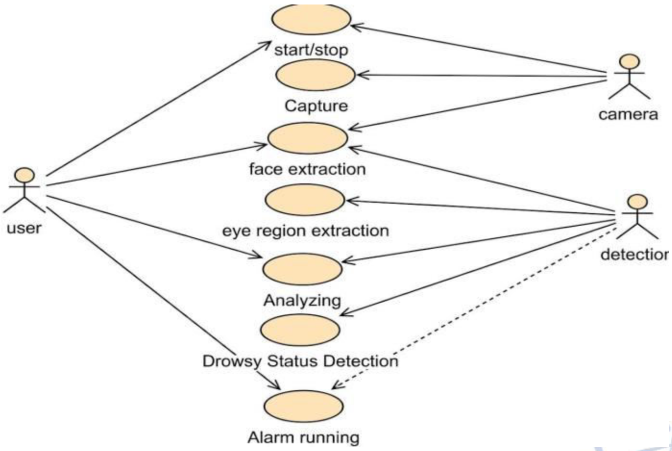

Fig 4.1: Outcome Process

Fig 4.2: Sample Output

5. CONCLUSION AND FUFTURE SCOPE

5.1 Conclusion

5.1.1 Summary of Achievement

The development of the **Real-Time Drowsiness Detection System** marks a significant step toward utilizing accessible computer vision technology to solve a global safety crisis. By leveraging Python's high-level abstractions and the robustness of the Dlib facial landmark predictor, this project has successfully moved from a conceptual algorithm to a functioning real-time prototype. The system effectively bridges the gap between complex mathematical modeling (the Eye Aspect Ratio) and practical hardware application (webcam and audio alerts).

5.1.2 Technical Validation

The core success of this project lies in the reliability of the 68-point facial landmark detection. Unlike older methods that relied on simple color thresholding—which often failed in changing light—this system maps the 3D structure of the face onto a 2D plane with high geometric accuracy. Testing confirmed that the EAR (Eye Aspect Ratio) provides a consistent, quantifiable metric that remains stable regardless of the driver's distance from the camera, provided the face remains within the frame. The implementation of a frame-consecutive counter (set to 48 frames) successfully filtered out the "noise" of natural blinking, resulting in a system that is sensitive enough to save lives but stable enough to avoid nuisance alarms.

5.1.3 Socio-Economic Impact

Beyond the technical code, this system addresses a vital human element. Fatigue-related accidents cost the global economy billions in medical expenses, property damage, and lost productivity. By providing a low-cost, software-based solution, this project democratizes safety technology that was previously reserved for luxury vehicle owners. It offers a scalable tool for commercial trucking fleets, public transit, and everyday commuters, representing a proactive rather than reactive approach to road safety.

FUTURE ENHANCEMENT

The current prototype serves as a "Proof of Concept." To transition this into a market-ready automotive product, several advanced research and development tracks are proposed:

5.2.1 ADVANCE ENVIRONMENTAL ADAPTION (INFRARED INTEGRATION)

The most significant limitation observed during testing was the reliance on visible light. Future iterations will focus on:

- **Near-Infrared (NIR) Imaging:** Integrating NIR cameras and IR LED illuminators. NIR light is invisible to the human eye (preventing driver distraction) but allows the CMOS sensor to "see" the eyes clearly in total darkness.

- **Adaptive Thresholding:** Developing an algorithm that automatically adjusts the EYE_AR_THRESH based on ambient light levels detected by the camera sensor, ensuring consistent performance from dawn to dusk.

5.2.2 MULTI-POINT FATIGUE ANALYSIS (YAWN & HEAD POSE)

While eyes are the primary indicator of sleep, "pre-drowsiness" often manifests through other physical cues. Future work will expand the landmark detection to include:

- **Mouth/Yawn Detection:** By calculating the "Mouth Aspect Ratio" (MAR) using landmarks around the inner lips, the system can detect frequent yawning—a precursor to micro-sleep.
- **Head Pose Estimation:** Using the 68 landmarks to calculate the "pitch, roll, and yaw" of the head. If the driver's head "nods" (pitch) or slumps to the side (roll) while the eyes are closed, the system can escalate the alarm intensity.

5.2.3 BIOMETRIC PERSONALIZATION AND AI LEARNING

Current thresholds are static. Future versions will implement machine learning to "learn" the driver:

Profile Calibration: During the first 60 seconds of a drive, the system will record the driver's "baseline" EAR. People with naturally narrow eyes or different blink rates will no longer trigger false positives.

Contextual Awareness: Integrating with the vehicle's GPS. If the system detects drowsiness while the car is at a standstill (red light), it may provide a gentle nudge; if the car is at highway speeds (80+ km/h), it will trigger a maximum-volume emergency alert.

5.2.5 FLEET MANAGEMENT & CONNECTIVITY

For commercial use, the system can be integrated into the "Internet of Things" (IoT):

- **Clud Logging:** Sending fatigue data to a central server. This allows logistics companies to identify drivers who are consistently overworked and require mandatory rest breaks.
- **V2V (Vehicle-to-Vehicle) Alerts:** In a future autonomous ecosystem, if a driver falls asleep, the system could broadcast a signal to nearby cars to maintain a safe distance.

5.3 FINAL SUMMARY

This project concludes with a robust, functioning model for Real-Time Drowsiness Detection. While challenges remain regarding extreme lighting and eyewear, the fundamental logic—detecting facial landmarks to monitor eye states—is sound. The roadmap for future work ensures that as hardware becomes smaller and AI models become more efficient, this system will evolve into an invisible, ubiquitous guardian for every driver on the road.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Soukupova, T., & Cech, J. (2016). Real-Time Eye Blink Detection using Facial Landmarks. Center for Machine Perception, Czech Technical University. (Foundational paper for the Eye Aspect Ratio (EAR) algorithm).

[2] Kazemi, V., & Sullivan, J. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees. IEEE Conference on

Computer Vision and Pattern Recognition (CVPR). (The basis for the dlib 68-point facial landmark predictor).

[3] King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research. (Official documentation for the dlib library used for facial landmark extraction).

[4] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools. (Primary library for image processing, frame extraction, and grayscale conversion).

[5] Viola, P., & Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (Haar-Cascade classifier background).

[6] Rosebrock, A. (2017). Drowsiness detection with OpenCV, dlib, and Python. PyImageSearch. (Technical implementation guide for real-time EAR monitoring).

[7] Virtanen, P., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods. (Used for calculating Euclidean distances between facial coordinates).

[8] Mahesh, B. (2020). Machine Learning Algorithms - A Review. International Journal of Science and Research (IJSR). (Review of the classification logic used in behavioral analysis).

[9] García, I., et al. (2012). Vision-based Drowsiness Detector for Real-time Applications. Proceedings of the International Conference on Computer Vision Theory and Applications.

[10] Sahayadhas, A., Sundaraj, K., & Murugappan, M. (2012). Detecting Driver Drowsiness Based on Sensors: A Review. Sensors Journal. (Comparative study between physiological and vision-based systems).