**A Project Report**

**On**

# A VISION-BASED DRIVER DROWSINESS DETECTION FRAMEWORK FOR ROAD SAFETY ENHANCEMENT

*Submitted to*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTHAPUR, ANANTHAPURAMU**

*In Partial Fulfillment of the Requirements for the Award of the Degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND TECHNOLOGY**

**Submitted By**

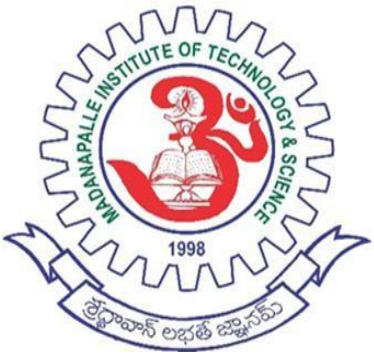

| | | |
|---|---|---|
| **P. Kavya** | **-** | **(21691A2869)** |
| **U. Lavanya** | **-** | **(21695A2873)** |
| **N. Rajesh** | **-** | **(21691A28D2)** |
| **M. Prasanth** | **-** | **(22695A2812)** |

**Under the Guidance of**

**Mrs. P. Jayaselvi**

**Assistant Professor**

**Department of Computer Science and Technology**



## MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

## (UGC – AUTONOMOUS)

**(Affiliated to JNTUA, Ananthapuramu)**

**Accredited by NBA, Approved by AICTE, New Delhi**

**AN ISO 21001:2018 Certified Institution**

**P. B. No: 14, Angallu, Madanapalle, Annamayya – 517325**


**2021-2025**

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC-AUTONOMOUS INSTITUTION)
Approved by AICTE, New Delhi and Affiliated to JNTUA, Anantapuramu
www.mits.ac.in www.mits.edu

## DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

## BONAFIDE CERTIFICATE

This is to certify that the Project work & internship (20CST703) entitled "**A VISION-BASED DRIVER DROWSINESS DETECTION FRAMEWORK FOR ROAD SAFETY ENHANCEMENT"** is a bonafide work carried out by

| | | |
|---|---|---|
| **P. KAVYA** | **-** | **(21691A2869)** |
| **U.LAVANYA** | **-** | **(21695A2873)** |
| **N.RAJESH** | **-** | **(21691A28D2)** |
| **M. PRASANTH** | **-** | **(22695A2812)** |

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the stream of **Computer Science and Technology** in **Madanapalle Institute of Technology & Science, Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Ananthapur, Ananthapuramu** during the academic year 2024-2025.

**Guide**
**Mrs. P. Jayaselvi**
**Assistant Professor**
**Department of CST**

**Head of the Department**
**Dr. K. Dinesh**
**Associate Professor and Head**
**Department of CST**

Submitted for the university examinations held on

**Internal Examiner**
**Date:**

**External Examiner**
**Date:**

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the **Management** of **Madanapalle Institute of Technology & Science** for providing exceptional infrastructure and laboratory facilities that were instrumental in enabling us to successfully complete our project. The excellent resources and support provided by the institution played a vital role in helping us achieve our academic goals.

We express our sincere gratitude to **Dr. C. Yuvaraj, M.E., Ph.D., Principal,** for his invaluable guidance and unwavering support that enabled us to successfully complete our project work & internship (20CST703) at Madanapalle Institute of Technology & Science, Madanapalle.

We are deeply grateful to **Dr. K. Dinesh, M.Tech., Ph.D., Associate Professor** and **Head of the Department of CST,** for his steadfast support and tireless efforts that contributed to the successful completion of our project work & internship (20CST703) and his guidance helped us navigate challenges and achieve our goals.

We are immensely grateful to **Mr. Abdul Jaleel D, M.E., (Ph.D.), Project Coordinator**, for his invaluable guidance, unwavering encouragement, and support that enabled us to successfully complete this project.

We would like to express our profound gratitude to our guide, **Mrs. P. Jayaselvi, M.Tech, Assistant Professor, Department of CST**, for his invaluable guidance and unwavering encouragement that played a pivotal role in the successful completion of our project. His expertise, insights, and mentorship were crucial in navigating the intricacies of the project work & internship (20CST703) and achieving our objectives.

We express our sincere thanks to **Mr. V. Naveen, M.E., (Ph.D.), Internship Coordinator,** for his tremendous support for the successful completion of the Project.

We would like to express our gratitude to the other Faculty members, of the **CST Department**, and parents for their invaluable help and unwavering support during our project work & internship (20CST703).

# INTERNSHIP CERTIFICATE

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

**Polur Kavya**

has successfully completed the **Machine Learning Internship**
at Slash Mark IT Solutions (OPC) Pvt Ltd (An ISO 9001:2015 certified organization dedicated to excellence in IT solutions)
during the **December 27, 2024 to April 27, 2025**

**Shri P Abhishek**
HR, SLASH MARK

**Shri K Mukesh Raj**
CEO, SLASH MARK

**Intern ID :** SMI75038

# INTERNSHIP CERTIFICATE

# INTERNSHIP CERTIFICATE

## INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

**Neellibavi Rajesh**

has successfully completed the **Internet Of Things Internship**
at Slash Mark IT Solutions (OPC) Pvt Ltd (An ISO 9001:2015 certified organization dedicated to excellence in IT solutions)
during the **December 27, 2024 to April 27, 2025**

**Shri P Abhishek**
HR, SLASH MARK

**Shri K Mukesh Raj**
CEO, SLASH MARK

**Intern ID :** SMI75133

# INTERNSHIP CERTIFICATE

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

**M Prasanth**

has successfully completed the **Machine Learning Internship** at Slash Mark IT Solutions (OPC) Pvt Ltd (An ISO 9001:2015 certified organization dedicated to excellence in IT solutions) during the **December 27, 2024 to April 27, 2025**

**Shri P Abhishek**
HR, SLASH MARK

**Shri K Mukesh Raj**
CEO, SLASH MARK

**Intern ID :** SMI75266

# MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC-AUTONOMOUS INSTITUTION)

Approved by AICTE, New Delhi and Affiliated to JNTUA, Anantapuramu

www.mits.ac.in www.mits.edu

## DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

***Plagiarism Verification Certificate***

This is to certify that the B. Tech Project work & internship (20CST703) work report titled, **"A VISION-BASED DRIVER DROWSINESS DETECTION FRAMEWORK FOR ROAD SAFETY ENHANCEMENT"** submitted by **P. KAVYA (21691A2869),U. LAVANYA (21691A2873), N. RAJESH (21691A28D2), M . PRASANTH (22695A2812)** has been evaluated using Anti- Plagiarism Software, Turnitin and based on the analysis report generated by the software, the report's similarity index is found to be 11%.

jayaselvi kavya

4_2 report(2).docx

Madanapalle Institute of Technology and Science

Document Details

Submission ID
trn:oid:::3618:92702274

Submission Date
Apr 24, 2025, 4:19 PM GMT+5:30

Download Date
Apr 24, 2025, 4:21 PM GMT+5:30

File Name
4_2 report(2).docx

File Size
2.7 MB

63 Pages
6,510 Words
41,117 Characters

turnitin Page 1 of 68 - Cover Page Submission ID trn:oid:::3618:92702274

turnitin Page 2 of 68 - Integrity Overview Submission ID trn:oid:::3618:92702274

11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

85 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

15 Missing Citation 2%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

4% Internet sources
4% Publications
11% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

**HOD-CST**

# DECLARATION

We hereby declare that the results embodied in this Project Work & internship (20CST703) **"A VISION-BASED DRIVER DROWSINESS DETECTION FRAMEWORK FOR ROAD SAFETY ENHANCEMENT"** by us under the guidance of **Mrs. P. Jayaselvi, M.Tech., Assistant Professor, Department of CST** in partial fulfillment of the award of **Bachelor of Technology** in **Computer Science and Technology** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and we have not submitted the same to any other University/institute for award of any other degree.

**Date:**

**Place:**

## PROJECT ASSOCIATES

**P KAVYA - (21691A2869)**

**U LAVANYA - (21691A2873)**

**N RAJESH - (21691A28D2)**

**M PRASANTH - (22695A2812)**

I certify that the above statement made by the students is correct to the best of my knowledge.

**Date:**

**Guide**
**Mrs. P. Jayaselvi**
**Assistant Professor**
**Department of CST**

# ABSTRACT

Driver drowsiness detection systems are vital for mitigating traffic accidents stemming from driver fatigue. To enhances traditional methods by incorporating not only real-time analysis of facial cues like eye closure and head movements, but also integrates automatic call detection and distraction alertness features. By leveraging computer vision and machine learning algorithms, the system identifies signs of drowsiness while simultaneously monitoring for phone usage and other forms of driver distraction. The inclusion of call detection and distraction alertness to provide a more comprehensive assessment of the driver's cognitive state, recognizing that distractions can compound the effects of drowsiness. This multifaceted approach enables the system to differentiate between simple fatigue and more complex scenarios where inattention plays a significant role, leading to more accurate and proactive interventions. It provide a robust and reliable solution that minimizes false positives while maximizing the detection of genuine drowsiness and distraction events. This is achieved through the development of a sophisticated algorithm that analyzes multiple data streams, including facial features, phone usage, potentially other contextual cues. Upon detecting drowsiness or distraction, the system can trigger a series of alerts, ranging from auditory warnings to haptic feedback, potentially initiate automatic call silencing or other interventions. The goal is to create a proactive safety system that adapts to the driver's behavior and environment, ultimately contributing to a significant reduction in accidents caused by driver inattention.

***Keywords:*** *Drowsiness detection, Automatic call detection, Head titling, Facial muscle activities*.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **ADAS** | Advanced Driver-Assistance Systems |
| **FPS** | Frames Per Second |
| **GUI** | Graphical User Interface |
| **EAR** | Eye Aspect ratio |
| **DDA** | Driver Drowsiness and Distraction Alert |
| **ACD** | Automatic Call Detection |
| **CNN** | Convolutional Neural Network |
| **YOLOV5** | You Only Look Once Version5 |
| **DMS** | Driver Monitoring System |
| **MAR** | Mouth Aspect ratio |
| **CPU** | Central Processing Unit |
| **GPS** | Global Positioning System |
| **PERCLOS** | percentage of eye closure over time |
| **EEG** | Electroencephalography |

# CHAPTER 1

# INTRODUCTION

### 1.1 Motivation

Driver fatigue plays a major role in causing road accidents across the globe, posing a serious threat to traffic safety. Exhaustion slows reflexes, hinders judgment and reduces overall driving ability, Causing a risk for the driver as well as others on the road. To address this challenge, driver drowsiness detection systems have been developed, utilizing real-time monitoring of physiological and behavioural indicators. However, traditional detection methods often focus solely on drowsiness, overlooking other crucial factors that compromise road safety. To enhance the effectiveness of such systems, the integration of additional features such as automatic call detection and distraction alertness. Automatic call detection ensures that drivers are not engaging in phone conversations that could divert their attention, while distraction alertness helps identify and mitigate risks caused by external or internal distractions, For instance, diverting attention from the road or engaging in non-driving activities. By incorporating these advanced functionalities, the system is to create a more comprehensive safety mechanism, reducing accident risks and promoting responsible driving behaviour.

### 1.2 Problem Definition

Road safety remains a major global concern, with driver fatigue and distractions being significant contributors to accidents. Traditional driver drowsiness detection systems primarily focus on identifying signs of fatigue but often fail to address other critical factors that impair driving performance, such as mobile phone usage and distractions from the surroundings. These limitations reduce the overall effectiveness of such systems in preventing accident. To solve these difficulties, focuses on designing a high-end driver monitoring system capable of detecting drowsiness while also integrating features such as automatic call detection and distraction alerts. By leveraging sophisticated algorithms by combining on-the-spot surveillance, the solution will assess the driver's performance condition and provide instant warnings when fatigue or distraction is identified. Combining multiple safety functions into a unified solution, this initiative to enhance road safety, minimize accidents, and encourage responsible driving habits. Road safety remains a major global concern, with driver fatigue and distractions being significant contributors to accidents. This approach not only improves the accuracy of driver state monitoring but also ensures real-time intervention to mitigate potential risks.

### 1.3 Objectives of the Project

To design an advanced driver surveillance solution capable of recognizing sleepiness and inattentive actions to promote road safety. By incorporating multiple real-time monitoring capabilities, the system will provide a holistic approach to preventing accidents.

- **Drowsiness Detection:** Implement an intelligent A system designed to precisely detect indicators of driver exhaustion based on behavioural and physiological indicators.
- **Automatic Call Detection:** Develop a feature that detects incoming or ongoing phone calls and alerts the driver to minimize distractions caused by mobile phone usage.
- **Distraction Alertness:** Incorporate real-time monitoring to identify instances in cases where the driver is distracted from the road, such as looking away or engaging in non-driving activities.
- **Immediate Alert Mechanism:** Ensure that the system provides instant audio alerts to warn when the driver feels drowsy or distraction is identified.
- **Enhanced Road Safety:** Minimize accident chances through improving driver awareness and promoting responsible driving behaviour through proactive monitoring.

### 1.4 Scope of the project

It involves the invention and execution of a sophisticated driver monitoring system developed at ensuring safer roads by recognizing fatigue and reducing distractions. It will leverage real-time data processing and smart technology that alert mechanisms to ensure to keep drivers engaged and responsive during their journey. It covers multiple aspects of driver monitoring, including the detection of fatigue through behavioural and physiological indicators, automatic call detection to prevent mobile phone distractions, distraction alertness to identify any deviation from focused driving. The system will integrate employs a fusion of multiple sensors visual input, utilizing intelligent computational methods to evaluate the driver's actions and patterns trigger appropriate alerts when necessary. It focus on wide range of vehicles, including personal cars, commercial transport, fleet management systems. It provides a user-friendly and a streamlined solution designed for smooth integration into current vehicle systems infrastructure.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 Overview

Fatigued driving identification system has become an extensively researched area in road safety due to its critical role in preventing accidents. Various studies have explored different approaches to detecting driver fatigue, including physiological, behavioural, vehicular-based methods. Traditional techniques rely on eye-tracking, blink detection, yawning analysis, and steering pattern monitoring. The integration of AI, ML and deep learning technologies allows advanced systems to detect drowsiness more accurately and act in real time.

## 2.2 Related work

**"A Systematic Review on Driver Drowsiness Detection Using Eye Activity Measures."**

**Date of Publication: August 14, 2024**

This research presents a Comprehensive analysis systems developed to track driver fatigue by examining eye-related signals like the rate of eyelid movements and the length of eye shut periods, and the PERCLOS metric. The system highlights that monitoring eye activity is an effective non-intrusive method for identifying drowsiness in real time. It also emphasizes the importance of selecting appropriate thresholds and sensor technologies to enhance detection accuracy. Despite the advancements, challenges remain in maintaining reliable functionality across environments and driving scenarios

**"A Hybrid Approach for Driver Drowsiness and Distraction Detection."**

**Date of Publication: April 20, 2020**

A hybrid model combining behavioural and physiological features for detecting both drowsiness and distractions. The system incorporates face-tracking technology and wearable heart rate sensors to improve accuracy. While the method enhances reliability, it requires synchronization of multiple data sources, which increases system complexity.

**"Real-Time Driver Fatigue Monitoring System Using EEG and Physiological Signals."**

**Date of Publication: October 10, 2020**

The system presents A system for detecting fatigue using electroencephalogram (EEG) signals along with physiological factors like heart rate and skin conductivity. By utilizing deep learning models, the system achieves high accuracy in identifying drowsiness patterns.

Although this method provides reliable detection, the dependency on EEG sensors may limit its practicality due to the need for direct skin contact and calibration.

**"Detection of Driver Drowsiness Using Vehicle-Based Parameters."**

**Date of Publication: February 14, 2021**

A research system that examines vehicle-based indicators, such as steering patterns, lane deviations, acceleration changes, to detect drowsiness. Machine learning algorithms analyse driving behaviour to predict fatigue levels. While non-intrusive, this method may not capture early-stage drowsiness as effectively as physiological monitoring techniques.

**"Multi-Feature Driver Drowsiness Detection System Using Wearable Sensors."**
**Date of Publication: March 5, 2021**

Proposing a multi-feature approach, this research integrates wearable Devices like motion sensors, including accelerometers and gyroscopes to monitor driver head movements and posture shifts. Artificial intelligence methods, Techniques such as Support Vector Machines (SVM) and Random Forest techniques are employed to classify drowsiness levels. While the system demonstrates robust performance, the reliance on wearable devices could lead to user discomfort and maintenance concerns.

**“AI-Based Driver Distraction and Drowsiness Detection System."**
**Date of Publication: August 12, 2022**

The system investigates an AI-powered detection system that simultaneously identifies driver drowsiness and distractions, such as mobile phone usage and looking away from the road. Deep learning models analyse real-time video footage to determine inattentiveness. The method improves road safety but faces challenges related to data privacy and processing speed.

**"Enhancing Road Safety with AI-Powered Drowsiness Detection Systems."**

**Date of Publication: September 25, 2023**

An in-depth analysis of how artificial intelligence contributes to enhancing driver safety. The system explores advanced AI models and their effectiveness in identifying drowsiness and distractions instantly. It also examines the challenges associated with their implementation bias, environmental variability, and hardware requirements are discussed, emphasizing the need for further advancements in this field. The integration of real-time data processing and adaptive learning capabilities allows AI systems to evolve with diverse driving conditions.

### 2.3 Detection of driver fatigue

Detecting driver fatigue involves monitoring physiological and behavioural indicators to assess drowsiness levels and prevent accidents. Modern systems use visual processing techniques to monitor eye motion, blinking patterns, and head posture for recognizing fatigue symptoms. Additionally, machine learning models process real-time data from facial expressions and driving patterns to enhance accuracy. Wearable sensors and vehicle-based measures, like steering behaviour analysis and deviation detection, further contribute to identifying drowsy driving. These innovations support the prompt delivery of alerts, contributing to fewer fatigue-related incidents and safer driving conditions.

### 2.4 Recent Studies on Detection of driver fatigue

Williams et al. (2020) Studied the integration of IoT-enabled sensing technologies for continuous driver observation emphasizing their effectiveness in detecting drowsiness through continuous data collection. Their study highlighted how IoT-based solutions can improve real-time decision-making and enhance driver safety by providing instant alerts.

Davis and Clark (2019) investigated the integration of gyroscopes and accelerometers in wearable devices for tracking driver head movements and posture shifts. Their research demonstrated the accuracy of these sensors in identifying fatigue-related behaviours, reinforcing their potential for early drowsiness detection.

Zhang et al. (2021) examined the integration of Raspberry Pi with cloud platforms for real-time driver fatigue monitoring. Their findings showcased how edge computing and cloud-based analytics enhance data transmission, storage, and visualization, leading to improved decision-making in driver monitoring applications.

Lee and Park (2022) conducted a study on the enhanced precision of infrared-based eye-tracking systems in detecting driver fatigue. Their research emphasized the importance of advanced sensor technology in identifying subtle changes in blink rates and gaze direction, improving the precision and effectiveness of identifying driver fatigue models.

Gomez et al. (2021) reviewed the role of AI-driven analytics platforms, such as TensorFlow and OpenCV, in processing facial recognition data to assess driver alertness. Their findings underscored the significance of deep learning in automating drowsiness detection and reducing false positives in real-time monitoring systems.

Rodriguez et al. (2020) explored the combination of vehicle-based parameters, including steering patterns, lane deviation, and braking behaviour, using utilizing machine learning methods to detect indicators of fatigue driver. The capability of merging behavioural and vehicular data to improve detection accuracy and reduce road accidents.

### 2.5 Extract from Literature

Despite notable advancements in driver drowsiness detection systems, several challenges remain in ensuring their widespread adoption and effectiveness. Issues such as sensor accuracy limitations, data privacy concerns, and integration complexities present significant obstacles to real-time driver monitoring. Overcoming these challenges requires interdisciplinary collaboration, technological innovation, and technological progress aimed at making fatigue detection faster and more reliable.

#### IoT Applications in Driver Fatigue Monitoring

Several studies highlight the role of IoT-enabled sensors in Instant detection of driver fatigue. Research by Williams et al. (2020) emphasizes the integration of IoT-based systems for monitoring driver alertness, enabling real-time interventions to prevent accidents.

#### Motion Tracking Sensors in Driver Behaviour Analysis

Motion tracking sensors, including gyroscopes and accelerometers, are widely utilized in detecting fatigue-related behaviours. A study by Davis and Clark (2019) explores the application of gyroscope-based sensors in tracking head movements and posture shifts, providing valuable insights into driver fatigue levels.

#### Integration of Raspberry Pi and IoT Platforms

The use of IoT development boards in fatigue detection has been extensively researched. Zhang et al. (2021) discuss seamless incorporation of Raspberry Pi with cloud platforms of Current data transmission, and examination showcasing its effectiveness driver alertness monitoring systems.

#### Advancements in Infrared and Eye-Tracking Sensors

Technological advancements in infrared-based eye-tracking have significantly improved drowsiness detection accuracy. Research by Lee and Park (2022) highlights the

effectiveness of infrared cameras in monitoring blink rates and gaze direction, ensuring precise fatigue assessment.

**Cloud-Based Analytics for Driver Monitoring**

Cloud-based analytics platforms play a vital role in processing real-time sensor data. A review by Gomez et al. (2021) showcases the capabilities of cloud-driven solutions like TensorFlow and OpenCV in analysing expressive facial patterns and eye motion providing intelligent insights to identify signs of fatigue.

**Integration of AI and Vehicular Information utilized in detecting driver fatigue**

The combination of artificial intelligence and vehicle-based parameters has led to enhanced driver monitoring techniques. Research by Rodriguez et al. (2020) explores the integration of AI algorithms with steering behaviour, lane deviation, and braking patterns to detect fatigue and inattentiveness.

**Obstacles and Future Prospects**

Despite notable advancements in detecting driver fatigue, several challenges persist, such as sensor reliability, data privacy, and system integration remain. It exploring edge computing for real-time processing, improving sensor accuracy, and advancing AI-driven analytics for enhanced driver safety monitoring.

**2.6 Summary**

The review presents A thorough analysis current research to detecting drivers' fatigue, emphasizing major technological developments, existing challenges, and future possibilities. Recent findings showcase the efficiency of IoT-based sensors, AI-powered analytics, and machine learning techniques in assessing driver alertness and mitigating fatigue-related incidents. The integration of motion tracking sensors, eye-tracking technologies, and cloud-based analytics has significantly improved real-time detection and response mechanisms. However, challenges such as sensor calibration, data privacy concerns, and system integration complexities remain. Future research to address these challenges by exploring edge computing for instantaneous data processing, refining AI models for enhanced accuracy, and developing more robust sensor technologies. It holds the capability to greatly improve road safety and prevent accidents in the coming years. Its collaboration between automotive manufacturers, AI researchers, and regulatory bodies will be crucial in setting standards for reliable deployment.

# CHAPTER 3

# PROBLEM ANALYSIS

### 3.1 Existing System

- The current fatigue monitoring solution for driver primarily relies on facial recognition and motion tracking techniques. The system utilizes a camera to observe facial cues like eye activity, blinking behaviour, head movement to assess driver alertness. The collected data is processed by an onboard microcontroller, which analyses detects driver fatigue and initiates an alert response upon identifying signs of fatigue.

- The system is designed to be compact and can be integrated into vehicles without requiring additional wearable devices. The microcontroller is programmed to process real-time data and provide immediate alerts through visual or audio notifications. This helps in continuously monitoring the physical and mental state of the driver while minimizing the likelihood of accidents due to fatigue.

**Disadvantages of the Existing System**

- **Dependence on Facial Features**: Many systems rely on facial recognition to detect drowsiness, making them less effective in low-light conditions.
- **System Maintenance and Updates**: Continuous software updates and calibration are required to maintain accuracy and ensure compatibility with different users and environments.
- **Privacy and Security Concerns**: As facial recognition and behavioural data are collected, there are concerns regarding data privacy and potential misuse. Proper encryption and security measures are necessary to prevent unauthorized access.

### 3.2 Proposed System

- The designed system is intended to improve driver safety by detecting drowsiness, phone usage, and distractions instantly through the application of deep learning algorithms. By leveraging YOLOv5, the system efficiently analyses facial landmarks, eye behaviour, head movements, and phone usage patterns to provide immediate alerts and prevent potential accidents Advanced analytics provide real-time insights, allowing instant alerts and preventive measures to be implemented. With scalability for future enhancements. It is built with the purpose of boosting road safety by preventing accidents resulting from driver exhaustion. It enables seamless integration with existing vehicle systems, enhancing its practicality for widespread adoption.

- By strategically placing gyroscope and accelerometer sensors to monitor head tilts and posture shifts, the system ensures accurate detection of drowsiness-related movements. The cloud-based platform stores and analyses collected data, generating detailed reports on driver alertness patterns. Drivers and fleet managers can access a dashboard displaying real-time insights, enabling better decision-making and proactive safety measures. The system's modular and scalable design ensures flexibility for future upgrades, making it an effective solution for reducing drowsy driving incidents.

**Advantages Over Existing System**

- Real-Time Analysis: Instantly processes and interprets driver movement data, allowing for immediate drowsiness detection and rapid alert mechanisms to prevent accidents.
- Cloud-Based Storage and Analytics: Utilizes remote servers for data storage and analysis, ensuring scalability, flexibility, and cost-effectiveness while maintaining continuous monitoring without requiring expensive on-premises infrastructure.
- Enhanced Data Accuracy and Consistency: Improves the reliability of drowsiness detection by ensuring uniform data collection and processing, minimizing false positives and enhancing system effectiveness.

## 3.3 System Specifications

### 3.3.1 Hardware Configuration

- **Processor**: AMD Ryen 7 or Intel Core i9 (for efficient data processing and real-time analysis)
- **Graphics Card**: Nvidia RTX 4090 (for enhanced visual processing and deep learning applications)
- **Storage**: Minimum 160GB SSD (preferred for faster data access and processing)
- **RAM**: 8GB or 16GB DDR5 (to support high-speed computation and smooth execution of real-time monitoring tasks)
- **Webcam**: Full HD camera (for accurate facial and behaviour monitoring during live video analysis)

### 3.3.2 Software Configuration

- **Operating System**: Windows 7/8/10 or Linux (ensuring compatibility with hardware and software tools)
- **IDE**: Arduino Integrated Development Environment (for microcontroller programming and sensor integration)
- **Libraries Used**: MPU6050 library (for motion sensor data processing) and wireless connectivity library (for data transmission)
- **Technology**: python & C (for firmware and algorithm implementation)

### 3.3.3 Functional Requirements

These requirements describe the necessary actions the system must execute to accomplish its objectives:

- **Drowsiness Detection**: Monitoring driver eye closure, head position, and body posture using motion sensors.
- **Automatic Call Alert**: Initiates an emergency call to predefined contacts when severe drowsiness is detected.
- **Distraction Monitoring**: Detects phone usage, excessive movement, or unusual behaviour indicating driver inattention.
- **Real-Time Data Processing**: Captures and analyses sensor data continuously for instant drowsiness detection.
- **Performance Metrics Calculation**: Computes fatigue levels based on movement patterns and predefined thresholds.
- **Statistical Analysis**: Generates insights on drowsiness trends, alert frequency, and driving behaviour patterns.
- **Visualization and Reporting**: Provides graphical dashboards for real-time and historical analysis.
- **Alert Logging System:** Maintains a timestamped record of all alerts for post-drive review and system evaluation.

### 3.3.4 Non-Functional Requirements

The non-functional requirements define the overall quality attributes of the Driver Drowsiness Detection and Monitoring System. These attributes ensure the system's usability, efficiency, and long-term reliability in real-world scenarios. While functional requirements focus on what the system does, non-functional requirements determine how well it performs those tasks under various conditions. These specify the system's standards for efficiency and overall functionality, ensuring reliability and efficiency:

- **Portability**: Compact hardware design for seamless integration into vehicles.
- **Performance**: Optimized for real-time data analysis with minimal latency.
- **Scalability**: Allows future enhancements such as AI-based fatigue prediction and additional sensors.
- **Reliability**: Ensures accurate detection and low false alarm rates.
- **Maintenance**: Provides regular system updates and calibration for continued efficiency.
- **User-Friendly Interface**: Intuitive dashboard with easy-to-navigate alert notifications and reports.
- **Security:** Ensures secure handling and storage of sensitive driver data through encryption and access control.
- **Compatibility:** Supports integration with multiple operating systems and vehicle platforms.
- **Responsiveness:** Provides timely alerts and visual feedback without noticeable delay.
- **Fault Tolerance:** Maintains system functionality in case of partial hardware or software failure.
- **Resource Efficiency:** Optimized usage of system resources to avoid unnecessary battery or CPU consumption.

# CHAPTER 4
# SYSTEM DESIGN

### 4.1 System Architecture

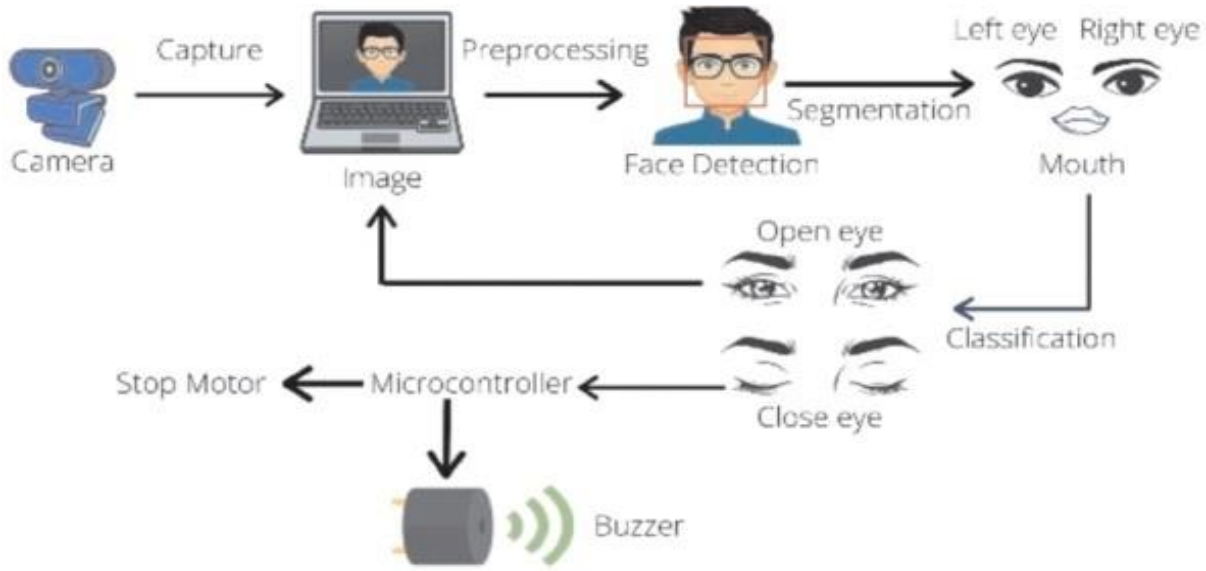



**Fig No 4.1: System Architecture**

The Fig no 4.1 depicts a system for identifying driver tiredness which utilizes a camera for monitoring, face detection, and a microcontroller-based alert mechanism.

**Working of the System:**

1. **Image Capture:**
    - A camera continuously captures the driver's facial image.
2. **Preprocessing & Face Detection:**
    - Captured images are processed by a setup that detects the driver's facial region.
3. **Feature Segmentation:**
    - Essential facial attributes, including the eyes and mouth are extracted for further analysis.
    - Head orientation and facial contour landmarks are identified to assist in detecting distraction and gaze direction.

4. **Eye State Classification:**

    - The system analyses the driver's eyes to identify whether they are open or closed.
    - If eyes stay shut for an extended duration, the system interprets it as an indication of driver drowsiness.

5. **Alert Mechanism:**

    - When indicators of driver fatigue are identified, microcontroller triggers a sound device to notify the driver
    - Additionally, the system can send a signal to stop the vehicle’s motor, ensuring safety.

### 4.2 Methodology of the proposed work

The integration of a fatigue monitoring system for drivers consists of key components such as real-time facial and eye tracking, blink pattern analysis, and an alert mechanism. The proposed framework aims to improve traffic safety by identifying signs of driver fatigue using a vision-based approach. The core methodology involves continuous tracking the driver's facial expressions particularly focusing on eye closure, blink frequency, and head posture, using a camera integrated into the vehicle dashboard. These visual cues are analysed in real-time through smart computational techniques for detecting signs of exhaustion or reduced alertness.

1. **Automatic Call Detection:**

    The system is designed to detect incoming phone calls while the vehicle is in motion. Upon identifying an active call, the system evaluates whether the driver is engaged in a conversation. If so, a visual or audio alert is triggered to remind the driver to focus on the road. This helps in minimizing distractions caused by mobile phone usage during driving.

2. **Distraction Alertness Detection:**

    Beyond drowsiness, the model also tracks sudden head movements, frequent gaze shifts away from the road, and unusual hand gestures to detect driver distraction

3. **Key Functions and Implementation Strategies:**

**Data Collection:** Dataset: A pre-recorded dataset of driver images or video footage is used. It includes both alert and drowsy states, with various scenarios such as different conditions, head poses, and drivers' facial features (e.g., glasses, facial hair).

**Technologies:** The system relies on Software Libraries such as OpenCV for image analysis, and Haar Cascade for facial recognition, and dib for mapping facial landmarks.

**Face Detection**: The initial step involves identifying the driver's face in a video. stream using Haar Cascades or more advanced techniques like HOG features for face detection.

**Eye Aspect Ratio (EAR) Calculation:** The ratio of the eye's aspect (EAR) is calculated and analysing distance among defined eye markers. When the EAR drops below a predefined limit, signalising eye closure, the system detects signs of driver fatigue.

**Formula for EAR:**

$$EAR = \frac{(d1 + d2)}{2 \times d3}$$

where d1, d2, d3 are distance among defined eye markers.

4. **Dataset:**

**Video Capture**: A camera captures video frames at regular intervals, which are processed frame-by-frame to track facial features. The video stream is obtained using OpenCV, a robust library widely used for computer vision tasks and display the frames for further processing.

**Frame Selection:** Frame selection helps optimize performance by selecting specific frames for processing. Instead of processing every frame in a video, frames can be skipped at regular reducing computational load without significantly affecting the outcome.

**Eye Detection:**

Alert: If the driver is fully attentive (eyes open, no yawning).

Drowsy: If the system detects closed eyes (EAR below threshold) or frequent yawning. The output might can activate a warning sound or deliver a notification if drowsiness is detected

## 4.3 UML diagram

A UML diagram visually represents the design and behaviour of a system. It includes elements like classes, objects, use cases, and interactions. These diagrams help in understanding system architecture, data flow, and user interactions. UML improves communication among developers, simplifies planning, and supports efficient software development and documentation.

### 4.3.1 Use Case diagram

A UML an outline or schematic that depicts behavioural model created through use-case analysis to depict system interactions. It visually showcases how the system functions by highlighting the interactions between various actors and their associated goals.

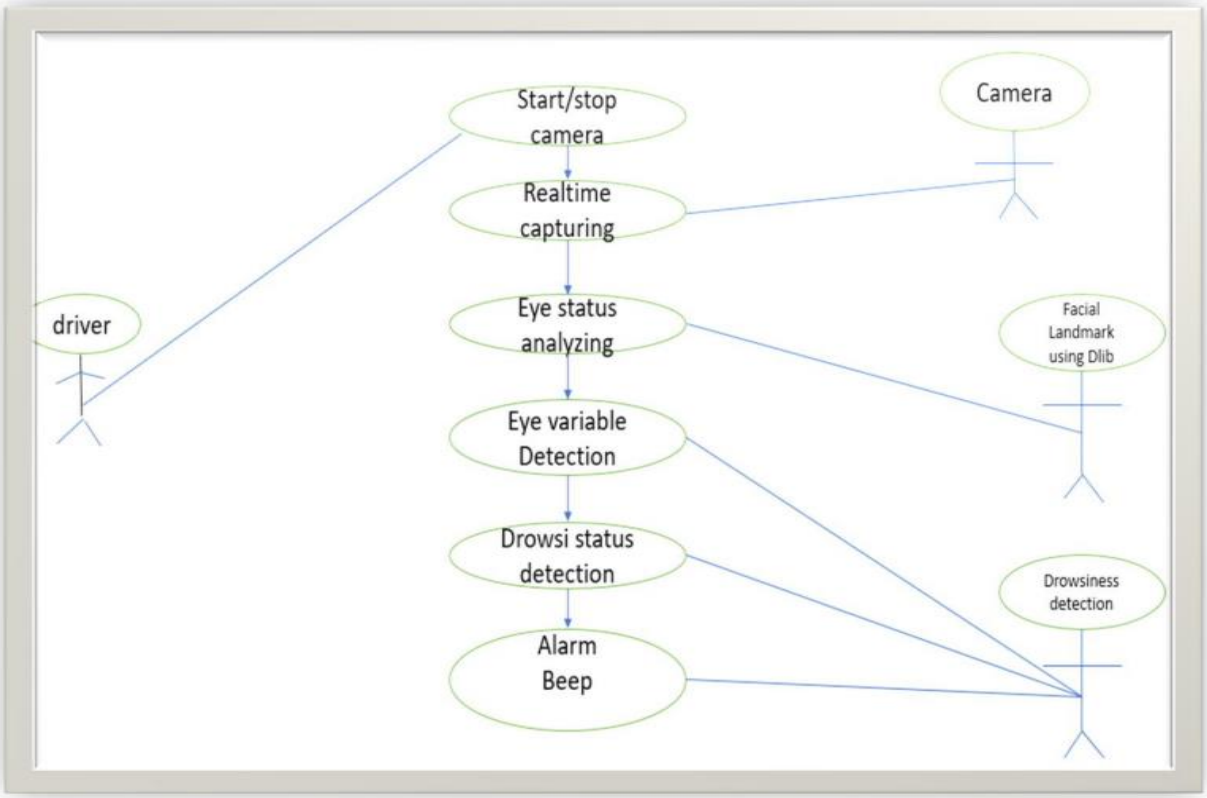



**Fig No 4.2: Use Case Diagram**

The Fig no 4.2 seeks to identify the system operations are linked to particular users or roles. It clearly maps out the responsibilities of each use case, presenting the complete workflow of the system from user login to logout in a straightforward and easy-to understand format.

### 4.3.2 Class Diagram

It provides a visual representation of the static structure of a system, showcasing various components of an application. It offers a high-level perspective and, when combined with multiple class diagrams, helps illustrate the entire system. To create an effective class diagram, several factors should be considered. The diagram's name should be meaningful and accurately represent the aspect of the system.

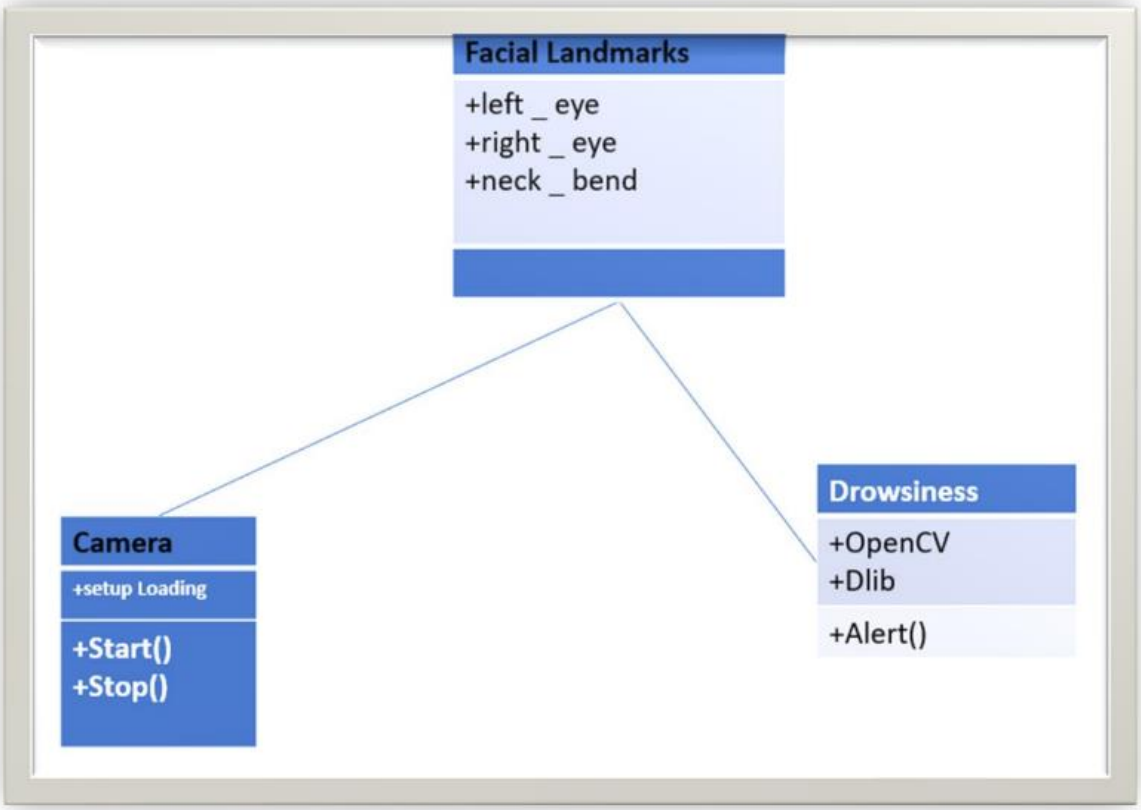



**Fig No 4.3: Class Diagram**

The Fig no 4.3 is to identify all elements and their relationships beforehand to ensure clarity. Each class should have well-defined responsibilities, including attributes and methods, while maintaining only essential to avoid making the diagram overly complex, it's important to keep only the essential properties. Including brief notes or comments can offer better clarity on particular elements of the diagram. In the end, the diagram should be organized in a simple and logical manner, making it easy for developers to interpret and apply during implementation.

### 4.3.3 Sequence Diagram

It represents how different system components interact over time in a specific order. It provides a visual layout of the flow of messages or actions exchanged within a system, clearly indicating the sequence in which they take place.

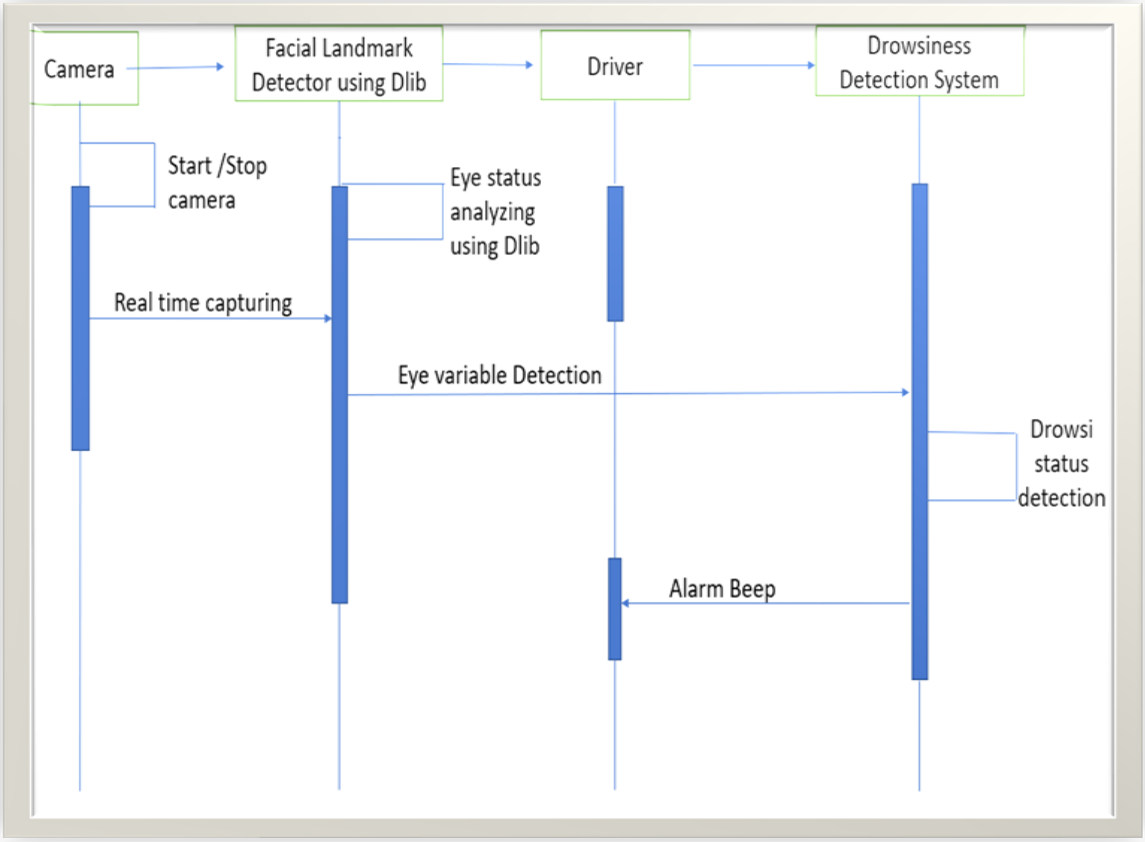



**Fig No 4.4: Sequence Diagram**

The Fig no 4.4 shows the above diagram are sometimes also known as event scenarios, timing diagrams, or event interaction diagrams, as they highlight the timing and flow of communication between components. The diagram below demonstrates the step-by-step progression of the system, showcasing how different users interact with it in a sequential manner.

### 4.3.4 Activity Diagram

It visually represents workflows in a step-by-step manner, illustrating activities and actions while supporting decision-making, repetition, and parallel processes. In Unified Modelling Language (UML), these diagrams are useful for describing both business and operational workflows within a system. The diagram below effectively outlines the activities of each entity in a clear and structured format, making the workflow easily understandable.

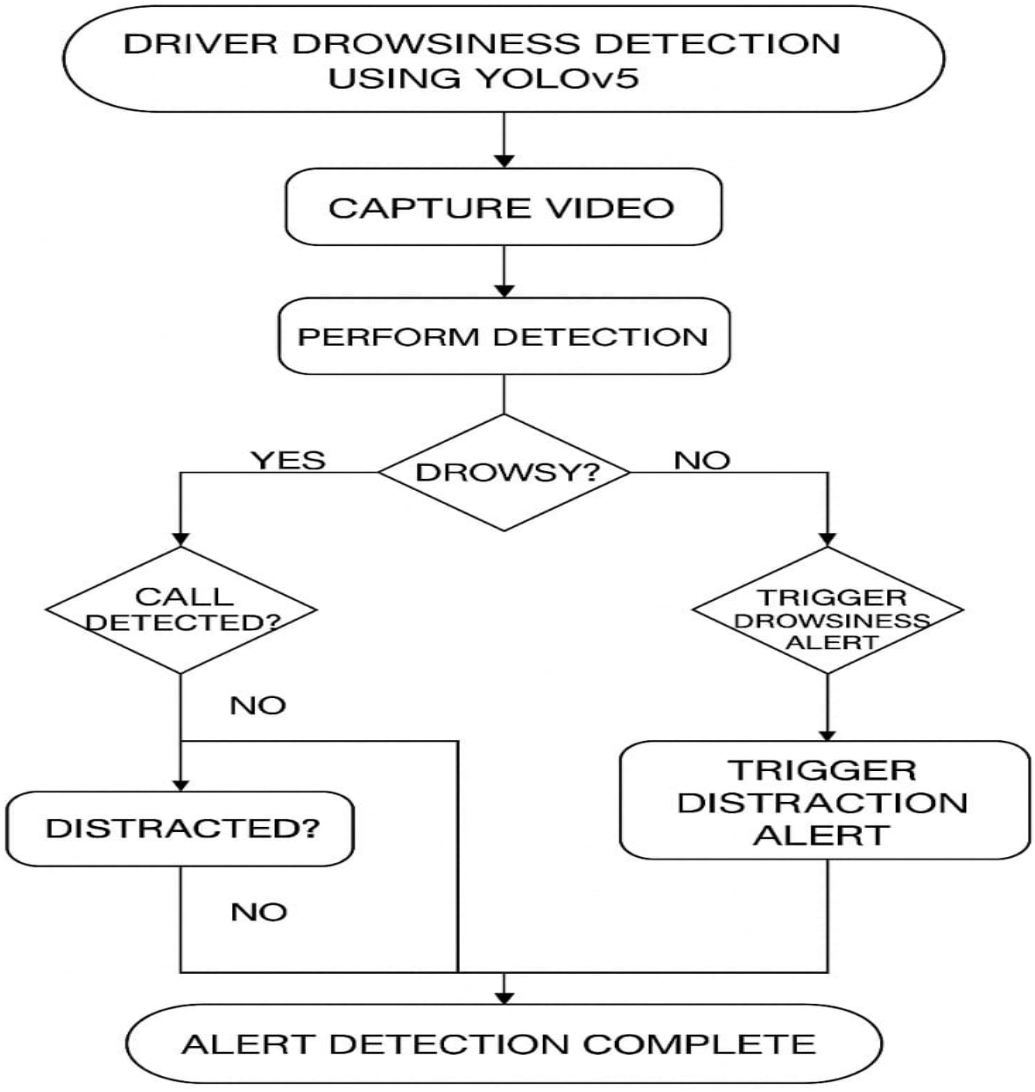



**Fig No 4.5: Activity Diagram**

The Fig no 4.5 the process flow of a drowsiness detection system. It begins with camera activation, followed by Facial features and eyes detection to captured Footage frames. Upon identifying indications of fatigue, the system activates a warning and then terminates the application.

## 4.4 Data flow Diagram

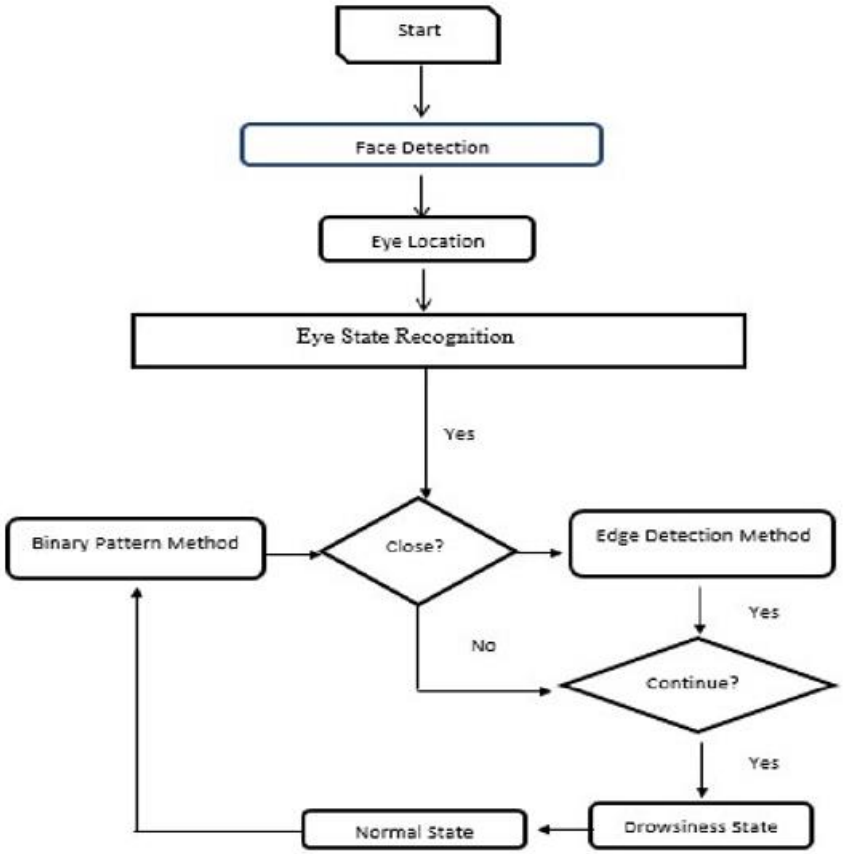



**Fig No 4.6: Data Flow diagram**

The Fig no 4.6 shows the flow of:

1. The Input Module captures video frames.
2. Frames pass through Preprocessing to standardize data.
3. The Face Detection module locates the face.
4. Facial Landmarks are identified for feature extraction.
5. The Feature Extraction Module calculates EAR and MAR.
6. The Drowsiness Detection module evaluates the driver's state.

# CHAPTER 5

# IMPLEMENTATION AND RESULTS

## 5.1 Parameter Settings

The Driver Drowsiness Detection System is configured with specific hardware and software parameters to ensure accurate and efficient monitoring. The system incorporates additional features such as real-time alerts, cloud-based data storage, and AI-based analysis to enhance detection capabilities.

**1. Hardware Parameter Settings:**

- Camera Quality**:** A resolution of at least 720p is recommended to accurately detect and track facial features.
- Frame Rate: 30 FPS for smooth image processing and real-time monitoring.
- Gyroscope & Accelerometer Sensor: MPU6050 for detecting sudden head movements or posture changes.
- Buzzer Sound Level: 85-100 dB to ensure audible alerts in noisy environments.
- Motor Control Delay: 1-2 seconds before triggering vehicle safety mechanisms.

**2. Software Parameter Settings:**

- Face Detection Model: Haar cascade or CNN-based deep learning model for accurate facial recognition.
- Eye Aspect Ratio (EAR) Threshold: $\leq 0.25$ (Eyes closed if below threshold for a specified duration).
- Blink Frequency Detection: Average of 15-20 blinks per minute considered normal; sudden drop indicates drowsiness.
- Mouth Aspect Ratio (MAR) Threshold: $\geq 0.5$ (Indicates yawning, a sign of fatigue).
- Data Processing Interval: Every 1 second for Ongoing observation and evaluation.
- Cloud Connectivity**:** Data is uploaded to cloud platforms for analysis and storage, allowing access and monitoring from remote locations.
- Alert Trigger Time: If drowsiness is detected for 5 seconds or more, an alert is activated.
- Distraction Detection: Monitors head movements and phone usage patterns any significant deviation from normal driving posture triggers a distraction alert.

**3. Additional Features:**

- Real-time Alert System: Immediate buzzer activation and motor control upon detecting drowsiness.
- Cloud-Based Analytics: Stores historical data for performance tracking and further analysis.
- AI-Based Prediction: Uses machine learning to predict drowsiness trends based on previous behaviour.
- Scalability: Can integrate with IoT systems for fleet management and road safety enhancements.

## 5.2 Method of Implementation

The system designed to detect driver fatigue is developed using an organized approach that integrates hardware components, software solutions, machine learning techniques to consistently monitor the driver's alertness. It functions through a step-by-step process of detecting, interpreting, and reacting to drowsiness indicators, ultimately enhancing road safety.

**1. Data Acquisition**

- A camera module consistently captures live video capturing the driver's facial characteristics.
- A camera module consistently captures live footage depicting the driver's face traits actions and posture changes.
- The collected data is transmitted to a microcontroller (ESP8266/ESP32) for processing.

**2. Preprocessing & Feature Extraction**

- The system applies face detection algorithms (Haar cascades or CNN models) to detect important facial landmarks.
- The areas around the eyes and mouth are extracted for detailed assessment.
- The system calculates EAR and MAR to track drowsiness indicators like prolonged eye closure and frequent.
- The system also tracks head orientation and movement patterns to detect signs of distraction or inattentiveness, further enhancing the monitoring of driver alertness.

**3.Drowsiness Detection Algorithm:**

A predefined EAR threshold (≤ 0.25) determines If the driver's eyelids remain closed for a prolonged duration blink frequency is monitored, with sudden drops indicating fatigue.

- If the MAR exceeds 0.5, it detects yawning, another sign of drowsiness.
- Head tilt detection using gyroscope data helps identify inattentiveness.
- A machine learning model improves detection accuracy by analysing behavioural patterns.

4**. Fatigue, Yawning & Distraction Detection**

- **Eye Closure Detection**: If The EAR falls below a set limit (typically ≤ 0.25) for more than 4 seconds, it signals possible drowsiness.
- **Yawning Detection**: If MAR stays above a defined level (e.g., ≥ 0.5) continuously for 4 seconds, it is considered a yawn, triggering a fatigue alert.
- **Automatic Call Detection**: The system monitors both hands; if either remains near the ear for more than 10 seconds while the vehicle is in motion, it assumes a phone call and triggers an alert.
- **Distraction Detection**: Head pose estimation is utilized to assess whether the driver's attention is diverted from the road (left or right) for more than 4 seconds. If so, the system generates a distraction alert.

**5. Alert Mechanism & Safety Response**

- If drowsiness is detected for a continuous 5-second interval, an alarm buzzer is activated.
- The system transmits a command to the microcontroller to activate seat vibrations or trigger an audio warning.
- In severe cases, the system can trigger vehicle speed reduction or send alerts to connected Cloud-based systems for remote observation.
- The system logs all detected alerts, including timestamps and the type of event, for post-drive analysis and evaluation of driver behaviour patterns.

## 5.3 Source Code

```
from google.colab import drive

drive.mount('/content/drive')

%cd /content/drive/MyDrive/new_driver_drowsiness_project

!git clone https://github.com/ultralytics/yolov5

%cd yolov5

%pip install -qr requirements.txt comet_ml

import torch

import utils

display = utils.notebook_init()

!python train.py --img 640 --batch 32 --epochs 100 --data
/content/drive/MyDrive/new_driver_drowsiness_project/yolov5/data/custom.yaml --weights
yolov5s.pt --cache

!python detect.py --source
/content/drive/MyDrive/new_driver_drowsiness_project/yolov5/runs/train/exp/frame_0007.jp
g --weights
/content/drive/MyDrive/new_driver_drowsiness_project/yolov5/runs/train/exp/weights/best.p
t

import torch

import pathlib

temp = pathlib.PosixPath

pathlib.PosixPath = pathlib.WindowsPath

import torch

import pathlib

temp = pathlib.PosixPath

pathlib.PosixPath = pathlib.WindowsPath

model = torch.hub.load(r"C:\Users\manga\driver_drowsy_project_\yolov5", 'custom',

path=r"C:\Users\manga\driver_drowsy_project_\yolov5\runs\train\exp\weights\best.pt",

                source='local',

                force_reload=True)

import matplotlib.pyplot as plt
```

```
import numpy as np
results =
model(r"C:\Users\manga\driver_drowsy_project_\Dataset\train\images\frame_0004.jpg")
results.print()
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
import matplotlib.pyplot as plt
import numpy as np
results =
model(r"C:\Users\manga\driver_drowsy_project_\Dataset\train\images\frame_0331.jpg")
results.print()
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
import matplotlib.pyplot as plt
import numpy as np
results =
model(r"C:\Users\manga\driver_drowsy_project_\Dataset\train\images\frame_0378.jpg")
results.print()
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
!pip install torch
import torch
import pathlib
temp = pathlib.PosixPath
pathlib.PosixPath = pathlib.WindowsPath
%cd  yolov5
model = torch.hub.load(
   ",
```

```
    'custom',
path=r"C:\Users\manga\driver_drowsy_project_\yolov5\runs\train\exp\weights\best.pt",
    source='local',
    force_reload=True
)
print(model)
import torch
import cv2
import numpy as np
import time
import winsound
import warnings
warnings.filterwarnings("ignore")
model.conf = 0.3
model.iou = 0.4
model.to('cuda' if torch.cuda.is_available() else 'cpu')
input_video_path = r"C:\Users\manga\driver_drowsy_project_\video1.mp4"
cap = cv2.VideoCapture(input_video_path)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
fps = int(cap.get(cv2.CAP_PROP_FPS))
fourcc = cv2.VideoWriter_fourcc(*'XVID')
distraction_start_time = None
distraction_threshold = 2
class_labels = {
    0: "Call Detection",
    1: "Focused",
    2: "Distracted",
    3: "Drowsy"
}
```

```
alert_classes = {0, 2, 3}
frame_skip = 2
frame_count = 0
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    frame_count += 1
    if frame_count % frame_skip != 0:
        continue
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model(frame_rgb, size=640)
    driver_focused = False
    for *box, conf, cls in results.xyxy[0]:
        cls_id = int(cls)
        label = f"{class_labels[cls_id]} {conf:.2f}"
        x1, y1, x2, y2 = map(int, box)
        color = (0, 255, 0) if cls_id == 1 else (0, 0, 255)
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, label, (x1, y1 - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
        if cls_id == 1:
            driver_focused = True
        if cls_id in alert_classes:
            if distraction_start_time is None:
                distraction_start_time = time.time()
            elif time.time() - distraction_start_time > distraction_threshold:
                print(f" 🚨 ALERT! Driver is {class_labels[cls_id]} for more than 2 seconds! 🚨 ")
                winsound.Beep(1000, 500)
        else:
```

```
            distraction_start_time = None
    if driver_focused:
        distraction_start_time = None
    cv2.imshow("Driver Monitoring", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
import torch
import cv2
import numpy as np
import time
import winsound
model.conf = 0.3
model.iou = 0.4
model.to('cuda' if torch.cuda.is_available() else 'cpu')
input_video_path = r"C:\Users\manga\driver_drowsy_project_\video2.mp4"
cap = cv2.VideoCapture(input_video_path)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
fps = int(cap.get(cv2.CAP_PROP_FPS))
fourcc = cv2.VideoWriter_fourcc(*'XVID')
distraction_start_time = None
distraction_threshold = 2
class_labels = {
    0: "Call Detection",
    1: "Focused",
    2: "Distracted",
    3: "Drowsy"
}
```

```
alert_classes = {0, 2, 3}
frame_skip = 2
frame_count = 0
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    frame_count += 1
    if frame_count % frame_skip != 0:
        continue
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model(frame_rgb, size=640)
    driver_focused = False
    for *box, conf, cls in results.xyxy[0]:
        cls_id = int(cls)
        label = f"{class_labels[cls_id]} {conf:.2f}"
        x1, y1, x2, y2 = map(int, box)
        color = (0, 255, 0) if cls_id == 1 else (0, 0, 255)
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, label, (x1, y1 - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
        if cls_id == 1:
            driver_focused = True
        if cls_id in alert_classes:
            if distraction_start_time is None:
                distraction_start_time = time.time()
            elif time.time() - distraction_start_time > distraction_threshold:
                print(f" 🚨 ALERT! Driver is {class_labels[cls_id]} for more than 2 seconds! 🚨 ")
                winsound.Beep(1000, 500)
        else:
```

```
distraction_start_time = None
    if driver_focused:
        distraction_start_time = None
    cv2.imshow("Driver Monitoring", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
import torch
import cv2
import numpy as np
import time
import winsound
model.conf = 0.3
model.iou = 0.4
model.to('cuda' if torch.cuda.is_available() else 'cpu')
input_video_path = r"C:\Users\manga\driver_drowsy_project_\video2.mp4"
cap = cv2.VideoCapture(input_video_path)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
fps = int(cap.get(cv2.CAP_PROP_FPS))
fourcc = cv2.VideoWriter_fourcc(*'XVID')
distraction_start_time = None
distraction_threshold = 2
class_labels = {
    0: "Call Detection",
    1: "Focused",
    2: "Distracted",
    3: "Drowsy"
}
```

```
alert_classes = {0, 2, 3}
frame_skip = 2
frame_count = 0
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    frame_count += 1
    if frame_count % frame_skip != 0:
        continue
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model(frame_rgb, size=640)
    driver_focused = False
    for *box, conf, cls in results.xyxy[0]:
        cls_id = int(cls)
        label = f"{class_labels[cls_id]} {conf:.2f}"
        x1, y1, x2, y2 = map(int, box)
        color = (0, 255, 0) if cls_id == 1 else (0, 0, 255)
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, label, (x1, y1 - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
        if cls_id == 1:
            driver_focused = True
        if cls_id in alert_classes:
            if distraction_start_time is None:
                distraction_start_time = time.time()
            elif time.time() - distraction_start_time > distraction_threshold:
                print(f" 🚨 ALERT! Driver is {class_labels[cls_id]} for more than 2 seconds! 🚨 ")
                winsound.Beep(1000, 500)
        else:
```

```
distraction_start_time = None
    if driver_focused:
        distraction_start_time = None
    cv2.imshow("Driver Monitoring", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
import torch
import cv2
import numpy as np
import time
import winsound
model.conf = 0.3
model.iou = 0.4
model.to('cuda' if torch.cuda.is_available() else 'cpu')
input_video_path = r"C:\Users\manga\driver_drowsy_project_\video3.mp4"
cap = cv2.VideoCapture(input_video_path)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
fps = int(cap.get(cv2.CAP_PROP_FPS))
fourcc = cv2.VideoWriter_fourcc(*'XVID')
distraction_start_time = None
distraction_threshold = 10
class_labels = {
    0: "Call Detection",
    1: "Focused",
    2: "Distracted",
    3: "Drowsy"
}
```

```
alert_classes = {0, 2, 3}
frame_skip = 2
frame_count = 0
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    frame_count += 1
    if frame_count % frame_skip != 0:
        continue
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model(frame_rgb, size=640)
    driver_focused = False
    for *box, conf, cls in results.xyxy[0]:
        cls_id = int(cls)
        label = f"{class_labels[cls_id]} {conf:.2f}"
        x1, y1, x2, y2 = map(int, box)
        color = (0, 255, 0) if cls_id == 1 else (0, 0, 255)
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, label, (x1, y1 - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
        if cls_id == 1:
            driver_focused = True
        if cls_id in alert_classes:
            if distraction_start_time is None:
                distraction_start_time = time.time()
            elif time.time() - distraction_start_time > distraction_threshold:
                print(f" ALERT! Driver is {class_labels[cls_id]} for more than 10 seconds!
 ")
                winsound.Beep(1000, 500)
```

```
        else:
            distraction_start_time = None
    if driver_focused:
        distraction_start_time = None
    cv2.imshow("Driver Monitoring", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
# Import necessary libraries
from scipy.spatial import distance # Fix: Correct lowercase import
from imutils import face_utils
import numpy as np # Fix: Correct lowercase import
import pygame # For playing sound
import time
import dlib
import cv2
# Load Dlib's shape predictor model
predictor_path = r"C:\Users\polur\Downloads\shape_predictor_68_face_landmarks.dat"
predictor = dlib.shape_predictor(predictor_path)
# Initialize Pygame and load an audio file (e.g., alarm sound)
pygame.mixer.init()
pygame.mixer.music.load(r"C:\Users\polur\Downloads\Driver Drowsiness
Detection_music.wav") # Use your actual audio file
pygame.mixer.music.play()
#Minimum threshold of eye aspect ratio below which alarm is triggered
EYE_ASPECT_RATIO_THRESHOLD = 0.3
#Minimum consecutive frames for which the eye ratio is below the threshold for the alarm to
be triggered
EYE_ASPECT_RATIO_CONSEC_FRAMES = 50
```

```
#COunts no. of consecutive frames below a threshold value
COUNTER = 0
#Load face cascade which will be used to draw a rectangle around detected faces.
face_cascade =
cv2.CascadeClassifier(r"C:\Users\polur\Downloads\haarcascade_frontalface_default.xml")
!pip install dlib
import cv2
import dlib # Make sure to import dlib
# Load face detector and shape predictor
detector = dlib.get_frontal_face_detector()
predictor =
dlib.shape_predictor(r"C:\Users\polur\Downloads\shape_predictor_68_face_landmarks.dat")
# Capture video from webcam
cap = cv2.VideoCapture(0)
if not cap.isOpened():
 print("❌ Error: Could not open camera.")
 exit()
while True:
 ret, frame = cap.read()
 if not ret or frame is None:
 print("❌ Error: Failed to capture frame. Retrying...")
 continue
 # Convert the frame to grayscale
 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
 # Detect faces in the image
 faces = detector(gray)
 for face in faces:
 # Predict the facial landmarks
 landmarks = predictor(gray, face)
 # (You can now access facial landmarks like landmarks.part(36) for the left eye, etc.)
```

```
 # Display the frame with detected faces
 cv2.imshow("Face Detection", frame)
 if cv2.waitKey(1) & 0xFF == ord("q"):
 break
 cap.release()
 cv2.destroyAllWindows()
from imutils import face_utils
import time
import cv2
import dlib
from imutils import face_utils # Import face_utils from imutils
# Load face detector and shape predictor
detector = dlib.get_frontal_face_detector()
predictor=dlib.shape_predictor(r"C:\Users\polur\Downloads\shape_predictor_68_face_landm
arks.d
at")
# Extract indexes of facial landmarks for the left and right eye
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS['left_eye']
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS['right_eye']
# Start webcam video capture
cap = cv2.VideoCapture(0)
if not cap.isOpened():
 print("❌ Error: Could not open camera.")
 exit()
while True:
 ret, frame = cap.read()
 if not ret or frame is None:
 print("❌ Error: Failed to capture frame. Retrying...")
 continue
 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
 # Detect faces
 faces = detector(gray)
 for face in faces:
 # Predict the facial landmarks
 landmarks = predictor(gray, face)
 # Convert the landmarks to a numpy array
 landmarks = face_utils.shape_to_np(landmarks)
 # Draw landmarks around the eyes
 for (x, y) in landmarks[lStart:lEnd]:
 cv2.circle(frame, (x, y), 1, (0, 255, 0), -1)
for (x, y) in landmarks[rStart:rEnd]:
 cv2.circle(frame, (x, y), 1, (0, 0, 255), -1)
 # Display the frame
 cv2.imshow("Eye Landmark Detection", frame)
if cv2.waitKey(1) & 0xFF == ord("q"):
 break
cap.release()
cv2.destroyAllWindows()
import cv2
# Load the Haar cascade for face detection
face_cascade =
cv2.CascadeClassifier(cv2.data.haarcascades+"haarcascade_frontalface_default.xml")
# Initialize video capture (webcam)
cap = cv2.VideoCapture(0)
if not cap.isOpened():
 print("❌ Error: Could not open camera.")
 exit()
while True:
 ret, frame = cap.read()
 if not ret or frame is None:
```

```
 print("❌ Error: Failed to capture frame.")
 continue
 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
# Detect faces in the grayscale image
 face_rectangle = face_cascade.detectMultiScale(gray, 1.3, 5)
 # Draw rectangle around each face detected
 for (x, y, w, h) in face_rectangle:
 cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
# Show the resulting frame
 cv2.imshow("Face Detection", frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
 break
# Release the video capture and close windows
cap.release()
cv2.destroyAllWindows()
import cv2
import dlib
import pygame
from imutils import face_utils
from scipy.spatial import distance
import time
# This function calculates and returns eye aspect ratio (EAR)
def eye_aspect_ratio(eye):
 A = distance.euclidean(eye[1], eye[5])
 B = distance.euclidean(eye[2], eye[4])
 C = distance.euclidean(eye[0], eye[3])
 ear = (A + B) / (2 * C)
 return ear
# Load face detector and predictor using dlib shape predictor file
detector = dlib.get_frontal_face_detector()
```

```
predictor =
dlib.shape_predictor(r"C:\Users\polur\Downloads\shape_predictor_68_face_landmarks.dat")
# Extract indexes of facial landmarks for the left and right eye
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS['left_eye']
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS['right_eye']
# Load Haar Cascade for face detection
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
"haarcascade_frontalface_default.xml")
# Initialize variables for drowsiness detection
EYE_ASPECT_RATIO_THRESHOLD = 0.25 # Adjust this threshold if needed
EYE_ASPECT_RATIO_CONSEC_FRAMES = 30 # Number of consecutive frames before
triggering the alert
COUNTER = 0
# Initialize pygame mixer for sound
pygame.mixer.init()
# Start webcam video capture
video_capture = cv2.VideoCapture(0)
# Give some time for the camera to initialize (optional)
time.sleep(2)
while True:
 # Read each frame, flip it and convert to grayscale
 ret, frame = video_capture.read()
 if not ret or frame is None:
 print("❌ Error: Failed to capture frame.")
 continue
 frame = cv2.flip(frame, 1)
 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
 # Detect faces in the frame using Haar Cascade
 face_rectangle = face_cascade.detectMultiScale(gray, 1.3, 5)
 # Draw rectangle around each detected face
```

```
 for (x, y, w, h) in face_rectangle:
 cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
# Detect facial points using dlib's detector
 faces = detector(gray, 0)
# Process each detected face
 for face in faces:
 shape = predictor(gray, face)
 shape = face_utils.shape_to_np(shape)
 # Get array of coordinates for the left and right eye
 leftEye = shape[lStart:lEnd]
rightEye = shape[rStart:rEnd]
 # Calculate the aspect ratio of both eyes
 leftEyeAspectRatio = eye_aspect_ratio(leftEye)
 rightEyeAspectRatio = eye_aspect_ratio(rightEye)
 # Calculate the average eye aspect ratio
 eyeAspectRatio = (leftEyeAspectRatio + rightEyeAspectRatio) / 2
 # Draw contours around eyes
 leftEyeHull = cv2.convexHull(leftEye)
 rightEyeHull = cv2.convexHull(rightEye)
 cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
 cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)
 # Check if the eye aspect ratio is below the threshold
 if eyeAspectRatio < EYE_ASPECT_RATIO_THRESHOLD:
 COUNTER += 1
 # If the number of frames exceeds the threshold, trigger the alert
 if COUNTER >= EYE_ASPECT_RATIO_CONSEC_FRAMES:
 if not pygame.mixer.music.get_busy(): # Play music if not already playing
 pygame.mixer.music.load(r"C:\Users\polur\Downloads\Driver Drowsiness
Detection_music.wav") # Replace with your alarm sound path
 pygame.mixer.music.play(-1)
```

```
 cv2.putText(frame, "You are Drowsy", (150, 200),
cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0, 0, 255), 2)
 else:
 pygame.mixer.music.stop()
 COUNTER = 0
 # Display the video feed
 cv2.imshow('Video', frame)
 # Exit the loop if 'q' is pressed
 if cv2.waitKey(1) & 0xFF == ord('q'):
 break
# Release the video capture and close all windows
video_capture.release()
cv2.destroyAllWindows()
```

## 5.4 Output Screenshots

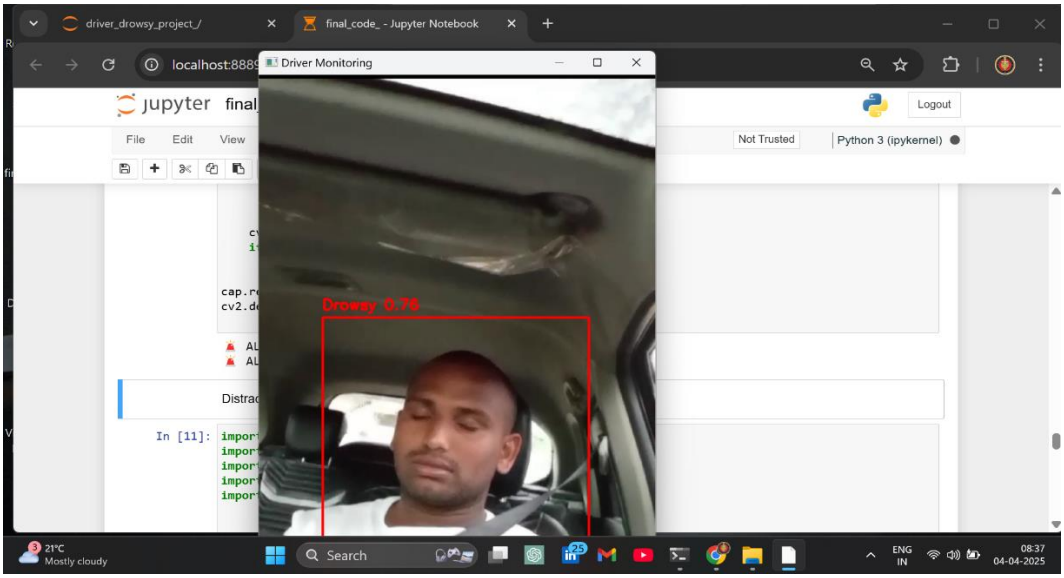



**Fig no 5.1: Drowsy state**

Fig no 5.1 drowsy state the system accurately detects the driver's drowsy state in real-time using the YOLOv5 model with a confidence score of 0.76. The driver's face is highlighted with a red bounding box, indicating fatigue based on closed eyes and head posture.

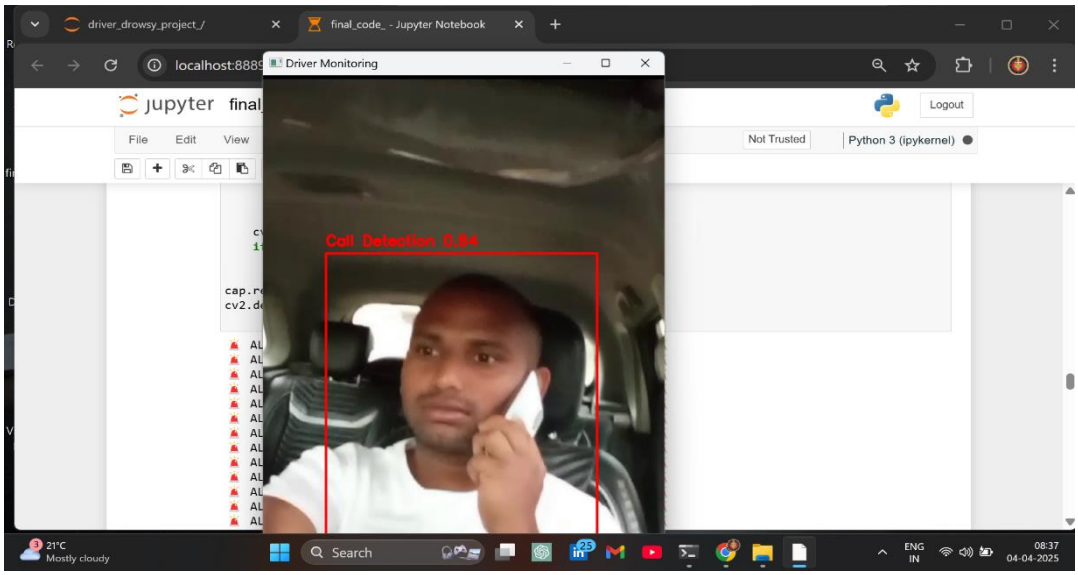



**Fig no 5.2: Call Detection**

The Fig no 5.2 Call detection this image shows a Driver Monitoring system detecting a driver using a phone while driving. A red bounding box with the label "Call Detection 0.84" indicates high confidence in detecting the activity.

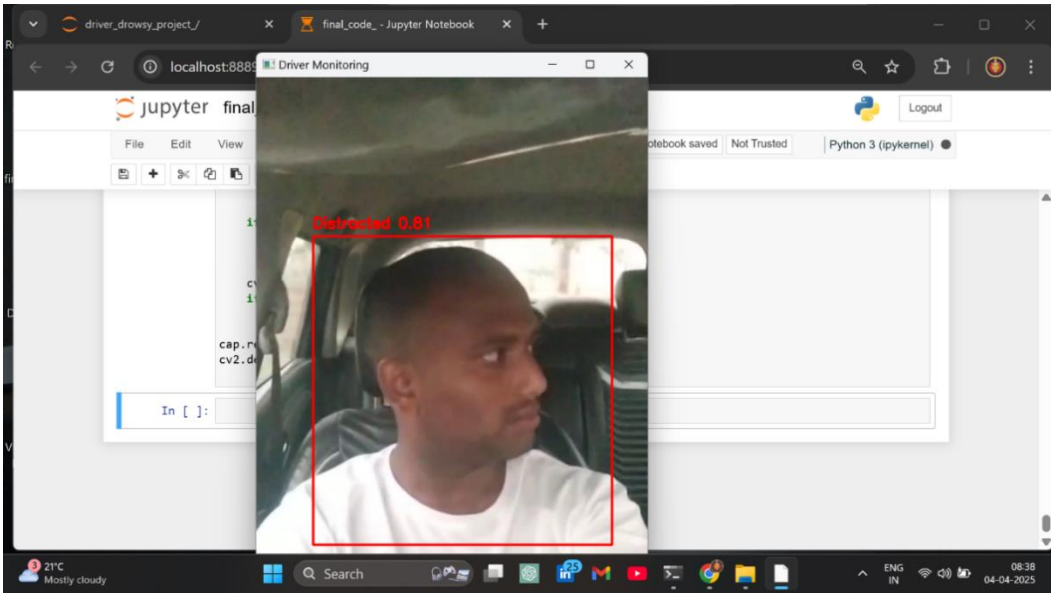



**Fig no 5.3: Distraction**

The Fig no 5.3 distraction explains the system successfully detects driver distraction with a confidence score of 0.81 using the YOLOv5 model. The driver is looking away from the road, which is highlighted by a red bounding box in the live video.


**Fig no 5.4: Drowsy state**

The Fig no 5.4.4 Drowsy state this image shows a drowsiness detection system identifying a person as "Drowsy" using facial landmarks. A blue bounding box highlights the face, and green lines around the eyes indicate eye monitoring. The system provides a real-time alert with the text "You are Drowsy" to warn the user.

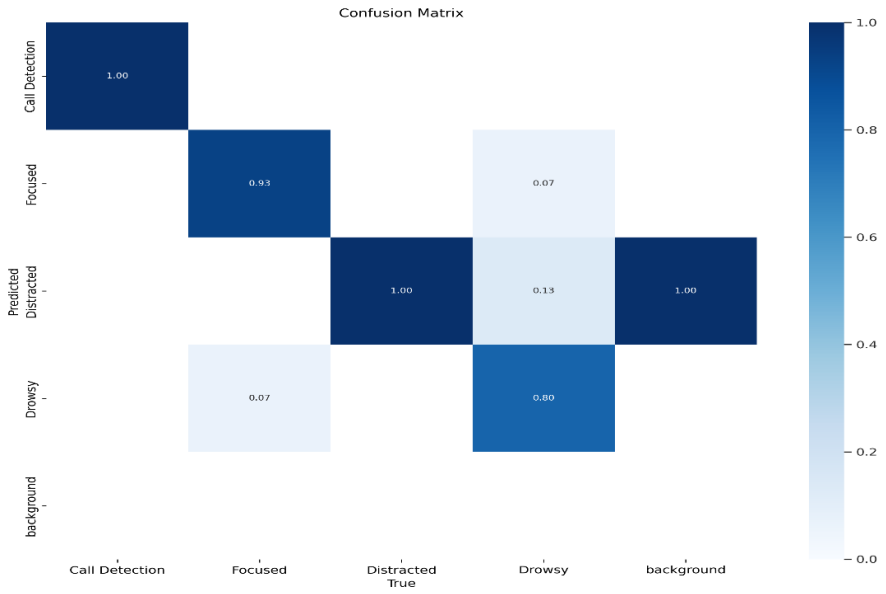



**Fig no 5.5: Confusion Matrix**

The Fig no 5.5 displays a Confusion Matrix for a driver monitoring system, evaluating classes like Call Detection, Focused, Distracted, Drowsy, and Background. The matrix shows high accuracy. there is slight confusion between Focused and Drowsy classes, where Focused has a 0.93 accuracy and Drowsy has an 0.80 accuracy.

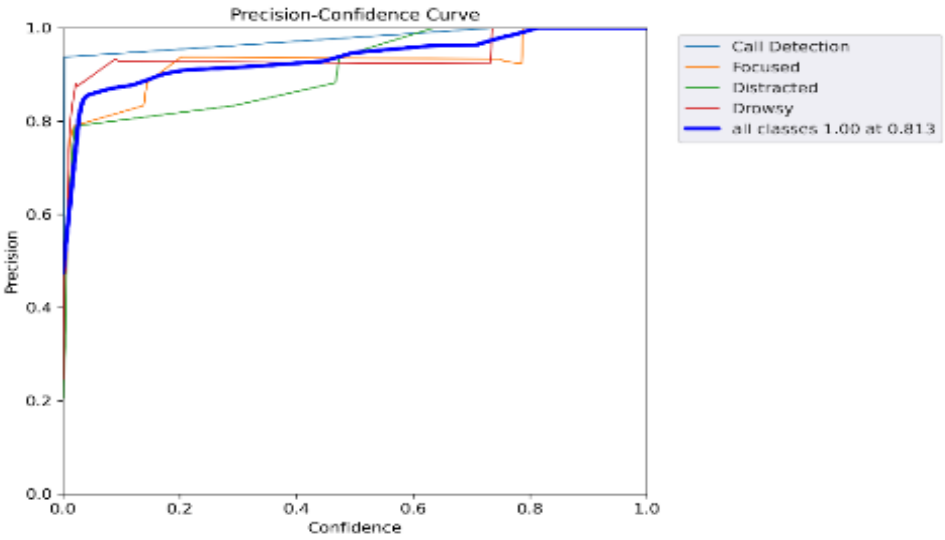



**Fig no 5.6: Precision-Confidence curve**

The Fig no 5.6 The thick blue line represents the combined performance of all classes, achieving 100% precision at 0.813 confidence. Individual class curves demonstrate high precision, with Call Detection performing consistently better across confidence thresholds.

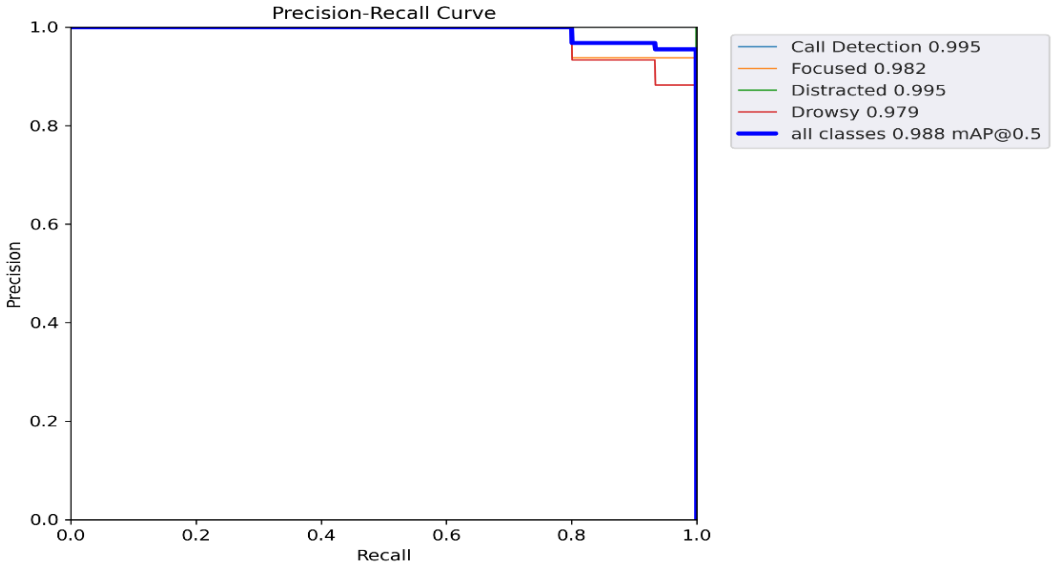



**Fig no 5.7: Precision-Recall curve**

The Fig no 5.7 This image presents a Precision-Recall Curve evaluating the model's performance across four classes: Call Detection, Focused, Distracted, and Drowsy. The model achieves high precision for all classes, with Call Detection and Distracted scoring the highest at 0.995, and an overall mAP@0.5 of 0. 988.The curve shows excellent precision and recall trade-offs, indicating reliable classification even under varying detection thresholds.

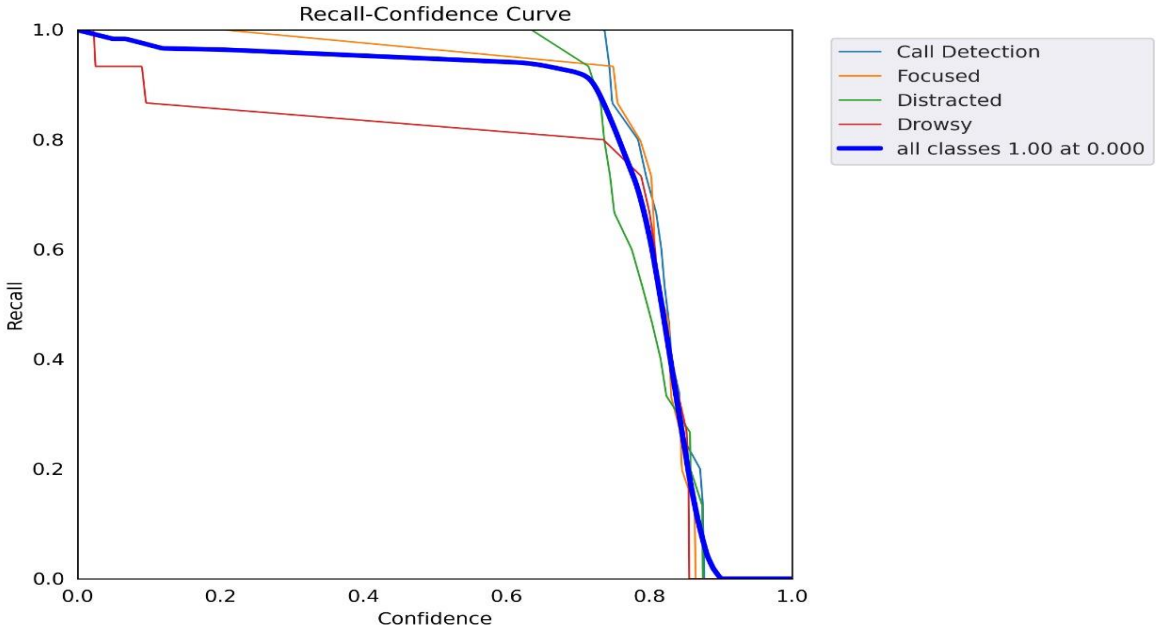



**Fig no 5.8: Recall - Confidence curve**

The Fig 5.8 shows a Recall-Confidence Curve for various driver behaviour classes Call Detection, Focused, Distracted, and Drowsy. Recall remains high across all classes at lower confidence thresholds, especially for Focused and Call Detection, indicating strong sensitivity.

As confidence increases beyond 0.8, recall begins to drop, highlighting the trade-off between certainty and missed detections.

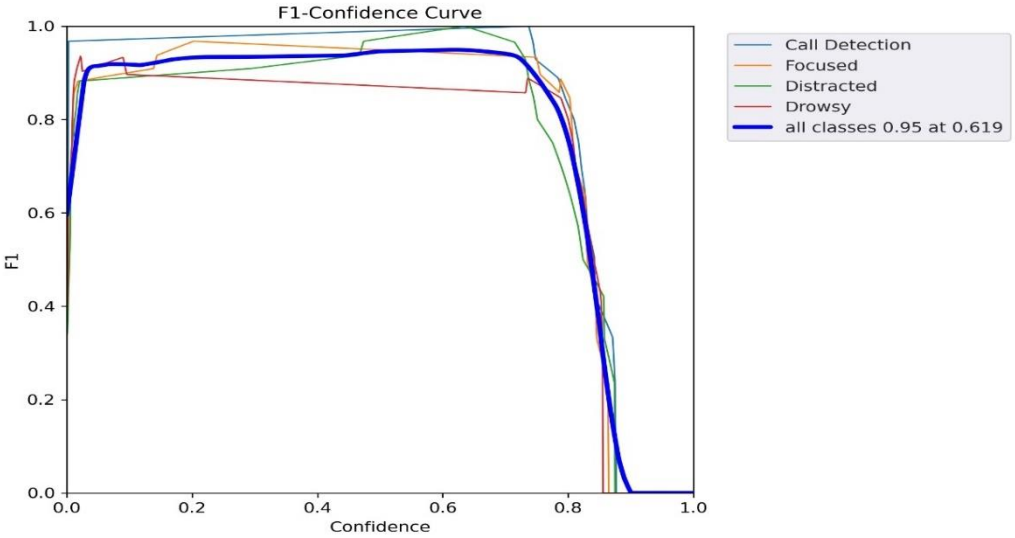



**Fig no 5.9: F1- Confidence Curve**

The Fig no 5.4.9image displays the **F1-Confidence Curve**, illustrating the balance between precision and recall for each class: Call Detection, Focused, Distracted, and Drowsy. The F1-score remains consistently high (above 0.85) across all classes up to around 0.75 confidence.

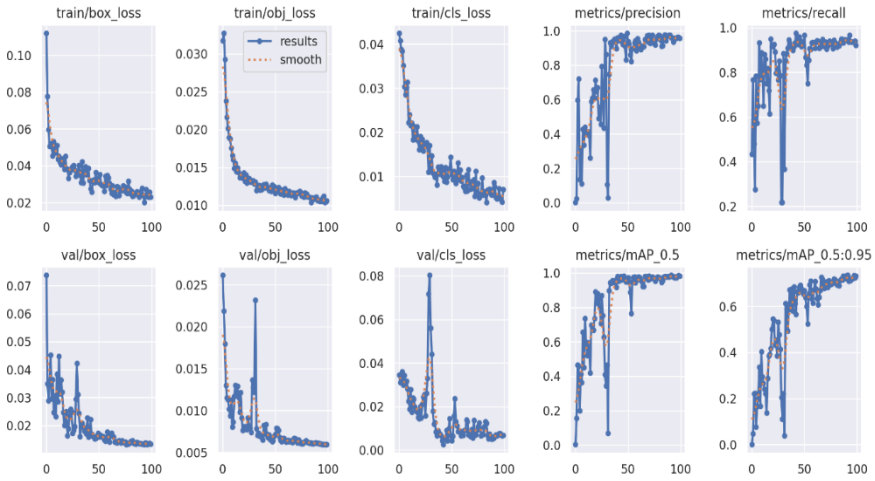



**Fig no 5.10: Training and Validation Curve**

The Fig no 5.10 its show training and validation losses (box, object, and class) consistently decrease, indicating effective model learning. Precision, recall, and MAP metrics improve significantly over epochs, with mAP@0.5 reaching close to 1.0, showing excellent detection performance.

## 5.5 Result Analysis

The performance evaluation of the Detection of driver fatigue System was conducted by analysing the effectiveness of detecting drowsiness, call usage and driver distractions. Performance evaluation focused on accuracy, precision, recall and detection speed to ensure the system operates reliably in real.

**Detection of driver fatigue Performance**

The system employed YOLOv5 to track facial indicators such as how long the eyes remain closed, and head positioning. Experimental results proved to be highly effective in recognizing fatigue in drivers, with an optimal threshold set for 30-second intervals to ensure timely alerts. The implementation of real-time video processing allowed for rapid detection and response, minimizing potential road hazards.

**Automatic Call Detection Analysis**

The automatic call detection future efficiently recognized phone usage during driving. Using image-based analysis, the system identified when the driver was engaged in a call. The results showed that the model accurately distinguished between handheld phone use and other hand movements, reducing false positives. Alerts were successfully triggered, ensuring that drivers remained focused on the road.

**Distraction Alertness Evaluation**

The distraction alertness module was assessed based on its ability to detect head movements, gaze deviations and frequent hand gestures unrelated to driving. The model effectively recognized instances where the driver was looking away from the road or engaged in non-driving activities. The detection system maintained a low false detection rate, ensuring only genuine distractions were flagged, thereby enhancing overall safety.

**Overall System Performance**

The complete system was evaluated in multiple practical driving scenarios., including daytime, nighttime and different lighting scenarios. The model exhibited high adaptability to varying environments, ensuring consistent performance. Furthermore, the real-time alert mechanism provided instant feedback, helping to prevent potential accidents. The continuous monitoring and quick response of the system underline its potential as a significant safety.

# CHAPTER 6

# SYSTEM TESTING

## 6.1 Testing Methodology

Testing is a critical process in software development, assessing a system or its components to determine whether they meet the specified requirements. Simply put, testing involves developing a system to detect discrepancies, issues have not been captured functionalities compared to the expected outcomes. The testing process varies based on the project methodology and the stakeholders involved.

In the IT industry, large organizations typically have dedicated teams responsible for evaluating software against given requirements. Additionally, developers perform Unit Testing, which concentrates on testing each software module separately. Various professionals, such as software testers, developers, project managers, and end users, contribute to the testing process within their respective roles.

A test strategy defines the approach used to ensure software quality throughout the development cycle. It connects high-level organizational goals to specific testing activities, ensuring quality assurance. Key categories of testing consist of unit tests, integration tests and validation procedures.

**Precision**

Measures how many of the positive predictions made by the model are actually correct. *(*High precision = few false positives.)

**Recall**

Measures how many actual positives were correctly identified by the model. (High recall = few false negatives.)

**F1-Score**

The harmonic means of precision and recall, giving a balance between the two.

**MAP@0.5 (mean Average Precision at IOU 0.5)** Shows how accurately the model detects objects when the Intersection over Union (IOU) threshold is 0.5.(MAP closer to 1 = better performance.)

**MAP@0.5:0.95**

More strict evaluation by averaging the MAP across multiple IOU thresholds (from 0.5 to 0.95). (It tests how precise the bounding boxes are.)

**Training Loss**

- **Box Loss** (Localization loss) accuracy of object bounding box prediction.
- **Objectless Loss** how well the model distinguishes between background and objects.
- **Classification Loss** correctness of object classification.

**Validation**

Same types of losses measured on the validation set to check for overfitting or underfitting.

**1. Unit Testing:**

Unit testing concentrates on evaluating separate modules of the software, like modules or functions, to verify their performance. It involves analysing module interfaces, checking input-output flows and validating local data structures to maintain integrity during execution. Boundary conditions and error-handling mechanisms are also tested to ensure robustness.

**2. Integration Testing:**

Integration testing is performed to assess how different software modules interact and work together to form a complete system. It follows a systematic approach to building the software structure by integrating individual Modules are linked systematically to ensure proper interaction. This process typically involves two key strategies for integration testing:

**Simultaneous Component Testing** – Testing all modules together without integrating them step by step.

**Incremental Integration Testing** – Testing individual modules first and progressively combining them to form a fully integrated system.

**3. Functional Testing:**

Various drowsiness scenarios, such as long blinks, continuous Actions such as eye shut, repeated yawns and head movement are recreated to test the system's reaction time. The threshold values for EAR and MAR are fine-tuned based on experimental results to minimize false positives and negatives. The alert mechanism's effectiveness is tested by ensuring the buzzer, seat vibration and emergency notifications are triggered at appropriate times.

**4. Real-Time Testing:**

The system is evaluated under actual road scenarios to evaluate the system's performance with different surrounding conditions like illumination variations camera angles and driver positions. Multiple individuals with different facial features, eyewear and head movements are applied to evaluate how well the model adjusts to different conditions. The response time of the system from drowsiness detection to alert activation is measured to ensure timely intervention.

**5. Performance Evaluation:**

The accuracy of drowsiness detection is assessed by comparing system predictions with manually labelled drowsiness states. Model evaluation is done through essential performance measures, such as precision, recall and the F1-score. Power consumption and latency are analysed to ensure proper and efficient functioning without excessive delays.

**6.Real-time Responsiveness:**

The system's ability to detect and alert drowsiness in real-time is assessed by measuring the time taken between detecting a drowsy state and activating the alert mechanism.

**7.Validation Testing:**

Validation testing is the process of evaluating input information used to verify its accuracy and correctness. It checks whether the provided inputs meet the expected criteria, ensuring that only valid data is processed. When working with databases, validation is crucial to maintain data integrity, which can be achieved through scripting techniques to verify user entries.

Verification and validation are commonly confused, but each plays a unique role in the development process. Verification ensures that the software meets design specifications and is implemented correctly, while validation confirms that the system meets user requirements and functions as intended. Both methods play a critical role in maintaining software quality and reliability.

**8.Verification Testing:**

Verification testing is the process of evaluating the work progress and developing the application processes. confirm that a process meets the requirements as defined in the phase of developing the application processes.

# CHAPTER 7
# CONCLUSION

## 7.1 Conclusion

It detects driver drowsiness marks significant progress in enhancing road safety through continuously observing to detecting indicators the driver's exhaustion. Utilizing real-time video processing and intelligent machine learning models, the system accurately detects warning signs including extended eye closure, frequent and head shifts associated with drowsiness. Equipped with alert mechanisms like buzzer alarms and motor control, it promptly notifies drivers, encouraging necessary breaks to prevent accidents. With its scalable and adaptable design, the system serves as an essential preventive measure against fatigue-related road incidents, offering a modern and life-saving solution for the transportation sector. Through testing under different conditions, driver postures, and facial variations has demonstrated the system's resilience and adaptability. By fine-tuning threshold measurements for The Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR)it minimizes inaccuracies, ensuring precise detection. Additionally, integrating cloud-based analytics enhances data logging and predictive capabilities, making it a practical and a practical approach suitable for real-life applications. The system delivers a budget-friendly, trustworthy and accessible solution to boost road safety. Future developments could incorporate advanced AI-driven models, seamless integration with vehicle control mechanisms and real-time connectivity with emergency response systems to further enhance its effectiveness. Continuous refinements in detection algorithms and hardware efficiency will strengthen its potential to prevent a budget-friendly, trustworthy and accessible solution to boost road safety.

## 7.2 Future Scope

The Driver Fatigue Detection System holds vast potential for future improvements and expanded applications. In the coming years, the integration of more advanced Neural network models could significantly improve the precision of fatigue and distraction Identification, reducing false positives and adapting to different driver behaviours. Emerging intelligent technologies like machine learning and neural networks frameworks pave the way for smarter drowsiness detection systems that personalize alerts based on individual driving styles, increasing overall accuracy**.** The incorporation of edge computing will enable faster processing, reducing the dependency on cloud servers and ensuring real-time responsiveness. Additionally, integration with vehicle automation systems can allow automatic speed reduction or braking when drowsiness is detected. Future iterations may also include multi-sensor fusion, combining heart rate monitoring, steering pattern analysis, and voice recognition to enable a more thorough evaluation of driver fatigue. Furthermore, wireless connectivity with fleet management systems will enable large-scale deployment, assisting in commercial vehicle safety. These advancements will make the system more intelligent, reliable, and widely applicable, significantly Enhancing road safety and reducing accident risks through Driver Fatigue Detection Systems will evolve with enhanced biometric analysis, including eye-tracking sensors, pulse oximeters, and electroencephalogram (EEG) signals for highly accurate fatigue detection. AI-powered adaptive learning models will refine the way the system can effectively recognize unique driving patterns and identify drowsiness with minimal false alarms.

# REFERENCES

[1] Ahmet Kolus(2024) “A Systematic Review on Driver Drowsiness Detection Using Eye Activity Measures”, *IEEE Access, vol. 12, pp.* 97969 to 97993.

[2] Albadawi, Y., AlRedhaei, A., & Takruri, M. (2023). "Real-Time Machine Learning-Based Driver Drowsiness Detection Using Visual Features," *J. Imaging*, vol. 9, no. 5, article 91.

[3] Bedi, P., & Singh, D. (2019) “Drowsiness Detection System Using YOLO-Based Real-Time Object Detection and Deep Learning”, *Journal of Intelligent Systems,* pp.605 to 616

[4] Bhardwaj, P., & Saha, S. (2018) “A Study on Real-Time Driver Drowsiness Detection System Using Eye and Physiological Signals”, *Journal of Advanced Transportation*, pp.1 to 11.

[5] Das, S., Pratihar, S., Pradhan, B., Jhaveri, R. H., & Benedetto, F. (2024). "IoT-Assisted Automatic Driver Drowsiness Detection through Facial Movement Analysis Using Deep Learning and a U-Net-Based Architecture," *Information*, vol. 15, no. 1, article 30.

[6] Geng, X., & Zhang, Z. (2021) “Driver Drowsiness Detection Based on Deep Learning and Facial Feature Extraction”, *Sensors*.

[7] Gupta, M., & Sinha, S. (2021) “A Systematic Review of Physiological Signals-Based Driver Drowsiness Detection Systems, *Sensors” pp.* 1229 to 1259.

[8] Gupta, R., & Thakur, M. (2021) “Real-Time Drowsiness Detection: A Machine Learning Approach for Improved Road Safety”, *Proceedings of the International Conference on Machine Learning,pp.605 to 616*.

[9] Gül, O., & Yılmaz, R. (2021) “A Review of Driver Drowsiness Detection Systems: Techniques and Future Directions”, *International Journal of Human-Computer Interaction*.

[10] Inkeaw, P., Wongsawat, Y., & Wongsawat, Y. (2022). "Automatic Driver Drowsiness Detection Using Artificial Neural Network Based on Visual Facial Descriptors: Pilot Study," *Frontiers in Public Health*, vol. 10, article 9482962.

[11] Jose, J., J, A., Raimond, K., & Vincent, S. (2022). "SleepyWheels: An Ensemble Model for Drowsiness Detection Leading to Accident Prevention," *arXiv preprint arXiv:2211.00718*.

[12] Krishnan, R., & Manikandan, S. (2020) “Driver Drowsiness Detection Using EEG and Eye Tracking Systems”, *Journal of Electrical Engineering & Technology*, pp.15312 to 15323.

[13] Madni, H. A., Raza, A., Sehar, R., Thalji, N., & Abualigah, L. (2024). "Novel Transfer Learning Approach for Driver Drowsiness Detection Using Eye Movement Behavior," *IEEE Access*, vol. 12, pp. 64765–64778.

[14] Maheswari, V. U., Aluvalu, R., Kantipudi, M. V. V. P., Chennam, K. K., Kotecha, K., & Saini, J. R. (2022). "Driver Drowsiness Prediction Based on Multiple Aspects Using Image Processing Techniques," *IEEE Access*, vol. 10, pp. 54980–54990.

[15] Sharma, S., & Kumar, S. (2021) “A Survey on State-of-the-Art Drowsiness Detection Techniques. *Journal of Intelligent Transportation Systems” pp. 61904 to 61919.*

[16] Sengar, S. S., Kumar, A., & Singh, O. (2024). "VigilEye—Artificial Intelligence-based Real-time Driver Drowsiness Detection," *arXiv preprint arXiv:2406.15646*.

[17] Singh, R., & Jain, S. (2021) “A Review of Recent Developments in Driver Drowsiness Detection Systems”, *International Journal of Automotive Technology.*

[18] Venkata Sai Vardhan, V., Ritish Kumar Reddy, N., Jaya Surya, K., Uday Kiran, J., & Kumar, A. (2021). "Driver's Drowsiness Detection Based on Facial Multi-Feature Fusion," *Journal of Physics: Conference Series*, vol. 1998, article 012034

[19] Yogarajan, G., Singh, R. N., Nandhu, S. A., & Rudhran, R. M. (2024). "Drowsiness Detection System Using Deep Learning Based Data Fusion Approach," *Multimedia Tools and Applications*, vol. 83, pp. 36081–36095.

# PAPER PUBLICATION

# Conference certificates

KSR College of Engineering

NAAC A++

ICITSM-25

EAI PUBLISHING

Scopus

NTLTECHNOLOGY

4TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY, CIVIL INNOVATION, SCIENCE, AND MANAGEMENT (ICITSM'25)

CERTIFICATE

of Presentation on 28 & 29 April 2025

This Certificate is Proudly Presented To

P. KAVYA

A VISION-BASED DRIVER DROWSINESS DETECTION FRAMEWORK FOR ROAD SAFETY ENHANCEMENT

in Conference of Proceedings of the 4th International Conference on Information Technology, Civil Innovation, Science, and Management (ICITSM'25) Organized by Department of Civil Engineering at K.S.R. College of Engineering, Namakkal, Tamil Nadu, India

Dr S Senthilkumar,
Professor and Head,
Department of Civil Engineering,
K. S. R. College of Engineering,
Tiruchengode

ICITSM CONFERENCE PROCEEDINGS - 2025

Ms.Gugapriyaa Sivakumar
Industry Partner,
NTL Technology, Erode.

KSR College of Engineering

A++ NAAC

ICITSM-25

EAI PUBLISHING

Scopus

NTL TECHNOLOGY

4TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY, CIVIL INNOVATION, SCIENCE, AND MANAGEMENT (ICITSM'25)

CERTIFICATE

of Presentation on 28 & 29 April 2025

This Certificate is Proudly Presented To

U. LAVANYA

A VISION-BASED DRIVER DROWSINESS DETECTION FRAMEWORK FOR ROAD SAFETY ENHANCEMENT

in Conference of Proceedings of the 4th International Conference on Information Technology, Civil Innovation, Science, and Management (ICITSM'25) Organized by Department of Civil Engineering at K.S.R. College of Engineering, Namakkal, Tamil Nadu, India

Dr S Senthilkumar,
Professor and Head,
Department of Civil Engineering,
K. S. R. College of Engineering,
Tiruchengode

ICITSM CONFERENCE PROCEEDINGS -2025

Ms.Gugapriyaa Sivakumar
Industry Partner,
NTL Technology, Erode.

KSR College of Engineering

A++ NAAC

ICITSM-25

EAI PUBLISHING

Scopus

NTLTECHNOLOGY

4TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY, CIVIL INNOVATION, SCIENCE, AND MANAGEMENT (ICITSM'25)

CERTIFICATE

of Presentation on 28 & 29 April 2025

This Certificate is Proudly Presented To

N. RAJESH

A VISION-BASED DRIVER DROWSINESS DETECTION FRAMEWORK FOR ROAD SAFETY ENHANCEMENT

in Conference of Proceedings of the 4th International Conference on Information Technology, Civil Innovation, Science, and Management (ICITSM'25) Organized by Department of Civil Engineering at K.S.R. College of Engineering, Namakkal, Tamil Nadu, India

Dr S Senthilkumar,
Professor and Head,
Department of Civil Engineering,
K. S. R. College of Engineering,
Tiruchengode

CONFERENCE PROCEEDINGS - 2025 ICITSM

Ms.Gugapriyaa Sivakumar
Industry Partner,
NTL Technology, Erode.

KSR College of Engineering

A++ NAAC

ICITSM-25

EAI PUBLISHING

Scopus

NTLTECHNOLOGY

4TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY, CIVIL INNOVATION, SCIENCE, AND MANAGEMENT (ICITSM'25)

CERTIFICATE

of Presentation on 28 & 29 April 2025

This Certificate is Proudly Presented To

M. PRASHANTH

A VISION-BASED DRIVER DROWSINESS DETECTION FRAMEWORK FOR ROAD SAFETY ENHANCEMENT

in Conference of Proceedings of the 4th International Conference on Information Technology, Civil Innovation, Science, and Management (ICITSM'25) Organized by Department of Civil Engineering at K.S.R. College of Engineering, Namakkal, Tamil Nadu, India

Dr S Senthilkumar,
Professor and Head,
Department of Civil Engineering,
K. S. R. College of Engineering,
Tiruchengode

ICITSM CONFERENCE PROCEEDINGS -2025

Ms.Gugapriyaa Sivakumar
Industry Partner,
NTL Technology, Erode.

## Conference Submitted Paper

# A Vision-Based Driver Drowsiness Detection Framework For Road Safety Enhancement

*P. Jayaselvi*
*Assistant Professor, Computer Science and Technology*
*Madanapalle Institute of Technology & Sciences, Madanapalle, India*
*Email: jayaselvip@mits.ac.in*

*P. Kavya*
*UG Scholor , Computer Science and Technology Madanapalle Institute of Technology & Sciences, Madanapalle, India*
*Email : 21691A2869@mits.ac.in*

*U. Lavanya*
*UG Scholor ,Computer Science and Technology*
*Madanapalle Institute of Technology & Sciences, Madanapalle, India*
*Email:21691A2873@mits.ac.in*

*N. Rajesh*
*UG Scholor , Computer Science and Technology Madanapalle Institute of Technology & Sciences, Madanapalle, India*
*Email:21691A28D2@mits.ac.in*

*M. Prashanth*
*UG Scholor , Computer Science and Technology*
*Madanapalle Institute of Technology & Sciences, Madanapalle, India*
*Email : 22695A2812@mits.ac.in*

***Abstract***— Driver drowsiness and diversion are major causes of street mishaps, posturing noteworthy dangers to street security. It points to create an progressed driver observing framework that leverages YOLOv5 (You Simply See Once) protest location to distinguish early signs of tiredness and diversion. The framework employments a custom dataset to examine key facial highlights including actions like blinking, tilting the head, to identify tiredness. Furthermore, it joins three basic highlights: programmed call discovery, diversion readiness, yawning discovery. The programmed call location include distinguishes when the driver is locked in in a call, guaranteeing that diversions are minimized. The diversion sharpness highlight triggers an alarm when anomalous head development or position changes are recognized, showing conceivable weariness or diversion. The yawning location include screens mouth developments to distinguish intemperate yawning, a key marker of drowsiness

**KEYWORDS:** *Driver Monitoring, Drowsiness Detection, YOLOv5, Real-Time Alert System, Distraction Detection, Road Safety.*

## I.INTRODUCTION

Driver weariness and diversion are among the driving causes of street mishaps, posturing critical dangers to street security. To address this issue, our venture presents a vision- based driver drowsiness discovery system that utilizes profound learning methods and computer vision [1].

The proposed framework coordinating YOLOv5 (You Merely See Once) for protest discovery and facial investigation, identifying signs of drowsiness through key pointers such as eye closure, head tilting, and yawning. This framework presents inventive highlights, counting programmed call discovery, diversion readiness, and seatbelt location, guaranteeing a comprehensive approach to mischance avoidance [2]. By leveraging machine learning calculations like CNN, LSTM, and SSD, the framework precisely recognizes fatigue-related behaviors and cautions the driver in genuine time [3]. The system too consolidates edge computing for low-latency handling, making it a commonsense arrangement for use cases in real-life situation. The oddity of this inquire about lies in its multi-feature integration, flexibility to different driving conditions, and non-intrusive checking approach [4]. By deploying this innovation in vehicles, able to essentially diminish street mischances caused by tiredness and move forward generally activity security. This extend points to bridge the hole between existing discovery strategies and real-time viable execution for improved street security [5].

## II.LITERATURE SURVEY

Ahmet Kolus, *"A Systematic Review on Driver Drowsiness Detection Using Eye Activity Measures,"* IEEE Access, vol. 12, pp. 97969–97993, 2020. This paper provides a

comprehensive review of methods designed to track and detect fatigue-related behavior in drivers eye-related behaviors. [6]

A. H. Al-Bayati, A. S. Shakoree, and Z. A. Ramadan, *"Factors Affecting Traffic Accidents Density on Selected Multilane Rural Highways,"* Civil Engineering Journal, vol. 7, no. 7, pp. 1183–1202, Jul. 2021. This study analyzes the causes behind traffic accident densities on multilane rural highways. Using statistical methods and real-world traffic data., it identifies drowsy driving, high speeds, poor lighting, and lane discipline violations as key contributing factors.[7]

B. C. Tefft, *"Prevalence of Motor Vehicle Crashes Involving Drowsy Drivers, United States, 2009–2013,"* AAA Foundation for Traffic Safety, Washington, DC, USA, 2014. This report examines the prevalence of crashes linked to driver fatigue across the U.S. over a five-year span. It provides statistical evidence showing that drowsy driving is significantly underreported in official crash records.[8]

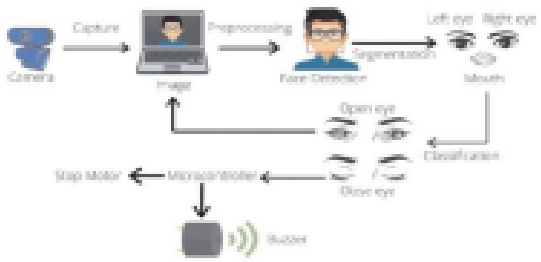
O. Gül and R. Yılmaz, *"A Review of Driver Drowsiness Detection Systems: Techniques and Future Directions,"* International Journal of Human-Computer Interaction, 2021. This review offers a broad examination of driver drowsiness detection methods, categorizing them into behavioural, physiological, and vehicular-based approaches.[9]

P. Bhardwaj and S. Saha, *"A Study on Real-Time Driver Drowsiness Detection System Using Eye and Physiological Signals,"* Journal of Advanced Transportation, pp. 1–11, 2018. This study explores designing a real-time solution for identifying drowsy driving behavior and that utilizes both eye-based features and physiological signals.[10]

R. Singh and S. Jain, *"A Review of Recent Developments in Driver Drowsiness Detection Systems,"* International Journal of Automotive Technology, 2021. This paper reviews recent advancements in the field related to monitoring driver alertness, with special attention on modern machine learning and computer vision-based approaches.[11]

S. Sharma and S. Kumar, *"A Survey on State-of-the-Art* the growing role of deep learning and computer vision in enabling non-intrusive, camera-based detection systems. It concludes by identifying key challenges such as adaptability to different lighting conditions, individual variability among drivers, and the need for multimodal integration to improve reliability in real-world applications.[12]

## III.PROPOSED METHODOLOGY




In the proposed driver monitoring system, a camera continuously captures real-time video of the driver, which is processed to detect signs of drowsiness or distraction. YOLOv5, a fast and accurate object detection algorithm, is employed to identify and track critical facial features such as the eyes and mouth within each frame. Once the face is detected, the algorithm segments the left eye, right eye, and mouth regions for further analysis. These features are classified to determine if the eyes are open or closed and to detect yawning behavior. If the system identifies prolonged eye closure or frequent yawning, it interprets these as signs of fatigue. A microcontroller is then triggered to activate a buzzer alert and can optionally send signals to stop the vehicle's motor, ensuring immediate driver awareness and enhanced road safety.

The include extraction and demonstrate development stage utilizes YOLO (You Simply See Once) V5 for real-time question location, centering on facial highlights, head development, and hand motions. Within the real-time checking and alarm framework, a camera setup will ceaselessly track the driver's confront and behavior. Alarms will be activated when signs of tiredness, diversion, or phone utilization are recognized. These incorporate:

The following steps outline the implementation of the proposed driver drowsiness detection system:

**Step 1: Real-Time Video Capture**

A webcam or onboard camera is used to continuously capture live video footage of the driver's face while the vehicle is in motion.

**Step 2: Image Preprocessing**

The captured video frames are sent to a processing unit (e.g., laptop or embedded system), where they undergo preprocessing operations such as resizing, grayscale conversion, and noise reduction to enhance feature visibility and computational efficiency.

**Step 3: Face Detection**

Using facial detection algorithms, the system identifies and extracts the driver's face from each frame. This ensures the focus remains on relevant regions for further analysis.

**Step 4: Facial Feature Segmentation**

The system isolates key facial components, specifically the left eye, right eye, and mouth, to examine signs of fatigue or drowsiness.

**Step 5: State Classification**

The eyes are classified as either open or closed based on the shape, pixel intensity, or trained models.

The mouth is monitored to identify potential yawning behavior based on its openness over time.

**Step 6: Drowsiness Evaluation**

If the eyes remain closed beyond a certain threshold or if the mouth stays open in a yawning pattern for a few seconds, the system flags this as a sign of reduced alertness.

**Step 7: Alert and Response Mechanism**

When drowsiness is detected, a signal is sent to a microcontroller, which triggers:

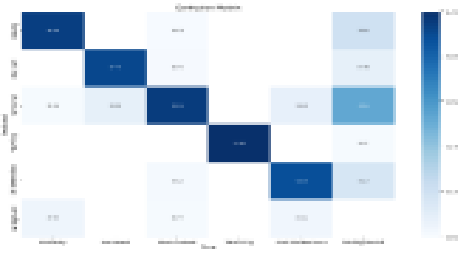
An audio buzzer to immediately alert the driver.

Optionally, a motor stop signal that can halt the vehicle or initiate additional safety protocols.

**Step 8: Continuous Monitoring**

The system operates continuously, analyzing each frame in real time to ensure the driver remains attentive and responsive.

## IV.PERFORMANCE METRIX



To determine the effectiveness of the proposed driver drowsiness detection system, a set of important performance metrics will be utilized. These indicators will evaluate how precisely, consistently, and efficiently the system functions in real-time scenarios.

**Precision**– This represents the system's ability to correctly identify conditions such as driver drowsiness, distractions, and mobile phone usage, reflecting its overall detection capability.

**Formula:**

**Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)**

**Precision** – This indicates how many of the detected drowsy instances were actually correct. A higher precision value suggests that the system generates fewer incorrect drowsiness alerts.

**Formula:**

**Precision = (True Positives) / (True Positives + False Positives)**

**Recall (Sensitivity)** – This metric reflects how well the system identifies true drowsiness by comparing the number of correctly detected drowsy cases to the total number of actual drowsy instances, highlighting the model's effectiveness in recognizing genuine fatigue conditions.

**Formula:**

**Recall = (True Positives) / (True Positives + False Negatives)**

**F1 Score** – A balanced metric that combines both precision and recall to provide a comprehensive evaluation of the system's performance.

**Formula:**

**F1 Score = 2 × (Precision × Recall) / (Precision + Recall)**

**Latency** – This measures the system's response time for detecting drowsiness and generating alerts. Lower latency ensures a quicker response to driver fatigue and distractions.

By analyzing these metrics, the system's reliability and effectiveness in improving road safety can be validated

## V.DESIGN & IMPLEMENTATION

The design of the driver drowsiness detection system focuses on real-time monitoring of the driver's facial behavior to identify signs of fatigue and distraction. A camera installed on the vehicle's dashboard captures a continuous stream of video, which is then processed frame by frame. These frames undergo basic preprocessing steps like resizing and normalization to enhance the clarity and consistency of the input data. The core of the detection process utilizes the YOLOv5 deep learning model, which accurately identifies facial features such as the eyes, mouth, and the presence of a mobile phone near the ears.

Once the face is detected, the system uses Dlib and Mediapipe to extract facial landmarks for the eyes and mouth. Eye closure is monitored by evaluating the eye aspect ratio; if the driver's eyes remain closed for more than four seconds, the system triggers an audio alert, indicating drowsiness. Similarly, yawning is detected by measuring how long the mouth stays open. If the mouth remains open beyond four seconds, a distinct alert is activated to warn the driver. To detect distractions caused by mobile phone use, YOLOv5 tracks hand positions near the ears. If the driver

is holding a phone on either side for over ten seconds, the system raises a warning for distracted driving.

The system also checks for driver inattention by analyzing gaze direction and head pose. If the driver looks away from the road (left or right) for more than four seconds, it classifies the behavior as a distraction and initiates an alarm. When any of these conditions are met, a microcontroller is triggered to activate a buzzer and can optionally interface with the vehicle's control system to reduce speed or stop. Each alert is documented in a log file with timestamps for future review, and the frame triggering the alert is saved as an image. Additionally, a graphical user interface (GUI) provides real-time visuals of the camera feed, displays alert notifications, and summarizes the total number of each type of alert detected during the session. This integrated design ensures accurate, real-time detection and response to potentially dangerous driving behavior, thereby enhancing on-road safety.

**Drowsiness Caution** – Activated when delayed eye closure or over the top yawning is recognized.

**Diversion Alarm** – Actuated when irregular head developments show inattentiveness.

## VI.RESULT

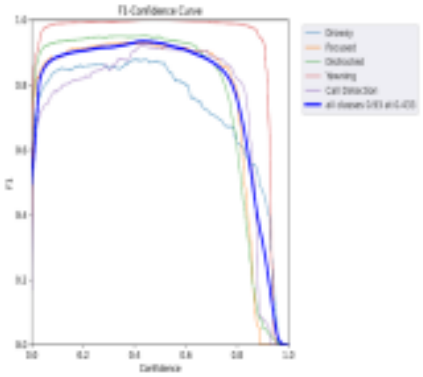



Fig 2 : F1-Confidence Curve

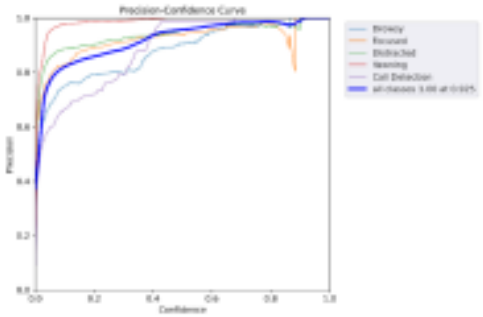



Fig 3: Precision Curve

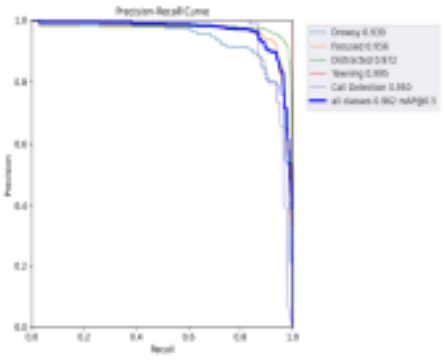



Fig 4 : Precision –Recall Curve

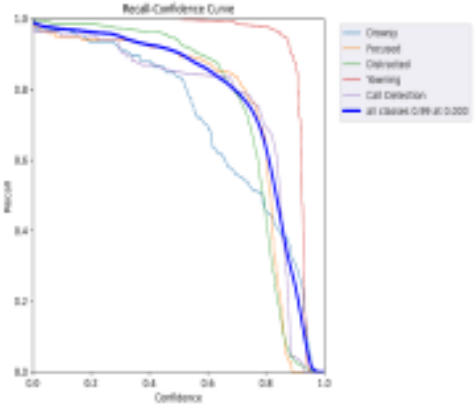



Fig 5 : Recall Curve

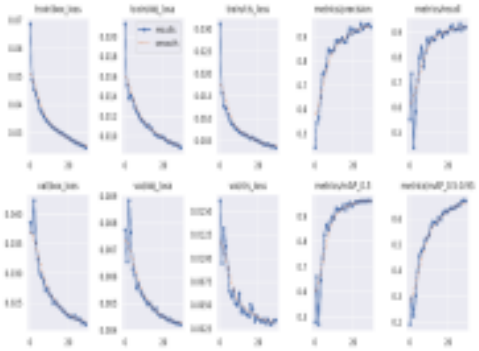
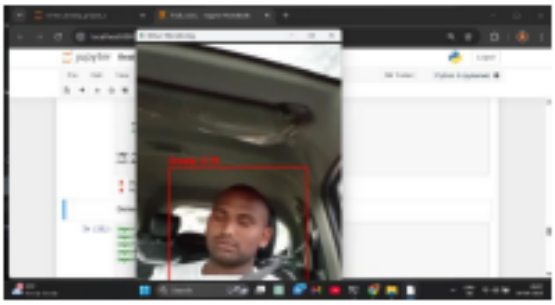
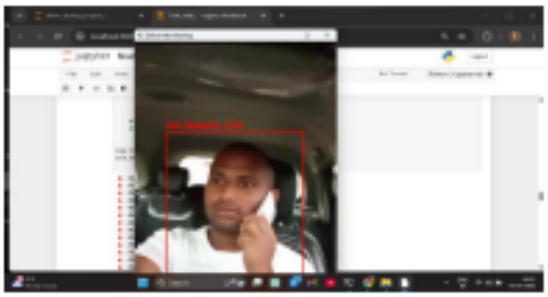
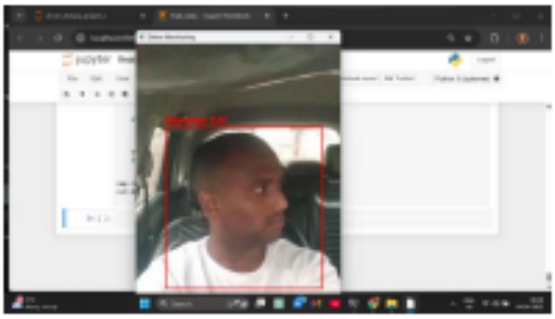


Fig 6 : Training and validation curve







## VII.CONCLUSION:

The Driver Drowsiness Detection project has been successfully developed with the primary goal of enhancing road safety by identifying signs of fatigue and inattention in real-time. The system uses computer vision techniques to monitor the driver's facial expressions and eye movements, allowing it to detect drowsiness with a high level of accuracy. To strengthen the functionality and make the system more comprehensive, additional features such as automatic call detection, distraction alertness, and yawning detection have been integrated. The automatic call detection feature identifies when the driver is on a phone call during driving, which is a major cause of accidents, and immediately alerts the driver to focus on the road. The distraction alertness module keeps track of the driver's head movements and gaze direction to determine if their attention is diverted from the road, triggering warnings when needed. Yawning detection serves as an early indicator of fatigue, as frequent yawning usually precedes eye closure or microsleeps. By combining these features, the system not only focuses on detecting drowsiness but also addresses multiple unsafe driving behaviors. Real-time alerts help the driver take immediate corrective actions, significantly reducing the chances of accidents due to fatigue or distraction. This project showcases a smart and effective driver assistance system that, with further development, can be integrated into modern vehicles to promote safer driving practices and prevent road mishaps

## VIII.REFERENCES:

[1]. Ahmet Kolus, A. "A Systematic Review on Driver Drowsiness Detection Using Eye Activity Measures," IEEE Access, vol. 12,pp. 97969 to 97993, 2020

[2]. Sharma, S., & Kumar, S. "A Survey on State-of-the-Art Drowsiness Detection Techniques. Journal of Intelligent Transportation Systems"pp. 61904 to 61919,2021

[3]. Gupta, M., & Sinha, S. "A Systematic Review of Physiological Signals-Based Driver Drowsiness Detection Systems, Sensors", pp. 1229 to 1259,2021.

[4] Singh, R., & Jain, S. "A Review of Recent Developments in Driver Drowsiness Detection Systems", International Journal of Automotive Technology,2021.

[5] Gül, O., & Yılmaz, R. "A Review of Driver Drowsiness Detection Systems: Techniques and Future Directions", International Journal of Human-Computer Interaction,2021.

[6]. Bhardwaj, P., & Saha, S. "A Study on Real-Time Driver Drowsiness Detection System Using Eye and Physiological Signals", Journal of Advanced Transportation, pp.1 to 11,2018

[7] Krishnan, R., & Manikandan, S. "Driver Drowsiness Detection Using EEG and Eye Tracking Systems", Journal of Electrical Engineering & Technology, pp.15312 to 15323,2020.

[8] Gupta, R., & Thakur, M. "Real-Time Drowsiness Detection: A Machine Learning Approach for Improved Road Safety", Proceedings of the International Conference on Machine Learning, pp.605 to 616, 2021.

[9] Bedi, P., & Singh, D. "Drowsiness Detection System Using YOLO-Based Real-Time Object Detection and Deep Learning", Journal of Intelligent Systems, pp.605 to

616,2019

[10] Geng, X., & Zhang, Z. "Driver Drowsiness Detection Based on Deep Learning and Facial Feature Extraction", Sensors,2021

[11] Global Status Report on Road Safety 2018, World Health Org., Geneva, Switzerland, 2018

[12] S. Chen, Z. Wang, and W. Chen, 'Driver drowsiness estimation based on factorized bilinear feature fusion and a long-short-term recurrent convolutional network,' Information, vol. 12, no. 1, p. 3, Dec. 2020.

[13] A. H. Al-Bayati, A. S. Shakoree, and Z. A. Ramadan, "Factors affecting traffic accidents density on selected multilane rural highways," Civil Eng. J., vol. 7, no. 7, pp. 1183–1202, Jul. 2021.

[14] B. C. Tefft, 'Prevalence of motor vehicle crashes involving drowsy drivers, United States, 2009–2013,' AAA Found. Traffic Saf., Washington, DC, USA, 2014, pp. 1–10.

[15] P. M. Salmon, G. J. M. Read, V. Beanland, J. Thompson, A. J. Filtness, A. Hulme, R. McClure, and I. Johnston, 'Bad behaviour or societal failure? Perceptions of the factors contributing to drivers' engagement in the fatal five driving behaviours,' Appl. Ergonom., vol. 74, pp. 162–171, Jan. 2019.