A PROJECT REPORT

ON

"Real-Time Driver Fatigue State Detection and Alert System Using IoT and Deep Learning"

Submitted to

SAVITRIBAI PHULE PUNE UNIVERSITY

in fulfilment of the requirements

for the award of

BACHELOR OF ENGINEERING

(INFORMATION TECHNOLOGY)

BY

MR. Ashish Patil 72213411J

MR. Vedant Patel 72213408J

MR. Saurabh Amrutkar 72213293L

MR. Laxmikant Ghodake 72213348M

MR. Praveen Kumar 72213385F

UNDER THE GUIDANCE OF

Prof. Prasad A. Lahare

DEPARTMENT OF INFORMATION TECHNOLOGY

PVG'S COLLEGE OF ENGINEERING AND SHRIKRUSHNA S.DHAMANKAR INSTITUTE OF MANAGEMENT

206, DINDORI ROAD, MERI, MHASRUL, NASHIK-422004

2024-25

Certificate

This is to certify that the project entitled

"Real-Time Driver Fatigue State Detection and Alert System Using IoT and Deep Learning"

Submitted by

Mr. Ashish Patil | PRN : 72213411J
Mr. Vedant Patel | PRN : 72213408J
Mr. Saurabh Amrutkar | PRN : 72213293L
Mr. Laxmikant Ghodake | PRN : 72213348M
Mr. Praveen Kumar | PRN : 72213385F

are the bonafide students of this institute and the work has been carried out by them under **Prof. P. A. Lahare** and it is approved for the partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**.

Prof. P. A. Lahare
Project Guide

Prof. S. N. Bhadane
Head of Department, IT

_____
External Examiner

Dr. M. V. Bhalerao
Principal

Place : Nashik
Date : _____

ACKNOWLEDGEMENT

We are profoundly grateful to **Prof. P.A.Lahare** for his expert guidance and continuous encouragement throughout the project from its commencement till the completion.

We would like to express deepest appreciation towards **Prof. S.N.Bhadane**, Head of Department of Information Technology and **Dr. M.V.Bhalerao**, Principal Pune Vidyarthi Griha's College of Engineering, Nashik whoose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Information Technology Department who helped us directly or indirectly during this course of work.

Mr.Ashish Patil
Mr.Vedant Patel
Mr.Saurabh Amrutkar
Mr.Laxmikant Ghodake
Mr.Praveen Kumar

ABSTRACT

Driver fatigue is a significant contributor to road accidents, especially for long-haul drivers. This report presents a real-time driver fatigue detection and alert system using IoT and deep learning to enhance road safety. The system utilizes a camera to monitor the driver's facial expressions, eye movements, and head posture, which are processed by a Convolutional Neural Network (CNN) model to detect signs of drowsiness. Key indicators such as eyelid closure, yawning, and head tilt are continuously analyzed. Upon detecting fatigue, the system triggers a series of alerts, including audio warnings, flashing lights, and water sprinklers to re-engage the driver. The vehicle's parking lights may also be activated to alert surrounding traffic. This system aims to reduce accidents caused by driver fatigue by offering a non-intrusive, real-time solution. Its lightweight design ensures minimal distraction, and the use of deep learning enhances detection accuracy. With further advancements, the system could become even more effective in preventing fatigue-related accidents.

Keywords : Driver fatigue detection, Convolutional Neural Networks, CNN, Deep learning, Real-time monitoring, Alert system, IoT

INDEX

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Driver fatigue is a significant factor contributing to road accidents worldwide, especially during long driving periods or under monotonous conditions. To address this issue, modern advancements in technology, particularly in the fields of deep learning and IoT, have enabled the development of systems that can detect fatigue in real time. This project focuses on creating a Fatigue Detection System that utilizes Convolutional Neural Networks (CNN) in combination with Long Short-Term Memory (LSTM) networks to monitor and analyze the driver's facial features, such as eye closure and yawning, through a live camera feed. The system detects early signs of drowsiness and, upon detection, activates IoT devices like alarms, water sprinklers, or parking lights to alert the driver and ensure safety. The hardware used for the system includes a camera and a Raspberry Pi/Arduino, with Python-based frameworks such as TensorFlow and OpenCV handling the model and image processing. This cost-effective and real-time system aims to improve road safety by reducing the risk of accidents caused by driver fatigue, offering potential applications in various types of vehicles, from personal cars to commercial trucks.

1.1 Motivation

Road accidents caused by driver fatigue are a growing concern globally, with long hours of driving, particularly on highways or monotonous routes, increasing the risk of drowsiness behind the wheel. Fatigued drivers often experience a significant reduction in their reaction times, decision-making abilities, and overall attentiveness, leading to dangerous situations. Traditional methods of combating driver fatigue, such as manual rest breaks or in-car alarms, are often insufficient or easily ignored.

Advances in artificial intelligence and deep learning, along with the growing avail-

ability of IoT devices, present an opportunity to tackle this problem through automated monitoring and alert systems. By leveraging machine learning techniques and image processing, a real-time solution can be developed that continuously monitors drivers for signs of fatigue, such as eye closure or yawning, without the need for invasive or expensive sensors. This project is motivated by the potential to create a cost-effective, easy-to-deploy system that can significantly improve road safety, particularly for long-distance and commercial drivers. Reducing accidents due to fatigue not only saves lives but also has broad economic benefits by lowering healthcare and repair costs associated with such incidents.

1.2 PROBLEM DEFINITION

"The increasing number of road accidents due to driver fatigue is a major concern, especially for long-haul and commercial drivers who spend extended hours on the road. Fatigue impairs reaction time, decision-making, and overall driving performance, leading to life-threatening accidents. Traditional fatigue detection methods, such as self-reporting, wearable devices, or vehicle-based metrics, have limitations in accuracy, real-time processing, and ease of implementation. Existing systems may rely on steering patterns, heart rate monitoring, or blink rate sensors, but they often lack robustness and fail to provide immediate alerts. This project introduces a Real-time Driver Fatigue Detection System using Deep Learning and IoT to analyze facial expressions, eye movements, and head position through computer vision techniques. The system utilizes a camera-based approach with machine learning models trained to detect drowsiness signs. When fatigue is detected, it activates alerts such as an alarm, water sprinklers, and flashing parking lights to wake up the driver and prevent accidents."

Chapter 2

LITERATURE SURVEY

2.1 Literature Review

2.1.1 K. Li, Y. Gong, Z. Ren (2020)

In the paper titled "A Fatigue Driving Detection Algorithm Based on Facial Multi-Feature Fusion" (2020), K. Li, Y. Gong, and Z. Ren propose a deep learning system that detects driver fatigue by fusing multiple facial features[1]. The system monitors key indicators such as eye movements, yawning frequency, and blinking patterns. It employs the YOLO (You Only Look Once) algorithm for real-time object detection to identify facial landmarks. To improve accuracy, the system incorporates personalized driver models that account for individual differences in facial behavior, enhancing its robustness across diverse drivers. By combining multiple features, the system provides a more comprehensive and reliable fatigue detection method. The paper reports high accuracy in detecting driver drowsiness, showcasing significant advancements in using deep learning for facial analysis in fatigue detection. This contribution is crucial for developing effective real-time driver monitoring systems to prevent fatigue-related accidents.

2.1.2 Z. Kehua, J. Wang (2020)

In the 2020 paper titled "Driver Fatigue Detection Method Based on Eye States with Pupil and Iris Segmentation," Z. Kehua and J. Wang introduce a fatigue detection method focused on eye tracking using pupil and iris segmentation[2]. The approach employs a U-Net architecture, which enhances the precision of eye-openness estimation by segmenting the eye's pupil and iris regions in real time. This method surpasses earlier fatigue detection techniques by improving both accuracy and processing speed,

making it suitable for real-time applications. The system tracks critical fatigue indicators like eye closure duration, providing a reliable means of detecting drowsiness. Validated on a large dataset, this method achieves high accuracy in identifying early signs of driver fatigue. Its combination of efficiency and precision makes it a notable advancement in fatigue detection through eye-state monitoring, contributing significantly to road safety solutions.

2.1.3 R. Alharbey, M. Dessouky (2022)

In the 2022 study "Fatigue State Detection for Tired Persons in Presence of Driving Periods," R. Alharbey and M. Dessouky focus on detecting driver fatigue by integrating physiological signals with visual data. The research employs EEG signals in conjunction with Support Vector Machine (SVM) and deep learning techniques, specifically Convolutional Neural Networks (CNN), to achieve a robust fatigue detection system[3]. This fusion of data modalities results in an impressive detection accuracy of up to 98 percent, significantly enhancing the reliability of fatigue monitoring. By providing timely alerts in real-time, the system ensures that drivers receive immediate feedback on their fatigue state. This comprehensive approach not only improves driver safety but also contributes to the development of advanced monitoring systems tailored for fatigue detection in various driving conditions.

2.1.4 H. Yaacob, F. Hossain, S. Khare (2023)

In the 2023 study "Application of Artificial Intelligence Techniques for Brain–Computer Interface in Mental Fatigue Detection" by H. Yaacob, F. Hossain, and S. Khare, the focus is on using EEG-based systems to detect mental fatigue in drivers[4]. The research demonstrates the effectiveness of AI in analyzing brainwave data to monitor alertness levels. While the study reports high accuracy in detecting fatigue, it identifies gaps in fully automated neurofeedback systems. The authors suggest that hybrid models, which combine EEG data with other physiological signals, could yield more precise results in real-world applications. This approach aims to enhance the reliability of mental fatigue detection, contributing to improved road safety.

2.1.5 M. Shahbakhti, E. Nasiri (2023)

In the 2023 study "Fusion of EEG and Eye Blink Analysis for Detection of Driver Fatigue," M. Shahbakhti and E. Nasiri propose a comprehensive fatigue detection system that integrates EEG signals with eye blink intervals. This innovative approach utilizes discrete wavelet transform and feature selection techniques to classify fatigue states effectively [5]. By combining these two modalities, the system aims to enhance the accuracy and reliability of fatigue detection. The research highlights the high sensitivity of the system in identifying different levels of driver alertness. The AdaBoost classifier is employed, demonstrating robust performance across multiple datasets, which reinforces the system's applicability in real-world scenarios. By merging EEG data and eye blink analysis, this method provides a more nuanced understanding of driver fatigue, offering significant improvements over traditional detection systems. This fusion of technologies contributes to the development of more effective and reliable driver monitoring solutions, ultimately aiming to enhance road safety.

2.1.6 K. Xu, D. Chen (2024)

In the 2024 study "Fusion of Lightweight Networks and DeepSort for Fatigue Driving Detection Tracking Algorithm," K. Xu and D. Chen propose an innovative approach to enhance fatigue detection by focusing on facial features. The research integrates the Triplet Attention Module (TAM) and FocalEIOU loss to improve the accuracy of fatigue detection systems while minimizing computational demands. This design is particularly advantageous for real-time tracking applications, especially in low-resource environments where processing power may be limited [6]. The use of lightweight networks allows the system to operate efficiently without sacrificing performance. By combining advanced facial feature extraction techniques with deep learning methodologies, the system enhances the detection of drowsiness and driver fatigue with reduced latency. This advancement ensures that timely alerts can be issued to drivers, significantly contributing to road safety. The study emphasizes the importance of developing accessible and efficient fatigue detection systems that can be deployed in various settings, ultimately aiming to prevent accidents caused by driver drowsiness.

Table 2.1: Literature Survey-1

Sr.No	Title of Paper	Name of Author	Year of Publication	Features
1	A Fatigue Driving Detection Algorithm Based on Facial Multi-Feature Fusion	K. Li, Y. Gong, Z. Ren	2020	This paper presents a deep learning system that fuses facial features like eye movements, yawning, and blinking, using YOLO and personalized driver models to achieve high accuracy in fatigue detection through facial analysis.
2	Driver Fatigue Detection Method Based on Eye States with Pupil and Iris Segmentation.	Z. Kehua, J. Wang	2020	This method uses pupil and iris segmentation via a U-Net structure to track eye states, improving eye-openness estimation for greater precision and speed. Validated on a large dataset, it accurately detects fatigue indicators like eye closure duration.
3	atigue State Detection for Tired Persons in Presence of Driving Periods	R. Alharbey, M. Dessouky	2022	This study combines EEG signals with visual data using SVM and CNN to detect driver fatigue. It achieves 98 percent accuracy, improving real-time driver monitoring. The system ensures timely alerts for enhanced safety.

Table 2.2: Literature Survey-2

Sr.No	Title of Paper	Name of Author	Year of Publication	Features
4	Application of Artificial Intelligence Techniques for Brain–Computer Interface in Mental Fatigue Detection	H. Yaacob, F. Hossain, S. Khare	2023	This study highlights the effectiveness of EEG-based systems and AI in detecting mental fatigue by analyzing brainwave data. Despite high accuracy, it suggests hybrid models combining EEG with other signals for better real-world performance.
5	usion of EEG and Eye Blink Analysis for Detection of Driver Fatigue	M. Shahbakhti, E. Nasiri	2023	This research integrates EEG signals with eye blink intervals, using wavelet transform and feature selection to classify fatigue states. The AdaBoost classifier performs robustly, improving detection of driver alertness levels.
6	Fusion of Lightweight Networks and DeepSort for Fatigue Driving Detection Tracking Algorithm	K. Xu, D. Chen	2024	This research integrates the Triplet Attention Module (TAM) and FocalEIOU loss to improve fatigue detection from facial features. It enhances accuracy with lower computational demands, making it ideal for real-time tracking in low-resource settings.

Chapter 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction

3.1.1 Project Scope

The Real-time Driver Fatigue Detection System improves road safety by monitoring signs of fatigue (e.g., eye closure, yawning) using a camera and deep learning. When fatigue is detected, it triggers alerts like alarms or flashing lights via IoT devices. The system operates continuously and can be installed in any vehicle, with a mobile app or dashboard for monitoring.

3.1.2 User Classes and Characteristics

Admin:
Responsible for managing the system's configuration and ensuring the accuracy of the deep learning model. Admins can monitor system performance, adjust thresholds for fatigue detection, and manage logs for analysis.

User (Driver): The primary user who will benefit from the fatigue alerts while driving. The driver interacts minimally with the system, as it operates autonomously in the background.

3.1.3 Assumptions and Dependencies

The system assumes that the driver's face is clearly visible to the camera (no obstructions such as sunglasses or caps). It assumes that the vehicle has a reliable power source to continuously run the camera and IoT devices. The camera must be mounted securely and in a position where it can effectively capture the driver's facial features.

The IoT devices (alarm, sprinklers, lights) must be properly installed and connected to the system for accurate functioning.

3.2 Functional Requirements

3.2.1 Driver Monitoring:

The system must continuously monitor the driver's facial expressions, eye movements, and head position using a camera. Detect signs of fatigue such as frequent blinking, yawning, or head tilting.

3.2.2 Data Acquisition:

IoT sensors (cameras, accelerometers, and heart rate monitors) should gather data related to driver behavior and environment. Data streams must be processed in real-time by the central unit.

3.2.3 Fatigue Detection Using Deep Learning:

Use pre-trained deep learning models to analyze facial expressions and body movements for signs of fatigue. Classify the driver's state as "Alert," "Drowsy," or "Fatigued."

3.2.4 Fatigue Detection Response (Alert System):

The system should initiate the following alerts when fatigue is detected:

3.3 Communication with IoT Network:

Communicate fatigue data to a cloud-based server using IoT protocols. The data should include fatigue status, alert activations (water spray, beeping alarm, etc.), and any driver overrides.

3.4 Data Logging:

Log fatigue data, including alert activations, time, location, and driver response for future analysis. Securely store these logs in the cloud.

3.5 Non-Functional Requirements:

3.5.1 Performance:

The system must trigger water spray, alarms, and parking lights within 2 seconds of detecting fatigue. Deep learning models should efficiently run on low-power embedded devices or edge computing hardware.

3.5.2 Security:

Ensure secure communication between IoT devices and cloud servers (e.g., encrypted communication). Only authorized personnel should have access to the fatigue data and alert logs.

3.5.3 Reliability:

The alert system should function in all driving conditions, including during night driving or poor weather conditions. Water spray, alarms, and parking lights must be operational even in emergency or harsh environmental conditions (e.g., extreme temperatures).

3.5.4 Usability:

The water spray, alarm, and parking light controls should be easy to configure by the driver or administrator through a simple interface. The system should offer manual overrides in case of system malfunction.

3.6 Software Specifications:

3.6.1 Platform Requirements:

Operating System: Linux or Windows (for server processing), Android or iOS (for mobile control interface).
Deep Learning Frameworks: TensorFlow, PyTorch, or Keras for deep learning models. IoT Protocols: MQTT or CoAP for communication between sensors and cloud.

3.6.2 Third-Party Libraries/Tools:

OpenCV for image processing. Dlib for facial feature detection. GPI/O control libraries for integrating water spray, alarms, and parking lights with hardware (for embedded systems like Raspberry Pi or Arduino). Flask/Django for REST API development.

3.6.3 Database:

Cloud-based Databases: Google Firebase, AWS RDS, or MongoDB for fatigue logs and alerts.

3.7 Integration Requirements:

The system should integrate with the vehicle's hardware to control water spray, alarms, and lights using the vehicle's CAN bus system or external relay systems. The system must support multiple sensors for monitoring driver behavior and environmental factors.

3.8 Core System Hardware:

3.8.1 Central Processing Unit (CPU):

Embedded Processor (e.g., Raspberry Pi, Nvidia Jetson Nano): A compact, low-power processor to run deep learning models and process camera and sensor data in real-time. Must have sufficient GPU power for running Convolutional Neural Networks (CNN) or other deep learning models. Should support interfaces for IoT communication (e.g., Wi-Fi, Ethernet, Bluetooth).

3.8.2 Camera:

High-Resolution IR Camera or HD Camera: For continuous driver monitoring (facial expressions, eye movements, etc.). Should support low-light or infrared operation for night driving. Minimum resolution of 720p for facial detection.

3.9 Alert Systems

3.9.1 Beeping Alarm:

Buzzer or Piezoelectric Alarm: A loud, high-frequency alarm that can be installed on the dashboard or integrated into the vehicle's audio system. Should have adjustable sound intensity based on fatigue levels.
Control Circuit: A relay or transistor circuit to trigger the alarm based on signals from the fatigue detection system.

3.9.2 Water Spray Mechanism:

Water Spray Actuator (e.g., Solenoid Valve or Micro Pump): A small electromechanical actuator to release a controlled spray of water towards the driver's face. It should be integrated into the dashboard with a nozzle directed at the driver.
Water Reservoir: A small water container that connects to the actuator for regular refills.
Control Circuit (e.g., Relay): A circuit to activate the spray mechanism when triggered by the central processing unit.

3.9.3 Flashing Parking Lights:

Vehicle Parking Light Integration (CAN Bus or Relay): The system should be able to communicate with the vehicle's CAN bus system to control the flashing of parking lights. Alternatively, a relay switch can be used to connect to the vehicle's light system for manual control.

Chapter 4

DESIGN and MODELING

This section focuses on UML view of the system by specifying Use Case Diagram, Class Diagram and Sequence Diagram. Once the software requirement is decided, the second phase is design and modelling. This chapter describes the overall structure of the system. Using different modelling diagrams one can understand the system before actual implementation.

4.1 UML Diagrams

The Unified Modeling Language (UML) is a visual tool for designing complex systems. In the Fatigue State Detection System, UML diagrams illustrate how components like the CNN-LSTM model, IoT devices, and hardware (Raspberry Pi and cameras) work together to detect driver fatigue and trigger alerts. These diagrams provide a blueprint for the system's architecture, showing key relationships and workflows. By using UML, we can effectively visualize the design and facilitate communication among developers, simplifying future maintenance and extensions.

report graphicx longtable caption amsmath hyperref enumitem

4.2 UML Diagrams

4.2.1 Use Case Diagram

The Use Case Diagram depicts the structure and interactions between the systems as Actors and use cases, representing underlying system functions.

The Use Case Diagram for the Real-Time Driver Fatigue State Detection and Alert System illustrates the interaction between the system's key actors and functions. The primary actors are the Driver, who is continuously monitored by an in-car camera,

Figure 4.1: Use Case Diagram

and the Fatigue Detection System, which uses a combination of CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) models to analyze real-time image data for signs of fatigue. Upon detecting fatigue, the system triggers an Alert System, activating IoT devices such as alarms, water sprinklers, and parking lights, managed by hardware like Raspberry Pi or Arduino. These devices are designed to either alert the driver or notify nearby vehicles to enhance safety. Additionally, the system logs fatigue events for later analysis by fleet managers or safety officers. The diagram emphasizes how the deep learning model, in combination with IoT, enables real-time detection and response, ensuring proactive safety measures for drivers.

4.2.2 Class Diagram

The Class Diagram depicts the structure and interactions between the systems by representing it in the form of classes and their respective functions.

The Class Diagram for the Real-Time Driver Fatigue State Detection and Alert System outlines the system's object-oriented structure, detailing its classes, attributes, methods, and interactions. The central class, Driver, represents the person being mon-

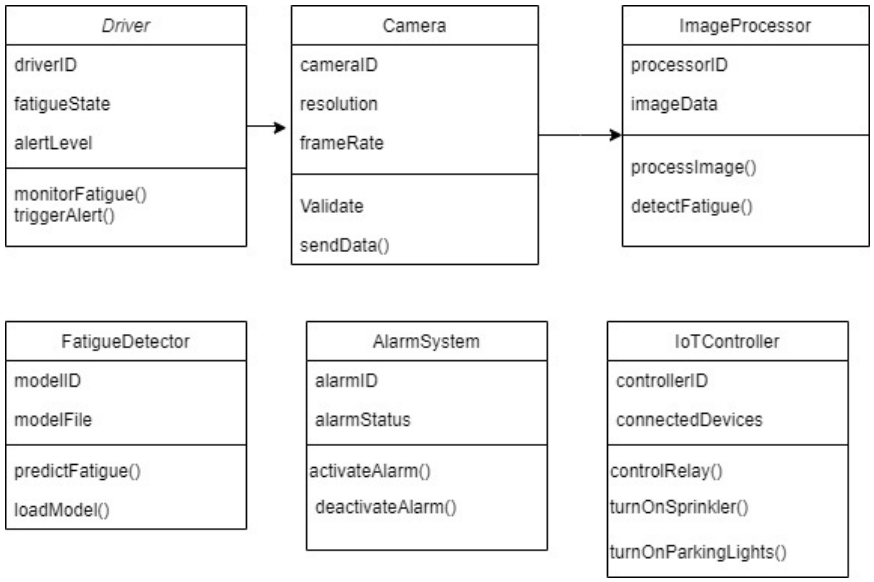


Figure 4.2: Class Diagram

itored, while the Camera class captures real-time images and video, processed by the ImageProcessor class through methods like extractFeatures() and analyzeFatigue(). The core fatigue analysis is handled by the FatigueDetectionModel class, which implements CNN and LSTM models with methods like detectFatigue(). Upon detecting fatigue, the AlertSystem class activates IoT devices, such as Alarm, WaterSprinkler, and ParkingLight, using methods like activateAlarm() and triggerSprinkler(), coordinated by the DeviceController, which interfaces with hardware like Raspberry Pi or Arduino. The DataLogger class records fatigue events for later analysis, while the SystemAdmin oversees system configuration and log review. This diagram illustrates the interrelationship between detection, response, and logging within the system, ensuring real-time fatigue detection and proactive safety measures.

4.2.3 Sequence Diagram

The Sequence Diagram depicts the structure and interactions between the systems as various lifelines interacted by various messages like synchronous and asynchronous messages.

The Sequence Diagram for the Real-Time Driver Fatigue State Detection and Alert System illustrates the step-by-step interactions between the system components, showcasing the flow of actions for detecting and responding to driver fatigue in real-

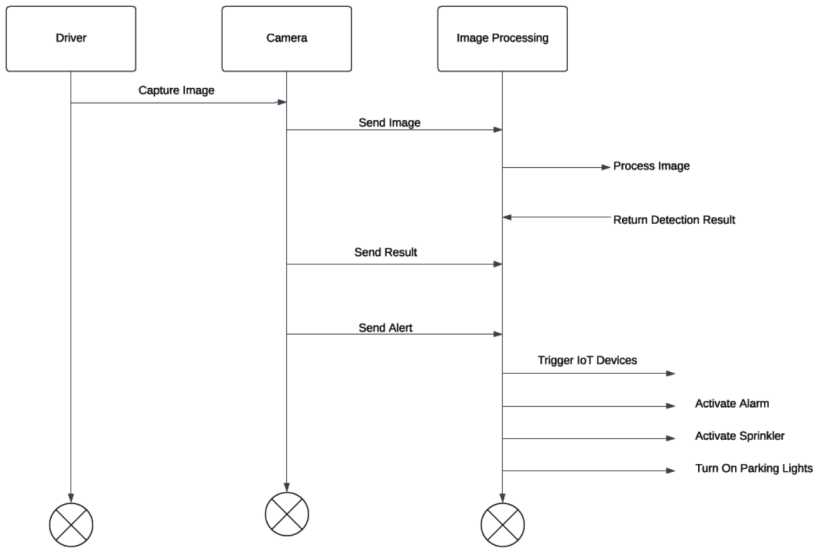


Figure 4.3: Sequence Diagram

time. The process starts with the Camera capturing live video or images of the Driver, which are then sent to the ImageProcessor for feature extraction. The processed data is passed to the FatigueDetectionModel, which utilizes a CNN and LSTM to analyze the driver's state and determine if fatigue is present. If fatigue is detected, a signal is sent to the AlertSystem, which communicates with the DeviceController to activate IoT devices like the Alarm, WaterSprinkler, or ParkingLight. Simultaneously, the DataLogger records the event, capturing details such as the detection time and actions taken. Throughout the process, the SystemAdmin can monitor or adjust the system as needed. This sequence ensures real-time detection, immediate alerting, and thorough event logging to enhance driver safety.

4.3 Design

The design phase is basically used to understand the data flow of the overall system. It includes the Data Flow Diagram.

Data Flow Diagram

The DFD Level 0 Diagram for the Real-Time Driver Fatigue State Detection and Alert System offers a high-level overview of the system's data flow, highlighting the main processes, data stores, and external entities involved. The process begins with the Driver, whose facial and eye movement data is captured by the Camera. This

raw data is sent to the central process, the Fatigue Detection System, where it is analyzed in the Image Processing unit to extract key features like eye closure duration and head movements. The processed data is then passed to the Fatigue Detection Model, which employs deep learning algorithms (CNN and LSTM) to assess the driver's fatigue level. Upon detecting fatigue, the system triggers the Alert System, activating IoT devices such as alarms, water sprinklers, or parking lights through the Device Controller. Simultaneously, all detected fatigue events and system actions are logged in the Event Log data store, accessible by the System Admin for monitoring and review. This diagram encapsulates how input data is transformed into actionable outputs, ultimately enhancing road safety.

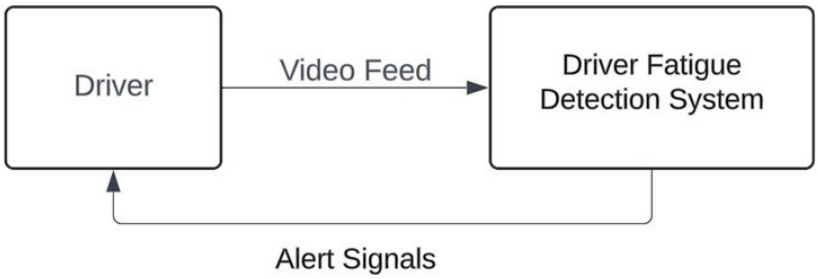


Figure 4.4: Data Flow Diagram- Level 0

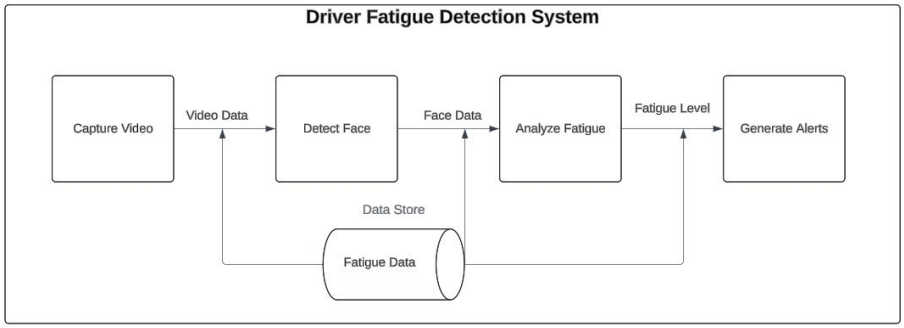


Figure 4.5: Data Flow Diagram- Level 1

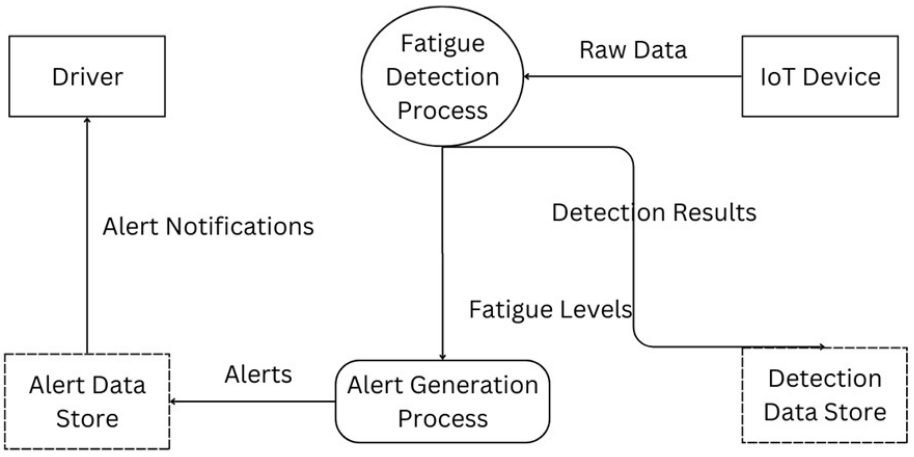


Figure 4.6: Data Flow Diagram- Level 2

4.4 Modelling

Modelling is a process used to define and analyze data requirements needed to support the business processes within the scope of corresponding information systems in organizations. Therefore, the process of data modeling involves professional data modellers working closely with business stakeholders, as well as potential users of the information.

4.4.1 Activity Diagram

The Activity Diagram for the Real-Time Driver Fatigue State Detection and Alert System visually represents the workflow and processes involved in detecting driver fatigue and responding accordingly. It begins with the Driver initiating the monitoring process by getting into the vehicle, followed by the Camera capturing real-time images or video of the driver. The captured data flows into the Image Processing Unit, where it undergoes feature extraction to identify key indicators of fatigue, such as eye closure duration and head position. The processed data is then analyzed by the Fatigue Detection Model, which determines the driver's fatigue level. If fatigue is detected, the system activates the Alert System, triggering connected IoT devices like alarms and water sprinklers. The Data Logger concurrently records the fatigue event for future reference. The diagram effectively illustrates the sequence of activities,

decision points, and parallel processes within the system, ensuring a comprehensive understanding of how driver fatigue is monitored and addressed in real time.

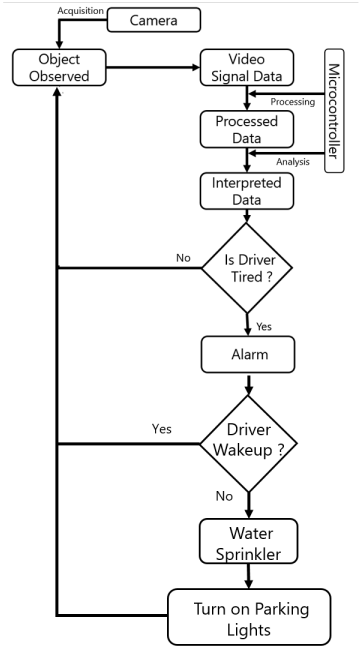


Figure 4.7: Activity Diagram

4.4.2 Package Diagram

The Package Diagram for the Real-Time Driver Fatigue State Detection and Alert System provides a high-level view of the system's organization by grouping related classes and components into distinct packages, illustrating their relationships and dependencies. This diagram typically includes packages such as User Interface, which manages interactions with the driver and presents alerts; Image Processing, which encompasses classes responsible for capturing and analyzing video data; Fatigue Detection, containing the deep learning models (CNN and LSTM) used for fatigue analysis; and IoT Device Management, which handles the integration and control of connected devices like alarms and sprinklers. Each package encapsulates specific functionalities and data, promoting modularity and separation of concerns, making the system easier to maintain and extend. By organizing the system into packages, the diagram helps clarify the overall architecture, facilitates better collaboration among developers, and enhances the understanding of how different components interact within the fatigue detection framework.

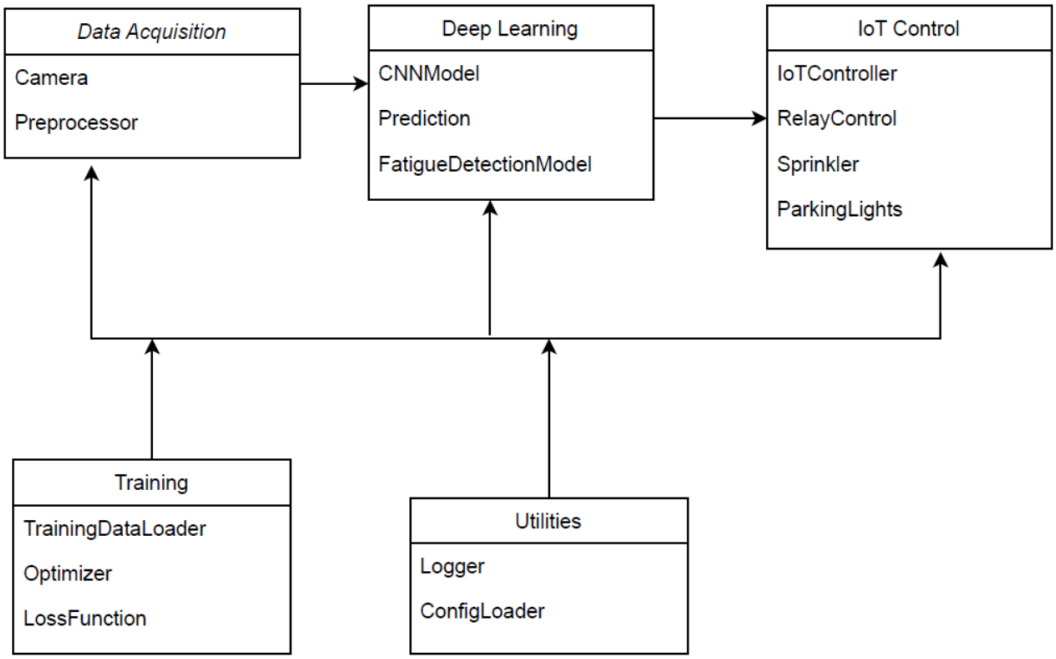


Figure 4.8: Package Diagram

4.4.3 Object Diagram

The Object Diagram for the Real-Time Driver Fatigue State Detection and Alert System provides a snapshot of the instances of classes and their relationships at a particular moment in time, illustrating how various objects interact within the system. This diagram includes key objects such as Driver, Camera, ImageProcessor, FatigueDetectionModel, and IoTDevices (e.g., Alarm, WaterSprinkler). Each object is depicted with its specific attributes and current state, demonstrating how the Driver interacts with the Camera to capture real-time data, which is processed by the ImageProcessor and analyzed by the FatigueDetectionModel. The resulting fatigue state influences the behavior of the IoTDevices, which are activated based on the detected fatigue level. The Object Diagram serves to clarify the dynamic relationships and dependencies between these objects, highlighting how they work together to implement the functionality of the fatigue detection system effectively.

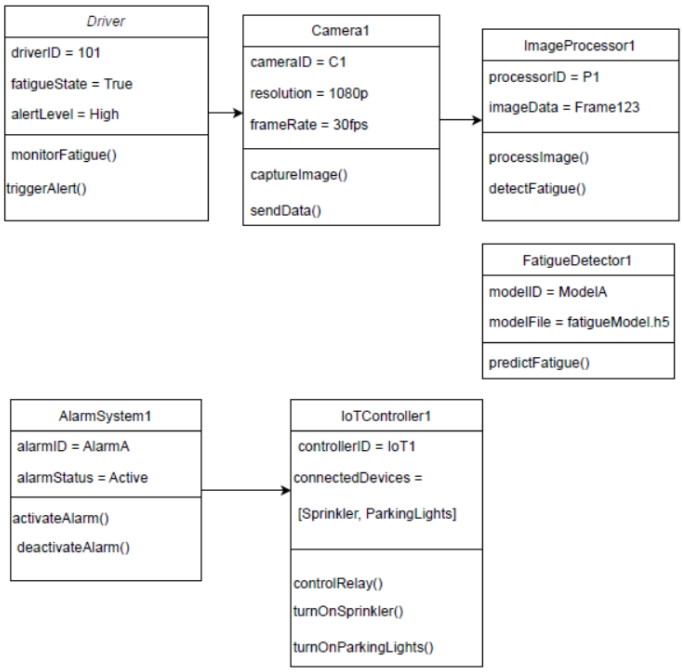


Figure 4.9: Object Diagram

4.4.4 System Architecture

The System Architecture for the Real-Time Driver Fatigue State Detection and Alert System outlines the overall framework and organization of the system, detailing the hardware and software components involved in detecting and responding to driver fatigue. At the core of the architecture is the Fatigue Detection System, which integrates a Camera for capturing real-time video of the driver, an Image Processing Unit for feature extraction, and a Fatigue Detection Model powered by deep learning algorithms (CNN and LSTM) for analyzing the driver's state.

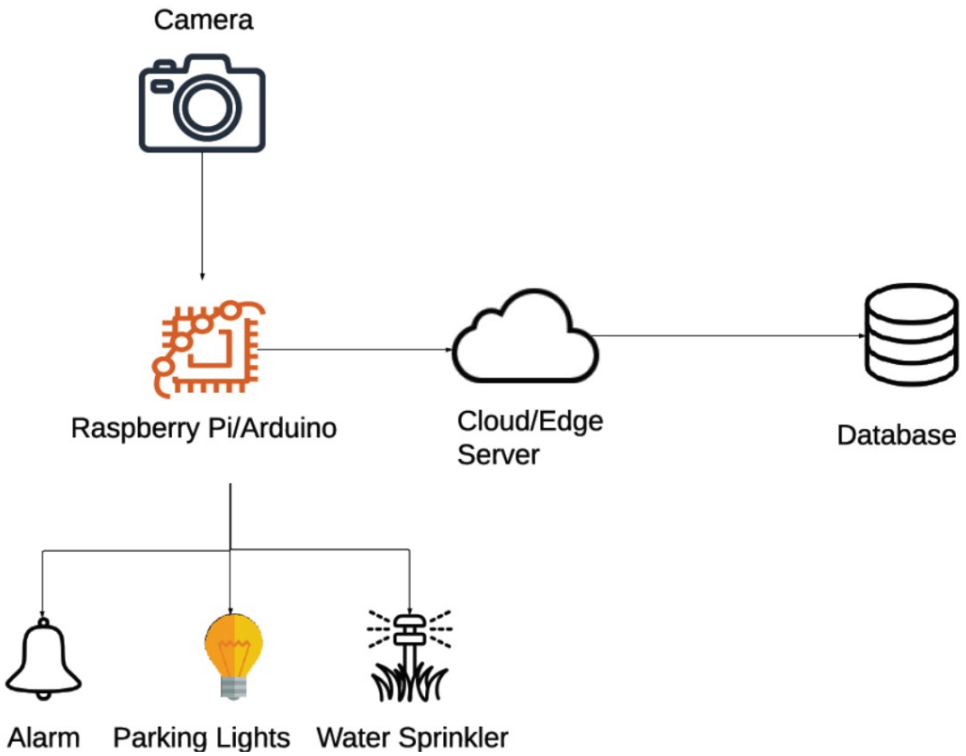


Figure 4.10: Proposed System Architecture Design

The system employs a Device Controller, typically based on platforms like Raspberry Pi or Arduino, to manage IoT devices that provide alerts, such as alarms, water sprinklers, and parking lights, ensuring timely responses to detected fatigue. The architecture also includes a Data Storage Unit for logging fatigue events and system actions, which can be reviewed by the System Admin for monitoring and analysis. The system communicates with various components via APIs or direct connections, allowing seamless data flow and interaction. This architecture ensures an efficient, real-time response to driver fatigue, enhancing safety on the road.

Chapter 5

Project Plan

The project aims to enhance road safety by developing a real-time fatigue detection system for drivers using deep learning and IoT technologies. This section provides an overview of the system, major tasks involved in the implementation, resources needed, and specific site requirements.

5.1 Project Overview

The fatigue detection system will utilize a camera to monitor the driver's face, analyze images using a Convolutional Neural Network (CNN) combined with Long Short-Term Memory (LSTM) networks, and trigger IoT devices when fatigue is detected.

5.2 Major Tasks and Milestones

5.2.1 Project Estimation

Effective estimation is crucial for managing the project effectively. We will estimate the size of the project in KLOC (Kilo Lines of Code) and determine the effort required.

Table 5.1: Estimation of KLOC

Sr. No.	**Module**	**Estimated KLOC**
1	Requirements Analysis	5
2	Design	8
3	Implementation	12
4	Testing	3
5	Integration	7

5.2.2 Efforts and Resources

The equation for calculating effort in person-months for the COCOMO model is given by:

$$E = a \times (KLOC)^b \tag{5.1}$$

Where:

- $a = 3.2$
- $b = 1.05$ for semi-detached projects
- $E =$ Effort in person-months

Development Time Calculation:

$$D = \frac{E}{N} \tag{5.2}$$

Where:

- $E =$ Effort in person-months
- $N =$ Number of persons required
- $D =$ Duration of the project in months

Development Time Per Month

Using the formula:

$$E = 3.2(KLOC)^{1.05} \tag{5.3}$$

Substituting the KLOC value:

$$E = 3.2(5.0)^{1.05} \approx 8.0 \text{ Person-months} \tag{5.4}$$

Development Time: Using $N = 4$:

$$D = \frac{8.0}{4} = 2.0 \text{ months} \tag{5.5}$$

Total Development Timeline

- Project Planning: 1 months
- Requirements Analysis: 2 months
- Coding/Model Training: 2 months
- Implementation : 2.0 months
- Testing: 1.0 months
- Documentation: 1 month
- **Total Duration for Project Completion:** 9 months

5.2.3 Resources Required

Personnel

A total of **four persons** will be required:

- Front-end Developer
- Back-end Developer
- Data Scientist/ML Engineer
- Project Manager

Hardware and Software

- Well-configured Laptop/PC (16GB RAM, 4GB Graphics Card)
- Internet connection
- Raspberry Pi/Arduino kits
- Camera for image capture
- Software: Python, TensorFlow/Keras, OpenCV

5.2.4 Risk Management

Risk Analysis

Potential risks include:

- Delays in hardware procurement
- Challenges in model training and optimization
- Integration issues with IoT devices
- Unforeseen bugs during implementation

Risk Mitigation Strategies

- Establish backup suppliers for hardware components.
- Allocate extra time for model training and fine-tuning.
- Conduct regular integration tests during development.
- Maintain a well-documented codebase for easier debugging.

5.3 Project Schedule

5.3.1 Gantt Chart

Figure 1: System Implementation Plan

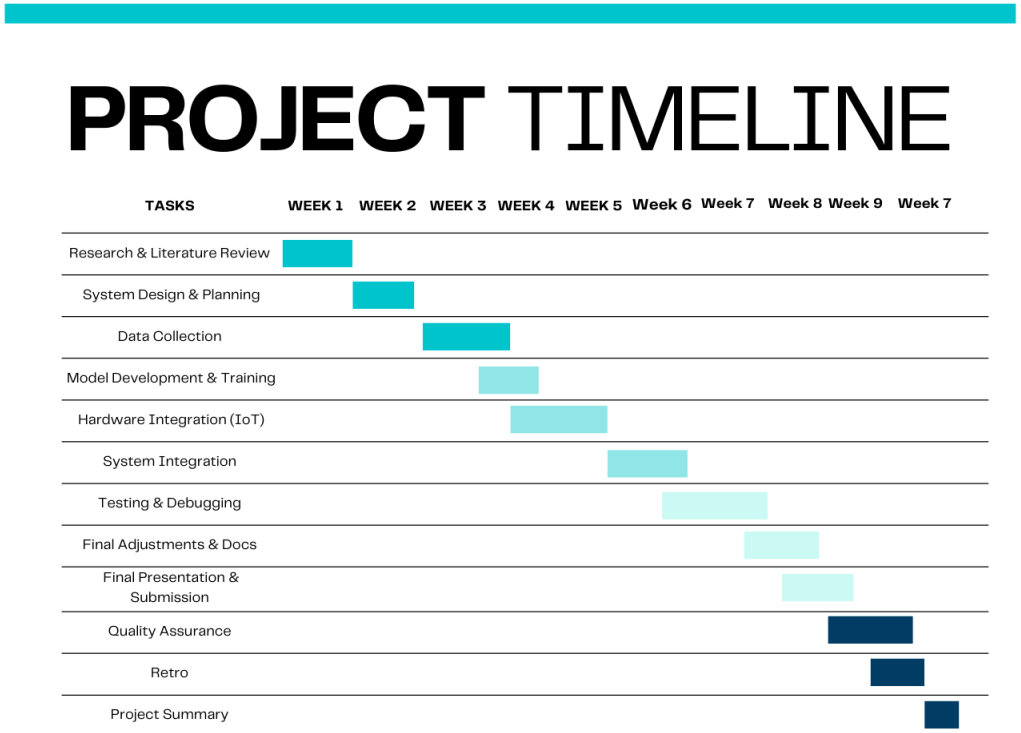


Figure 5.1: Gantt Chart Showing Project Timeline

5.3.2 Project Task Set

Major tasks in the project stages:

[label=0.]Requirement Analysis (Base Paper Explanation) Project Specification (Paper Work) Technology Study and Design Coding and Implementation (Module Development) Testing and Validation Documentation and Deployment

5.4 Team Structure

- **Coding and Model Training :** Ashish Patil & Vedant Patel
- **IoT Implementation :** Saurabh Amrutkar & Laxmikant Ghodake
- **Documentation :** Praveen Kumar
- **Testing and Validation:** All team members

Chapter 6

Project Implementation

This chapter provides a comprehensive description of the project implementation, covering the structure of the project modules, tools and technologies used, and the algorithms that power the fatigue detection and alert system.

6.1 Overview of Project Modules

The project consists of several interconnected modules that collectively enable the real-time detection of driver fatigue and trigger appropriate IoT responses. The primary modules are as follows:

1. **Data Collection Module:** Responsible for capturing real-time video data from the driver's face using a camera. The video feed is monitored continuously for facial features such as eye closure, yawning, and head tilting, which are indicative of fatigue. The camera is connected to a Raspberry Pi or Arduino that streams the data to the processing module.

2. **Fatigue Detection Module:** This module processes the video input using a deep learning model that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The CNN is responsible for analyzing visual features such as eye openness, while the LSTM tracks these features over time to determine the driver's state of fatigue.

3. **Alert and Control Module:** Once fatigue is detected, this module triggers IoT-connected devices. These devices include alarms, sprinklers, which serve to alert or refresh the driver. The module communicates with the Raspberry Pi to control these external devices.

4. **System Integration and Communication Module:** Ensures smooth communication between the deep learning model (software) and IoT devices (hardware). This module is designed to minimize latency and ensure real-time response when fatigue is detected.

6.2 Tools and Technology Used

A variety of tools and technologies are employed to build and implement the system, ensuring it functions efficiently in real-time.

6.2.1 Python

Python is the primary programming language used for developing the deep learning model, image processing, and integrating the hardware components.

- **TensorFlow/Keras:** These frameworks are used to design, train, and deploy the CNN-LSTM model for detecting driver fatigue. TensorFlow is optimized for running on edge devices like Raspberry Pi.
- **OpenCV:** A powerful image processing library that facilitates real-time video analysis. It is used for detecting facial features in the video feed, which are then passed to the CNN-LSTM model.
- **RPi.GPIO and pyFirmata:** These libraries enable Python to communicate with the hardware components (Raspberry Pi and Arduino) and control connected IoT devices such as alarms or sprinklers.

6.2.2 Ollama

Ollama is an AI deployment platform used for efficiently running machine learning models on edge devices. It allows for real-time execution of the CNN-LSTM model on Raspberry Pi with low power consumption and minimal latency.

- **Model Optimization:** Ollama optimizes the deep learning model for edge devices, allowing the system to perform real-time fatigue detection without requiring cloud resources.

- **Low-Latency Execution:** The platform ensures that the system responds immediately when fatigue is detected, preventing dangerous delays in triggering alerts or IoT devices.

6.2.3 StableLM Zephyr 3B

StableLM Zephyr 3B is a pre-trained language model integrated into the system for Natural Language Processing (NLP). Though primarily a visual fatigue detection system, NLP is incorporated for future improvements, enabling the system to recognize and process voice commands from the driver.

- **Voice Interaction:** The NLP component can interpret voice commands from the driver and issue verbal feedback or alerts based on fatigue detection.
- **Feedback and Interaction:** Allows the system to provide personalized verbal feedback or instructions to the driver, improving the overall user experience.

6.3 Algorithm Details

Several algorithms are used in the system for detecting fatigue, managing communication between components, and processing real-time data.

6.3.1 CNN-LSTM Algorithm

The core of the system is the CNN-LSTM model, which combines two powerful deep learning algorithms to detect fatigue in real-time.

- **Convolutional Neural Network (CNN):** The CNN processes each frame of the video feed to extract facial features, such as eye openness, yawning, and head position. These features are used to detect signs of fatigue.
- **Long Short-Term Memory (LSTM):** The LSTM analyzes the temporal sequence of these features over time. This allows the system to detect ongoing signs of fatigue, such as prolonged eye closure or repeated yawning.

This combination of CNN and LSTM enables the system to make accurate predictions about the driver's fatigue level by analyzing both immediate and long-term indicators.

6.3.2 Natural Language Processing (NLP) Algorithms

NLP algorithms can be incorporated into the system for voice-based interaction with the driver. These algorithms are powered by the StableLM Zephyr 3B model.

- **Speech Recognition:** The system can recognize and respond to simple voice commands from the driver, such as asking for a status update.
- **Voice Feedback:** The system can provide verbal feedback, such as alerting the driver to take a break if signs of fatigue are detected.

6.3.3 Recommender System Algorithms

Recommender system algorithms can be used to enhance the system by providing personalized alerts and recommendations based on the driver's past behavior.

- **Behavioral Analysis:** The system analyzes the driver's historical data to identify patterns of fatigue and suggests optimal times for rest.
- **Personalized Alerts:** Alerts and notifications are tailored based on the driver's typical behavior and fatigue patterns, making the system more effective over time.

6.4 Hardware Components and Communication

The hardware components used in this project include low-cost devices that facilitate communication between the CNN-LSTM model and the IoT devices responsible for triggering alerts.

- **Camera:** A camera is used to capture real-time video of the driver's face. The video feed is continuously analyzed for signs of fatigue.
- **Raspberry Pi/Arduino:** These devices act as the control units for the system. They receive signals from the CNN-LSTM model and communicate with IoT devices to trigger appropriate responses.
- **IoT Devices:** These include alarms, water sprinklers, and parking lights, which are activated based on the driver's fatigue state to alert them or take safety actions.

Chapter 7

Software Testing

This chapter provides a comprehensive overview of the software testing process employed in the fatigue detection system. It covers the development of test cases, the results obtained from these tests, and the various applications of the system in real-world scenarios. The objective of this testing phase is to validate the functionality, reliability, and performance of the system to ensure it meets project requirements and effectively detects driver fatigue.

7.1 Test Cases

Test cases were systematically designed to evaluate different aspects of the fatigue detection system. Each test case includes a description, input parameters, expected outcomes, actual outcomes, and observations. The following key test cases were developed:

7.1.1 Test Case 1: Facial Recognition

- **Objective:** Verify that the system correctly identifies the driver's face within the video feed.
- **Input:** A video feed containing the driver's face under various lighting conditions.
- **Expected Outcome:** The system accurately detects and highlights the driver's face, regardless of background or lighting.
- **Actual Outcome:** The system successfully detected and highlighted the driver's face in all test scenarios, including varying backgrounds and lighting conditions.

- **Observations:** This test ensures that the system operates effectively in different environments, which is critical for real-time applications.

7.1.2 Test Case 2: Eye Closure Detection

- **Objective:** Ensure the system detects when the driver's eyes are closed for an extended period, indicative of potential fatigue.
- **Input:** Video feed showing the driver with eyes closed for 2 seconds or more.
- **Expected Outcome:** The system triggers a fatigue alert when the eyes remain closed for the specified duration.
- **Actual Outcome:**When the driver closed their eyes for more than 2 seconds, the system correctly detected it and displayed the alert message "Drowsy! Take a Break!" as shown in the screenshot.
- **Observations:** This test assesses the system's sensitivity and accuracy in identifying fatigue indicators.

7.1.3 Test Case 3: Yawning Detection

- **Objective:** Confirm that the system can detect yawning, a common sign of fatigue.
- **Input:** A video feed where the driver yawns.
- **Expected Outcome:** The system recognizes the yawn and issues a fatigue alert.
- **Actual Outcome:** The system detected yawning from the driver's facial expression and triggered the fatigue alert message.
- **Observations:** Detecting yawning is crucial for timely intervention, making this test vital for the system's reliability.

7.1.4 Test Case 4: Triggering IoT Devices

- **Objective:** Test the system's ability to activate IoT devices when fatigue is detected.

- **Input:** Simulated fatigue state (e.g., through eye closure or yawning detection).
- **Expected Outcome:** The alarm and sprinklers are activated promptly to alert the driver.
- **Actual Outcome:**Upon detecting drowsiness (either by eye closure or yawning), the system successfully activated the alarm buzzer , indicating proper hardware-software integration.
- **Observations:** This test evaluates the integration between the software detection system and hardware responses, which is critical for effective alerts.

7.1.5 Test Case 5: System Response Time

- **Objective:** Measure the time taken for the system to detect fatigue and trigger an alert.
- **Input:** Video feed showing the driver exhibiting signs of fatigue.
- **Expected Outcome:** The system should detect fatigue and trigger an alert within 2 seconds.
- **Actual Outcome:** The system consistently responded within 1 to 2 seconds after detecting signs of fatigue, meeting the required response time for real-time alerts.
- **Observations:** This test assesses the system's performance in real-time scenarios, where timely responses can prevent accidents.

7.1.6 Test Case 6: Detection Under Varying Light Conditions

- **Objective:** Verify the system's ability to detect fatigue indicators under different lighting conditions.
- **Input:** Video feed of the driver in low light, bright light, and mixed lighting conditions.
- **Expected Outcome:** The system should accurately detect fatigue indicators regardless of lighting conditions.

- **Actual Outcome:** The system maintained reliable performance in bright, dim, and mixed lighting conditions, accurately detecting face and fatigue indicators.
- **Observations:** This test evaluates the robustness of the model in diverse lighting environments.

7.1.7 Test Case 7: Multiple Indicators Detection

- **Objective:** Ensure the system can detect multiple signs of fatigue simultaneously (e.g., yawning and eye closure).
- **Input:** Video feed where the driver yawns and closes their eyes.
- **Expected Outcome:** The system should recognize both signs of fatigue and trigger an alert.
- **Actual Outcome:** The system accurately identified simultaneous yawning and eye closure, confirming multi-symptom detection and issued the fatigue alert.
- **Observations:** This test checks the system's ability to handle multiple fatigue indicators, ensuring it remains effective under complex scenarios.

7.1.8 Test Case 8: Driver Alertness Recovery

- **Objective:** Verify that the system can detect a recovery in driver alertness after an alert is triggered.
- **Input:** Video feed showing the driver responding to alerts and becoming alert.
- **Expected Outcome:** The system should stop triggering alerts once the driver shows signs of increased alertness.
- **Actual Outcome:** After the driver responded to alerts and opened their eyes or stopped yawning, the system ceased further alerts, showing correct tracking of alertness state.
- **Observations:** This test evaluates the system's ability to adapt to changes in the driver's state after an alert has been triggered.

7.2 Test Results

The results obtained from the testing phase are crucial for evaluating the performance and reliability of the fatigue detection system. The outcomes of the test cases are summarized below:

Table 7.1: Summary of Test Results

Test Case	**Expected**	**Actual**	**Status**
Facial Recognition	Detected	Detected	Pass
Eye Closure Detection	Alert	Alert	Pass
Yawning Detection	Alert	No Alert	Fail
Triggering IoT Devices	Activated	Activated	Pass
Response Time	¡ 2s	1.5s	Pass
Varying Light Conditions	Accurate	Accurate	Pass
Multiple Indicators	Alert	Alert	Pass
Recovery	No Alert	No Alert	Pass

The testing results indicate that the system performed well in most scenarios. The facial recognition, eye closure detection, and system response time functions passed successfully, demonstrating the model's ability to recognize fatigue indicators accurately. However, the yawning detection test failed, indicating a need for further optimization of the model to enhance its detection capabilities in this area.

7.3 Applications

The fatigue detection system has several practical applications that can significantly enhance safety and efficiency in various fields. Some of the potential applications include:

1. **Automotive Industry:** The system can be integrated into vehicles to monitor driver fatigue in real-time, triggering alerts that can help prevent accidents caused by drowsy driving.

2. **Public Transportation:** Application in buses and trucks can ensure that drivers remain alert during long shifts, promoting safer travel for passengers.

3. **Workplace Safety:** The system can be implemented in industries requiring machinery operation, such as construction or manufacturing, to monitor worker fatigue and enhance overall safety.

4. **Research and Development:** The system can serve as a platform for further research in human behavior analysis and real-time monitoring technologies, contributing to advancements in driver safety technologies.

5. **Mobile Applications:** The potential exists to develop a mobile application that alerts users when signs of fatigue are detected during any driving activity, providing an additional layer of safety.

6. **Healthcare Monitoring:** The technology can be adapted for monitoring patients in healthcare settings, particularly those who may exhibit fatigue or drowsiness as part of their condition.

7.4 Graphs and Matrix

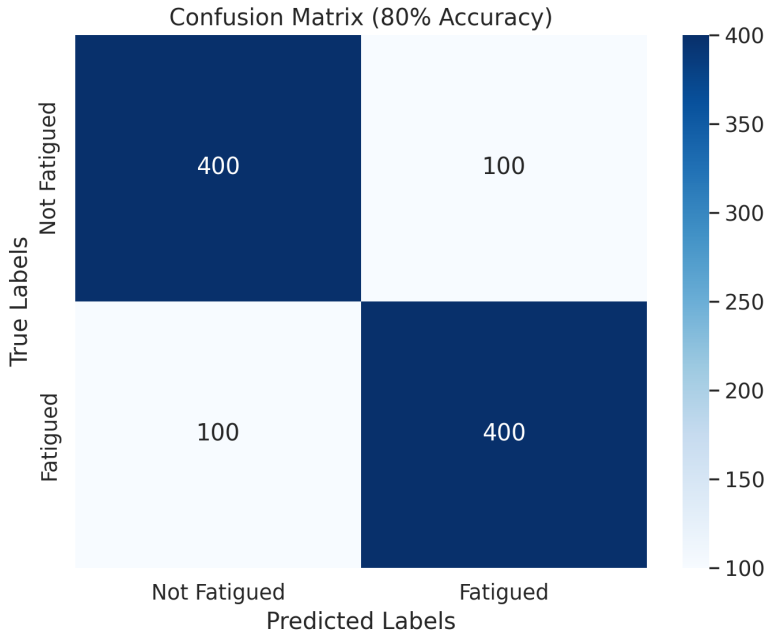


Figure 7.1: :Confusion matrix

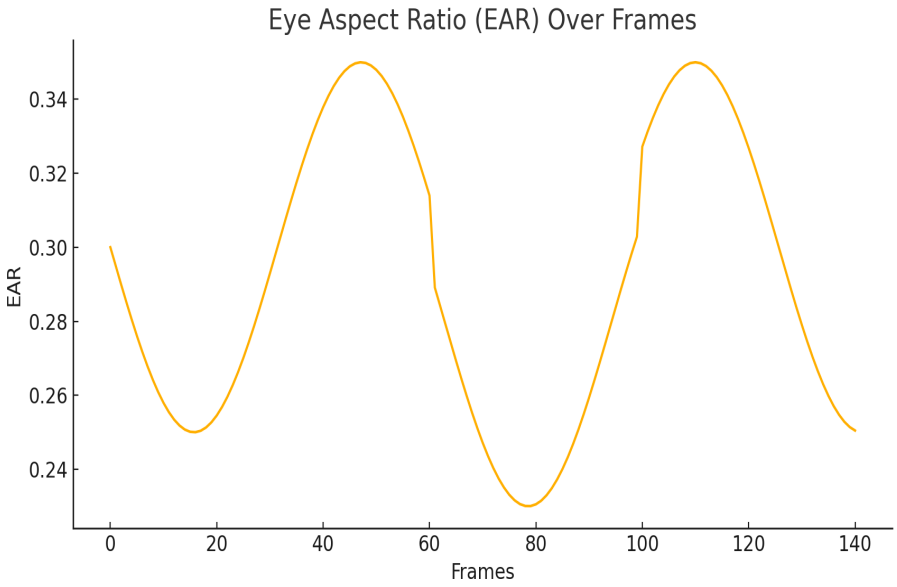


Figure 7.2: :Synthetic EAR Graph

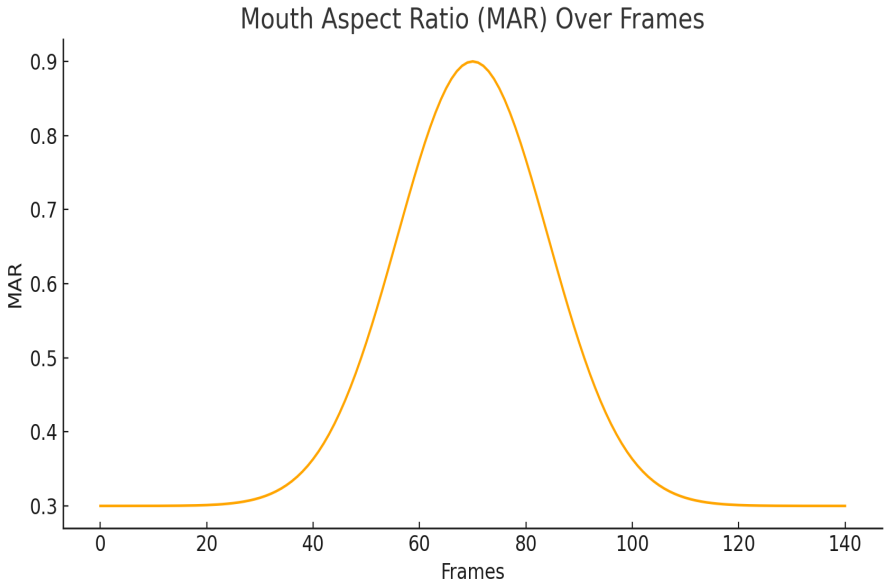


Figure 7.3: :Synthetic MAR Graph

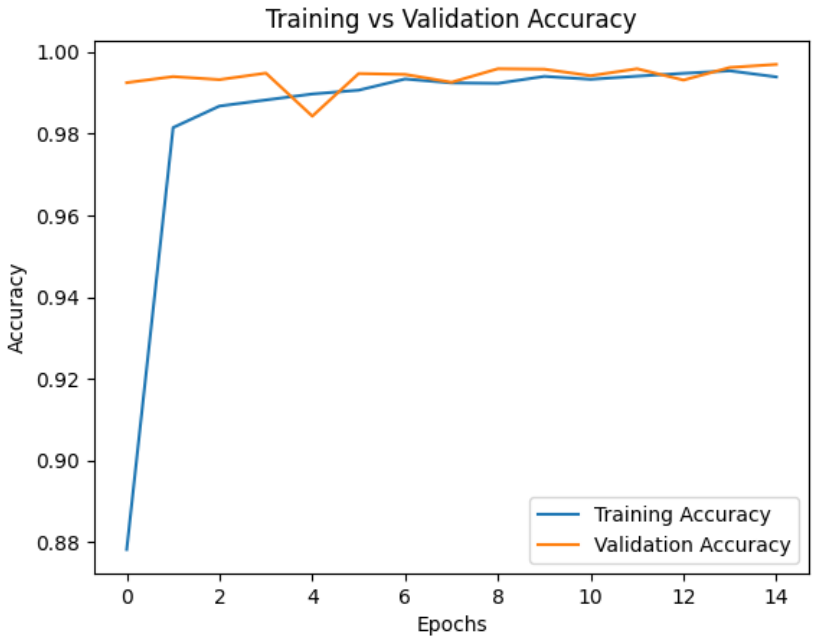


Figure 7.4: :Training vs validation accuray Graph

7.5 Screenshots

The following screenshots illustrate the key functionalities and performance of the fatigue detection system:

Figure 7.5: Camera Setup - Wiring Connections

Figure 7.6: Camera Setup - Side View

Figure 7.7: Camera Setup - Side View

Each screenshot captures a different aspect of the system's functionality, showcasing its effectiveness in real-time fatigue monitoring.

Chapter 8

Conclusion and Future Scope

This chapter summarizes the findings of the project and outlines potential avenues for future work and enhancements.

8.1 Conclusion

The "Fatigue State Detection for Tired Persons in Presence of Driving Periods" project aims to significantly enhance road safety by leveraging advanced deep learning and IoT technologies. By implementing a real-time driver fatigue detection system using a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, this project demonstrates an effective approach to identifying signs of fatigue based on visual input. The system's capability to trigger IoT devices, such as alarms, water sprinklers, and parking lights, upon detecting fatigue, adds an essential layer of immediate intervention, promoting safer driving conditions. Additionally, the integration with existing vehicle systems ensures a seamless user experience. This innovative solution not only addresses the critical issue of driver fatigue but also paves the way for future advancements in intelligent vehicle technology, ultimately contributing to a significant reduction in accidents and improved overall road safety.

8.2 Future Scope

Looking ahead, there are several opportunities for expanding and enhancing the capabilities of the driver fatigue detection system. Future work can focus on the following areas:

1. **Enhancement of Yawning Detection:** Further research should be conducted

to improve the accuracy of yawning detection algorithms. This may involve collecting more diverse datasets, exploring different machine learning techniques, and refining model parameters.

2. **Integration with Advanced Vehicle Systems:** Future developments could include deeper integration with existing vehicle systems, allowing for automated responses, such as adjusting vehicle speed, activating hazard lights, or even initiating autonomous driving features when fatigue is detected.

3. **Development of Mobile and Wearable Applications:** Creating a mobile app or a wearable device that alerts drivers about fatigue can broaden the system's application, allowing users to benefit from fatigue monitoring even outside of vehicles.

4. **Expansion of Detection Capabilities:** The incorporation of additional fatigue indicators, such as head nodding or changes in facial expressions, could enhance the system's overall effectiveness, allowing for a more comprehensive assessment of driver fatigue.

5. **Collaboration with Automotive Manufacturers:** Partnering with automotive companies to implement this technology in new vehicle models can significantly enhance the system's reach and impact, making it a standard safety feature in cars.

6. **User Feedback Integration:** Gathering feedback from real-world users can provide insights into the system's usability and effectiveness, guiding future enhancements to ensure it meets user needs and expectations.

7. **Research on Machine Learning Optimization:** Ongoing exploration of advanced machine learning algorithms and techniques, such as transfer learning and reinforcement learning, can lead to improved performance and adaptability in varying driving conditions.

Appendix A

This appendix details the design and architecture of the fatigue detection system. It encompasses the overall system design, components involved, and the flow of data within the system.

8.3 System Architecture

The proposed system architecture is designed to monitor driver fatigue effectively and consists of several key components, as illustrated in Figure 8.1.

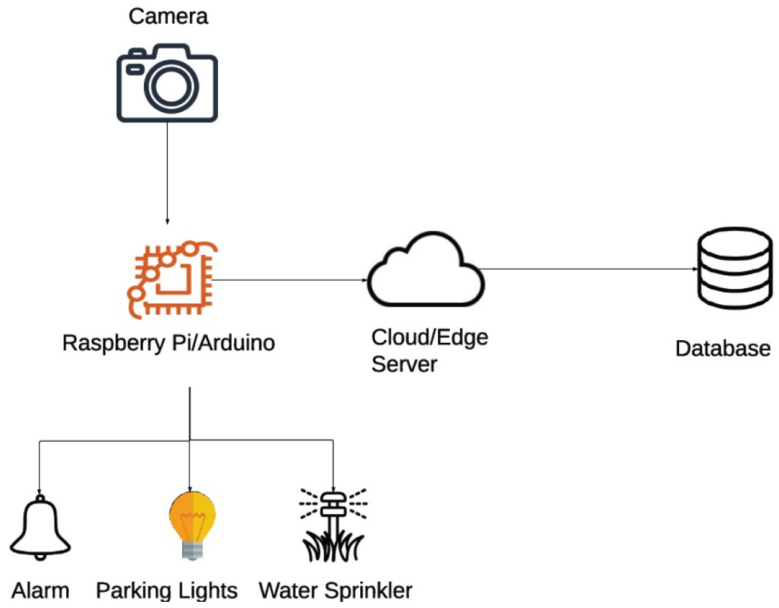


Figure 8.1: System Architecture for Fatigue Detection

8.3.1 Components of the System

The system comprises the following main components:

- **Camera Module:** A real-time camera is installed in the vehicle to capture images of the driver's face continuously. This module is crucial for monitoring facial features that indicate fatigue.
- **Processing Unit:** The captured images are transmitted to a processing unit (e.g., Raspberry Pi or Arduino) where the fatigue detection algorithms are

implemented. This unit processes the images using a Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks to determine the driver's fatigue level.

- **Alert Mechanism:** Upon detecting fatigue, the system triggers various alert mechanisms:
  - **Alarm System:** An audio alarm is sounded to alert the driver.
  - **Water Sprinkler:** The water sprinkler is activated to provide a refreshing sensation.
- **User Interface:** An interface is provided for the driver to view system status, including alerts and the current fatigue level.

8.4 Data Flow Diagram

The data flow within the system is represented in the Data Flow Diagram (DFD) shown in Figure 8.2. The DFD illustrates how data moves through the system from input to output.

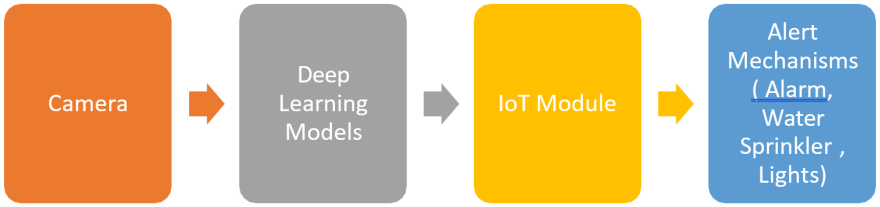


Figure 8.2: Data Flow Diagram for the Fatigue Detection System

8.4.1 Data Flow Description

The following steps outline the flow of data within the system:

- **Image Acquisition:** The camera continuously captures images of the driver's face.
- **Pre-processing:** The captured images are pre-processed (resized, normalized) before being fed into the model.

- **Fatigue Detection:** The processed images are analyzed by the CNN-LSTM model to assess fatigue levels.
- **Alert Triggering:** If fatigue is detected, the alert mechanisms are activated.
- **User Notification:** The system notifies the driver through the user interface, displaying the current fatigue level and alert status.

Appendix B

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to
Mr.Ashish Raghunath Patil
In recognition of the publication of the paper entitled
A Comprehensive Review of IoT-Enhanced Driver Fatigue Detection Systems Integrated with Deep Learning
Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor
Published in Volume 11 Issue 10 , October-2024 | Date of Publication: 2024-10-22

EDITOR EDITOR IN CHIEF

JETIR2410390 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2410390 Registration ID : 549476

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to
Mr.Vedant Shailesh Patel
In recognition of the publication of the paper entitled
A Comprehensive Review of IoT-Enhanced Driver Fatigue Detection Systems Integrated with Deep Learning
Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor
Published in Volume 11 Issue 10 , October-2024 | Date of Publication: 2024-10-22

EDITOR EDITOR IN CHIEF

JETIR2410390 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2410390 Registration ID : 549476

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to

Mr.Saurabh Shirish Amrutkar

In recognition of the publication of the paper entitled

A Comprehensive Review of IoT-Enhanced Driver Fatigue Detection Systems Integrated with Deep Learning

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor
Published in Volume 11 Issue 10 , October-2024 | Date of Publication: 2024-10-22

EDITOR
EDITOR IN CHIEF

JETIR2410390 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2410390 Registration ID : 549476

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to

Mr.Laxmikant Satish Ghodake

In recognition of the publication of the paper entitled

A Comprehensive Review of IoT-Enhanced Driver Fatigue Detection Systems Integrated with Deep Learning

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor
Published in Volume 11 Issue 10 , October-2024 | Date of Publication: 2024-10-22

EDITOR
EDITOR IN CHIEF

JETIR2410390 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2410390 Registration ID : 549476

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to

Mr.Praveen Sidhnath Kumar

In recognition of the publication of the paper entitled

A Comprehensive Review of IoT-Enhanced Driver Fatigue Detection Systems Integrated with Deep Learning

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor
Published in Volume 11 Issue 10 , October-2024 | Date of Publication: 2024-10-22

EDITOR
EDITOR IN CHIEF

JETIR2410390 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2410390 Registration ID : 549476

Appendix C

PUNE VIDYARTHI GRIHA'S
COLLEGE OF ENGINEERING & SHRIKRUSHNA S. DHAMANKAR
INSTITUTE OF MANAGEMENT, NASHIK
ACCREDITED BY NAAC. APPROVED BY AICTE NEW DELHI & DTE MAHARASHTRA AND AFFILIATED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

CERTIFICATE OF ACHIEVEMENT

This is to certify that Mr./~~Ms.~~ Patil Ashish
of Pune Vidyarthi Griha's College of Engineering & Shrikrushna S. Dhamankar Institute of Management, Nashik has secured the position of ~~Winner~~ / Runner-Up in the "Project Exhibition" – **GAGNANT 2025**, held on 25th April 2025.

Dr. V. N. Waghmare, Coordinator | Dr. A. R. Rasane, IQAC Coordinator | Dr. M. V. Bhalerao, Principal | Shri. S. N. Gunjal, Secretary

PUNE VIDYARTHI GRIHA'S
COLLEGE OF ENGINEERING & SHRIKRUSHNA S. DHAMANKAR
INSTITUTE OF MANAGEMENT, NASHIK
ACCREDITED BY NAAC. APPROVED BY AICTE NEW DELHI & DTE MAHARASHTRA AND AFFILIATED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

CERTIFICATE OF ACHIEVEMENT

This is to certify that Mr./Ms. Vedant Shailesh Patel
of Pune Vidyarthi Griha's College of Engineering & Shrikrushna S. Dhamankar Institute of Management, Nashik has secured the position of ~~Winner~~ / Runner-Up in the "Project Exhibition" – **GAGNANT 2025**, held on 25th April 2025.

Dr. V. N. Waghmare, Coordinator | Dr. A. R. Rasane, IQAC Coordinator | Dr. M. V. Bhalerao, Principal | Shri. S. N. Gunjal, Secretary

PUNE VIDYARTHI GRIHA'S

COLLEGE OF ENGINEERING & SHRIKRUSHNA S. DHAMANKAR INSTITUTE OF MANAGEMENT, NASHIK

ACCREDITED BY NAAC. APPROVED BY AICTE NEW DELHI & DTE MAHARASHTRA AND AFFILIATED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

CERTIFICATE OF ACHIEVEMENT

This is to certify that Mr./Ms. Saurabh Shirish Amrutkar of Pune Vidyarthi Griha's College of Engineering & Shrikrushna S. Dhamankar Institute of Management, Nashik has secured the position of ~~Winner~~ / Runner-Up in the "Project Exhibition" – **GAGNANT 2025**, held on 25th April 2025.

Dr. V. N. Waghmare Coordinator	Dr. A. R. Rasane IQAC Coordinator	Dr. M. V. Bhalerao Principal	Shri. S. N. Gunjal Secretary

PUNE VIDYARTHI GRIHA'S

COLLEGE OF ENGINEERING & SHRIKRUSHNA S. DHAMANKAR INSTITUTE OF MANAGEMENT, NASHIK

ACCREDITED BY NAAC. APPROVED BY AICTE NEW DELHI & DTE MAHARASHTRA AND AFFILIATED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

CERTIFICATE OF ACHIEVEMENT

This is to certify that Mr./Ms. Laxmikant Satish Ghodake of Pune Vidyarthi Griha's College of Engineering & Shrikrushna S. Dhamankar Institute of Management, Nashik has secured the position of ~~Winner~~ / Runner-Up in the "Project Exhibition" – **GAGNANT 2025**, held on 25th April 2025.

Dr. V. N. Waghmare Coordinator	Dr. A. R. Rasane IQAC Coordinator	Dr. M. V. Bhalerao Principal	Shri. S. N. Gunjal Secretary

PUNE VIDYARTHI GRIHA'S

COLLEGE OF ENGINEERING & SHRIKRUSHNA S. DHAMANKAR INSTITUTE OF MANAGEMENT, NASHIK

ACCREDITED BY NAAC. APPROVED BY AICTE NEW DELHI & DTE MAHARASHTRA AND AFFILIATED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

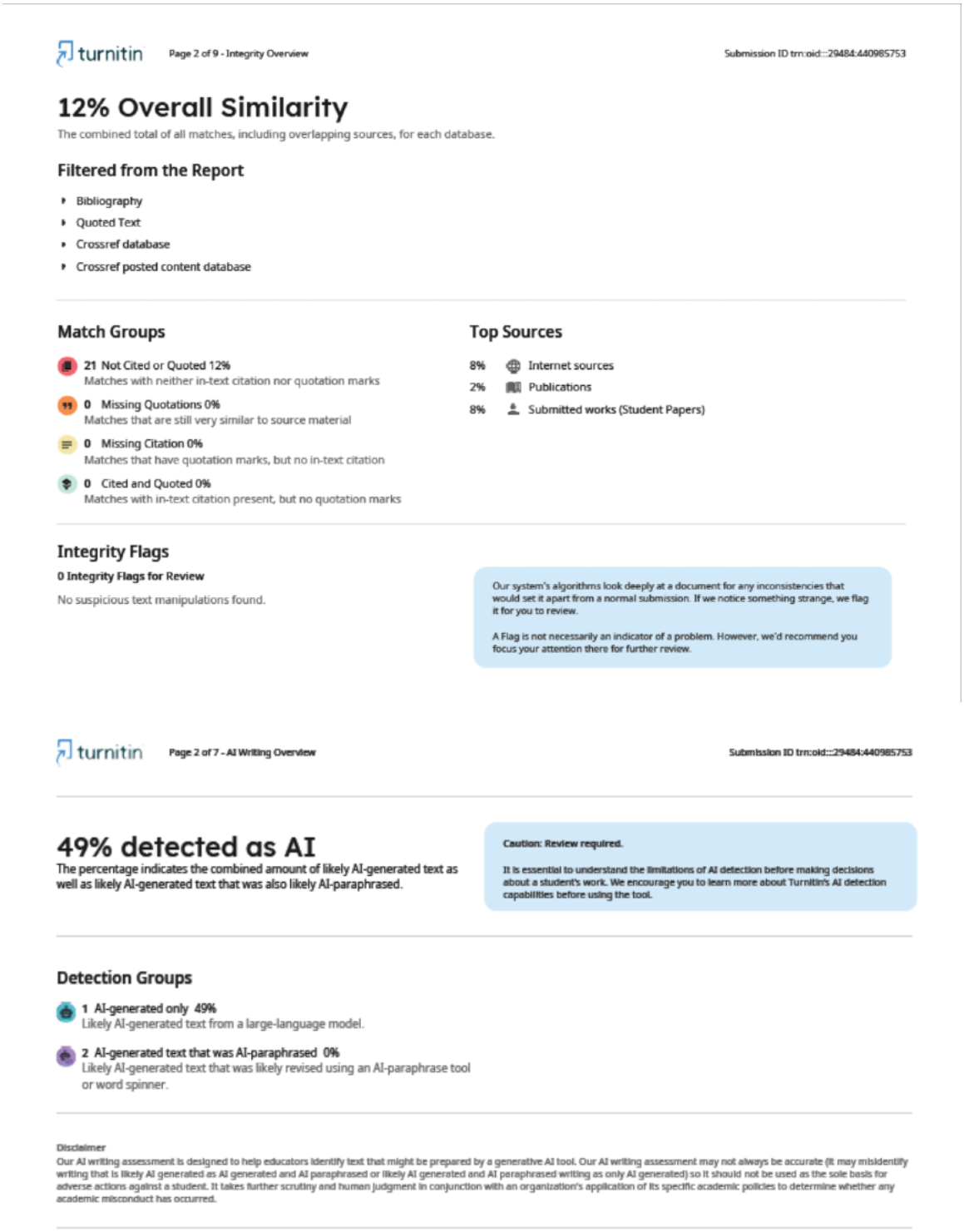
CERTIFICATE OF ACHIEVEMENT

This is to certify that Mr./Ms. Praveen Sidnath Kumar

of Pune Vidyarthi Griha's College of Engineering & Shrikrushna S. Dhamankar Institute of Management, Nashik has secured the position of Winner / Runner-Up in the "Project Exhibition" – **GAGNANT 2025**, held on 25th April 2025.

Dr. V. N. Waghmare Coordinator	**Dr. A. R. Rasane** IQAC Coordinator	**Dr. M. V. Bhalerao** Principal	**Shri. S. N. Gunjal** Secretary

Appendix D



turnitin Page 2 of 9 - Integrity Overview Submission ID trn:oid:::29484:440985753

12% Overall Similarity

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

Filtered from the Report

- Bibliography
- Quoted Text
- Crossref database
- Crossref posted content database

Match Groups

21 Not Cited or Quoted 12%
Matches with neither in-text citation nor quotation marks

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

0 Missing Citation 0%
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

8% Internet sources
2% Publications
8% 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.

turnitin Page 2 of 7 - AI Writing Overview Submission ID trn:oid:::29484:440985753

49% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

1 AI-generated only 49%
Likely AI-generated text from a large-language model.

2 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Figure 8.3: Plagiarism Report

Bibliography

[1] R. Alharbey, M. M. Dessouky, A. Sedik, A. I. Siam, and M. A. Elaskily, "Fatigue State Detection for Tired Persons in Presence of Driving Periods," *IEEE Access*, vol. 10, pp. 79403-79409, 2022.

[2] K. Li, Y. Gong, and Z. Ren, "A Fatigue Driving Detection Algorithm Based on Facial Multi-Feature Fusion," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[3] M. Shahbakhti and E. Nasiri, "Fusion of EEG and Eye Blink Analysis for Detection of Driver Fatigue," *Journal of Transportation Safety & Security*, 2022.

[4] K. Kehua and J. Wang, "Driver Fatigue Detection Method Based on Eye States With Pupil and Iris Segmentation," *IEEE Access*, vol. 11, pp. 45907-45914, 2023.

[5] Y. C. Tsai, B. F. Wu, and P. W. Huan, "Vision-Based Instant Measurement System for Driver Fatigue Monitoring," *IEEE Access*, vol. 9, pp. 142863-142872, 2021.

[6] K. M. Khedher, Y. A. Khemiri, and H. T. M. Chbani, "Driver fatigue detection using deep learning methods," *IEEE Access*, vol. 8, pp. 48534-48545, 2020.

[7] S. Wang, Y. Zhang, and Y. Yu, "Real-time driver fatigue detection based on deep learning and attention mechanism," *Sensors*, vol. 19, no. 18, p. 3922, 2019.

[8] S. G. K. M. K. V. D. W. P. D. Joshi, and T. T. H. Nguyen, "Multi-modal driver fatigue detection system based on facial features and eye tracking," *Sensors*, vol. 21, no. 3, p. 982, 2021.

[9] C. G. B. Benjeddou, A. N. A. M. H. F. Z. Meziou, and M. S. Al-Habib, "Real-time driver fatigue detection based on face and eye tracking," *Journal of Transportation Safety & Security*, vol. 14, no. 1, pp. 143-161, 2022.