REAL TIME YAWNING DETECTION FOR DRIVER DISTRACTION MONITORING USING DEEP LEARNING AND IOT

AI19711 PROJECT PHASE-I REPORT

Submitted by

SRIRAM K (221501144)
SUDHARSAN S (221501149)
TAMILARASAN D (221501157)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

RAJALAKSHMI ENGINEERING COLLEGE

(AUTONOMOUS), CHENNAI-602 105

NOVEMBER 2025

RAJALAKSHMI ENGINEERING COLLEGE

(an Autonomous Institution Affiliated to Anna University Chennai)

BONAFIDE CERTIFICATE

Certified that this Phase-I Thesis titled **"REAL TIME YAWNING DETECTION FOR DRIVER DISTRACTION MONITORING USING DEEP LEARNING AND IOT"** is the Bonafide work of **SRIRAM K (2116221501144), SUDHARSAN S (2116221501149), TAMILARASAN D (2116221501157)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. K. SEKAR
Professor and Head
Department of Artificial Intelligence
and Machine Learning
Rajalakshmi Engineering College
Chennai- 602 105

Mrs. J. JASMINE GABRIEL
Assistant Professor
Department of Artificial Intelligence
and Machine Learning
Rajalakshmi Engineering College
Chennai- 602 105

Submitted for the project viva-voce examination held on __________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT VISION

To promote highly Ethical and Innovative Computer Professionals through excellence in teaching, training and research.

DEPARTMENT MISSION

- To produce globally competent professionals, motivated to learn the emerging technologies and to be innovative in solving real world problems.
- To promote research activities amongst the students and the members of faculty that could benefit the society.
- To impart moral and ethical values in their profession.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: To equip students with essential background in computer science, basic electronics and applied mathematics.

PEO 2: To prepare students with fundamental knowledge in programming languages, and tools and enable them to develop applications.

PEO 3: To encourage the research abilities and innovative project development in the field of AI, ML, DL, networking, security, web development, Data Science and also emerging technologies for the cause of social benefit.

PEO 4: To develop professionally ethical individuals enhanced with analytical skills, communication skills and organizing ability to meet industry requirements.

PROGRAM OUTCOMES (POs)

PO1: Engineering knowledge: Apply the knowledge of Mathematics, Science, Engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO 4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

A graduate of the Artificial Intelligence and Machine Learning Program will demonstrate

PSO 1: Foundation Skills: Ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, web design, AI, machine learning, deep learning, data science, and networking for efficient design of computer-based systems of varying complexity. Familiarity and practical competence with a broad range of programming language, tools and open source platforms.

PSO 2: Problem-Solving Skills: Ability to apply mathematical methodologies to solve computational task, model real world problem using appropriate AI and ML algorithms. To understand the standard practices and strategies in project development, using open-ended programming environments to deliver a quality product.

PSO 3: Successful Progression: Ability to apply knowledge in various domains to identify research gaps and to provide solution to new ideas, inculcate passion towards higher studies, creating innovative career paths to be an entrepreneur and evolve as an ethically social responsible AI and ML professional.

COURSE OBJECTIVE

- To identify and formulate real-world problems that can be solved using Artificial Intelligence and Machine Learning techniques.
- To apply theoretical and practical knowledge of AI/ML for designing innovative, data-driven solutions.
- To integrate various tools, frameworks, and algorithms to develop, test, and validate AI/ML models.
- To demonstrate effective teamwork, project management, and communication skills through collaborative project execution.
- To instill awareness of ethical, societal, and environmental considerations in the design and deployment of intelligent systems.

COURSE OUTCOME

- **CO1:** Analyze and define a real-world problem by identifying key challenges, project requirements and constraints.
- **CO2:** Conduct a thorough literature review to evaluate existing solutions, identify research gaps and formulate research questions.
- **CO3:** Develop a detailed project plan by defining objectives, setting timelines, and identifying key deliverables to guide the implementation process.

- **CO4:** Design and implement a prototype or initial model based on the proposed solution framework using appropriate AI tools and technologies.
- **CO5:** Demonstrate teamwork, communication, and project management skills by preparing and presenting a well-structured project proposal and initial implementation results.

CO-PO-PSO Mapping

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	2	2	2	3	2	3	3	3	3	3
CO2	3	3	3	3	3	2	-	-	2	2	2	3	3	2	2
CO3	3	3	3	2	3	1	1	2	3	3	3	3	3	3	3
CO4	3	3	3	3	3	2	1	2	3	2	2	3	3	3	3
CO5	1	1	1	1	1	-	-	-	3	3	3	3	1	-	2

Note: Correlation levels 1, 2 or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

No correlation: “-”

ABSTRACT

Driver fatigue remains a major contributor to road accidents worldwide, making early detection essential for improving transportation safety. This project introduces an intelligent and fully automated driver monitoring system that identifies yawning, which is a strong physiological indicator of drowsiness, through continuous in cabin video analysis. A live camera captures real time facial movements, and a lightweight deep learning model processes each frame to accurately recognize yawning patterns with minimal latency. The system operates entirely on local edge hardware, ensuring immediate responses without relying on external servers and maintaining reliability even in low connectivity environments. Its optimized architecture supports efficient real time inference on resource constrained devices, making it suitable for commercial vehicles, personal cars, and low cost embedded platforms. Enhanced pre processing and improved feature extraction techniques increase detection stability under varying lighting conditions, diverse head poses, and partial occlusions that naturally occur during driving. The system also enables continuous monitoring, producing fatigue alerts when repeated yawning behavior is detected over a period of time, thereby helping prevent accidents by warning drivers before fatigue becomes hazardous. Overall, this solution provides a practical, reliable, and cost effective approach to improving road safety with high adaptability, low computational overhead, and strong performance in dynamic driving environments.

ACKNOWLEDGEMENT

Initially, we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D. and** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. K. SEKAR, M.E., Ph.D.,** Professor and Head of the Department of Artificial Intelligence and Machine Learning for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our Internal Guide and Reviewer, **Mrs. J. JASMINE GABRIEL M.E., (Ph.D).,** Assistant Professor, Department of Artificial Intelligence and Machine Learning. Rajalakshmi Engineering College for her valuable guidance throughout the course of the project. We also thank our Project Co-ordinator. **Dr. K. RAJU M.E., Ph.D.,** Associate Professor, Department of Artificial Intelligence and Machine Learning, Rajalakshmi Engineering College for her guidance throughout the project work.

Sriram K	**Sudharsan S**	**Tamilarasan D**
(221501144)	(221501149)	(221501157)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBRIEVATIONS

AI	Artificial Intelligence
API	Application Programming Interface
CNN	Convolution Neural Network
EAR	Eye Aspect Ratio
FN	False Negative
FP	False Positive
IoT	Internet of Things
LSTM	Long Short Term Memory
mAP	Mean Average Precision
NTHU-DDD	NTHU Driver Drowsiness Dataset
SVM	Support Vector Machine
TP	True Positive
TN	True Negative
ViT	Vision Transformer
VASC	Vascular Lesions
YOLO	You Only Look Once
YOLOv8n	YOLO Version 8 Nano
YAWDD	Yawning Detection Dataset

CHAPTER 1

INTRODUCTION

Road safety has become a critical global concern, with driver fatigue and drowsiness accounting for a significant percentage of traffic related accidents. Long driving hours, lack of sleep, monotonous road conditions, and poor cabin environments often lead to reduced alertness. One of the earliest and most visible signs of fatigue is yawning, which indicates decreased concentration and slower reflexes. Detecting such signs at the right moment can prevent accidents and save lives. Advancements in computer vision, deep learning, and embedded Internet of Things (IoT) systems have enabled the development of automated driver monitoring solutions capable of recognizing behavioral cues in real time. These technologies offer significant improvements over traditional manual or sensor based methods, which are costly, intrusive, or unreliable in dynamic driving conditions.

1.1 OVERVIEW OF ARTIFICIAL INTELLIGENCE IN DRIVER MONITORING

Artificial Intelligence (AI) has transformed modern vehicle safety by enabling advanced driver state monitoring systems capable of detecting fatigue and distraction, which are major causes of road accidents. Deep learning models, particularly Convolutional Neural Networks (CNNs) and real time object detection frameworks like You Only Look Once (YOLO), analyze facial expressions and behavioral patterns to identify early drowsiness indicators such as yawning, prolonged eye closure, reduced facial activity, and head tilting. By continuously processing live video streams, AI systems offer automated, accurate, and real time monitoring without human intervention. Unlike traditional sensor based methods, these AI driven solutions provide higher reliability, faster response times, and seamless deployment on embedded hardware platforms such as Raspberry Pi through optimized lightweight models. Overall, these advancements significantly enhance road safety and contribute to the development of intelligent transportation systems.

1.2 CHALLENGES IN DRIVER FATIGUE DETECTION

Although AI based detection systems have improved significantly, several challenges persist in diagnosing driver fatigue accurately:

- **Variations in Lighting Conditions:** Driving environments constantly change (daylight, night, tunnels, shadows), which can affect visibility and hinder facial feature detection.
- **Head Pose and Occlusion Issues:** Drivers may move their head, wear masks, spectacles, caps, or partially block the face, making detection harder.
- **Real Time Processing Constraints:** Deep learning models must run on low power hardware such as Raspberry Pi, requiring lightweight architectures and optimized inference.
- **Dataset Limitations** Experiments require a diverse set of facial images to ensure accuracy across various conditions, including differences in skin tones, facial shapes, age groups, and environmental variations.

These challenges motivate the development of a reliable, real time, and embedded deep learning solution for yawning detection.

1.3 ALGORITHMS USED

In order to tackle these problems, the project employs deep learning based computer vision algorithms that enable real time analysis of driver behavior.

Phase 1: A YOLO based object detection model is used to identify yawning by detecting the mouth region and classifying it as yawn or no yawn from live video frames.

Phase 2: The algorithm is extended to recognize additional indicators of drowsiness and distraction such as prolonged eye closure, mobile phone usage, and lack of attention through a multi class detection approach, allowing the system to monitor multiple driver states simultaneously and provide a more comprehensive assessment of driver alertness.

1.3.1 YOLO Object Detection

YOLO is a highly efficient real time object detection algorithm that performs both detection and classification in a single unified process, making it ideal for continuous video based applications. In this project, YOLO is utilized to accurately identify the driver's face and mouth region and determine whether each frame indicates a yawn or a normal state, enabling reliable detection of early drowsiness cues. Its fast inference speed, strong feature extraction capability, and robustness to variations in lighting, facial appearance, and head movement make YOLO particularly effective for in cabin driver monitoring on resource constrained embedded devices like the Raspberry Pi.

1.3.2 Raspberry Pi Optimized Deep Learning

Raspberry Pi optimized deep learning focuses on adapting computationally intensive deep learning models to operate efficiently on the limited processing power of embedded hardware. In this project, techniques such as model quantization, lightweight architecture selection, and optimized inference using Python, OpenCV, and hardware aware configurations are implemented to ensure real time performance on the Raspberry Pi. These optimizations allow the system to detect yawning and other driver behaviors with minimal delay, eliminating the need for cloud services or high end GPUs. By enabling fully on device processing, the Raspberry Pi provides a compact, portable, and cost effective platform suitable for continuous in vehicle driver monitoring.

- Low power consumption suitable for automotive environments
- Real time processing without external dependencies
- Cost effective alternative to high end computational hardware

- Portable and easy to integrate with cameras and sensors
- Supports deployment of optimized deep learning models for continuous monitoring

1.4 IOT BASED ALERT GENERATION SYSTEM

1.4.1 Automated Driver Alert System

The Automated Driver Alert System activates immediate warning mechanisms whenever a yawn is detected to help the driver regain focus. Upon identifying a fatigue related event, the Raspberry Pi triggers a connected buzzer to provide an instant audible alert, ensuring the driver becomes aware of their drowsy state. Additionally, the system can be extended to support optional features such as seat vibration, mobile notifications, or other alert modes, making the response more effective and customizable for different driving scenarios.

1.4.2 Remote Notification Support (Telegram Bot)

Additionally, the system uses a Telegram Bot to support remote notifications, which enables guardians, fleet managers, or transportation companies to receive fatigue alerts instantly. When consecutive yawns are detected, the Raspberry Pi automatically forwards warning messages via the Telegram Application Programming Interface (API), guaranteeing that the appropriate people are informed instantly. By facilitating prompt intervention and ongoing monitoring outside the vehicle, this feature greatly improves safety during long distance travel.

1.5 OBJECTIVES

- To build a real time AI system capable of detecting yawning from live video footage.
- To preprocess and extract facial features related to fatigue indicators.
- To train a YOLO based deep learning model for accurate yawning

classification.

- To deploy the optimized model on a Raspberry Pi for embedded real time execution.
- To generate automatic alerts using IoT components during fatigue events.
- To evaluate the system’s performance under different environmental and user conditions.

1.6 SUMMARY

Artificial Intelligence plays a crucial role in enhancing driver safety by enabling continuous monitoring of early fatigue indicators like yawning. As drowsiness related accidents continue to rise, there is a growing need for a reliable system that can detect yawning in real time and respond instantly. This project addresses the challenges of varying lighting, facial differences, occlusions, and limited processing capacity by using a YOLO based deep learning model optimized for Raspberry Pi. The model delivers fast and accurate yawn detection directly on the device without relying on external computing. Additionally, IoT alert mechanisms such as buzzer activation and Telegram notifications ensure immediate warnings when fatigue is detected. Together, these components create an efficient and practical real time monitoring solution aimed at preventing driver fatigue related accidents.

CHAPTER 2

LITERATURE SURVEY

Existing literatures provide a broad understanding of the methods used for detecting driver drowsiness and distraction, including vision based techniques, sensor fusion approaches, and lightweight embedded models. These studies consistently show that deep learning methods achieve high accuracy, but they also reveal common challenges such as sensitivity to lighting changes, variations in facial features, dataset imbalance, and the need for efficient real time processing. Considering these findings, the proposed system adopts a lightweight YOLO based real time detector optimized for Raspberry Pi, supported by simple IoT alert mechanisms, to deliver a practical balance between accuracy, speed, and deployability for real world driver monitoring.

2.1 LITERATURE REVIEW

Ngxande et al. [1] investigated how demographic imbalance affected fairness in driver drowsiness models. They used a ResNet based CNN supported by GAN based augmentation using datasets including NTHU Drowsy Driver Detection (NTHU DDD), DROZY, Closed Eyes in the Wild (CEW), and additional African driver facial samples. Their findings confirmed better performance across ethnic groups but required a minimum level of diverse real world images to avoid bias and ensure practical system deployment.

Senniappan et al. [2] conducted an intriguing study that examined the efficacy of multimodal input for fatigue detection by combining facial images, gyroscope readings, and EEG signals. They obtained extremely dependable detection results from driving simulator data by combining LSTM and neural network fusion. However, it was difficult to extrapolate the results to more diverse driving situations due to the small participant pool and controlled testing environment. The study emphasizes the need for extensive, practical evaluations while showcasing the potential of multimodal systems.

Altameem et al. [3] proposed an early fatigue recognition approach combining SVM classification with facial landmark feature segmentation. Their dataset consisted of 50 individuals and produced accuracy from 83% to 95%. They also reported reduced detection stability when drivers encountered low illumination or camera distance fluctuation, stressing the need for robust environmental adaptation inside vehicles.

Zhang et al. [4] recently developed a federated transfer learning framework to address privacy concerns in driver monitoring. This framework enables CNN models to be trained collectively without sharing raw data. The approach ensured robust data protection through encrypted communication while maintaining competitive accuracy using datasets such as NTHU-DDD and YAWDD. Nevertheless, difficulties like higher communication overhead, synchronization problems, and inconsistent data across client devices were observed, suggesting chances to maximize the effectiveness of federated learning.

Khan et al. [5] implemented an IoT assisted fatigue alert system by combining edge computing with Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) measurements. Their solution captured fatigue cues with around 90 % accuracy but experienced disturbances due to reflective glasses, sudden light transitions, and partial face blockage inside moving vehicles.

Venkateswarlu et al. [6] designed DrowsyDetectNet, a shallow CNN architecture optimized for lightweight embedded deployment. Their approach achieved 99.2% accuracy using limited eye state and yawning image datasets. They recommended additional cross dataset validation across nighttime, long duration driving, and unconstrained road conditions to assess true generalization.

Alguindigue et al. [7] analyzed models such as SNN, 1D CNN, and CRNN to analyze HRV, EDA, and eye tracking data in order to investigate the function of physiological biosignals in fatigue detection. Across 30 simulated driving participants, their results demonstrated high precision, with HRV based detection exhibiting particularly strong performance. Real world applicability is limited by the dependence on wearable sensors and tightly regulated environments, indicating that integrating biosignal data with visual cues could improve system reliability overall.

Jarndal et al. [8] applied Vision Transformers and Swin Transformers to datasets like MRL Eye, CEW, and NTHU-DDD, introducing significant advancements through Transformer-based visual models. Compared to CNN based models, their method showed strong contextual understanding and nearly 99% accuracy. Notwithstanding these advantages, the models' high processing costs and large training data sets rendered them inappropriate for low power embedded systems. It was suggested that future studies be conducted to create lightweight transformer architectures.

Further improvement in facial analysis based drowsiness detection was observed in the work of Dixith et al. [9], who used deep CNNs combined with multi feature fusion to classify drowsy and alert states. Their system achieved close to 94% accuracy using datasets like CEW and NTHU-DDD. While effective, the model’s computational intensity and occasional dependence on cloud services introduced latency and privacy concerns. They emphasized the importance of efficient on device processing for real time safety applications.

Dixith et al. [10] evaluated hybrid CNN SVM approach, which combined deep feature extraction with EAR and MAR features from the MRL and Kaggle datasets, was evaluated in a final noteworthy contribution [10]. With an accuracy of almost 99%, this hybrid model showed great promise for identifying early indicators of drowsiness. However, its real world deployment capabilities were limited by its high computational demands and dependence on controlled datasets. The authors suggested increasing the diversity of datasets and optimizing embedded devices further.

Abtahi et al. [11] focused specifically on yawning analysis by mapping sequential mouth geometric deformation from facial landmarks. Their computer vision approach performed effectively in structured lighting and clear face visibility. They observed difficulties in performance when drivers turned their heads or when unpredictable light reflections occurred during night travel.

Makhmudov et al. [12] combined yawning and eye state behavior to strengthen fatigue estimation stability. Their multimodal fusion helped reduce false alarms during extended driving periods and encouraged resource efficient embedded deployment for public transportation systems.

Kielty et al. [13] developed an event driven neuromorphic approach to support fast yawning recognition. Their system consumed minimal computational resources and provided high reaction speed required for safety critical automotive systems. They suggested future extension for tackling complex mouth coverings and occlusions.

Birajdar et al. [14] proposed a feature based method measuring mouth aspect changes to distinguish yawning from normal mouth movements, noted for its simplicity but showing reduced performance in dim light and when drivers wear masks or scarves.

Essahraui et al. [15] created a CNN model capable of real time fatigue behavior recognition while maintaining adaptability to diverse individuals and background lighting. Their performance indicated strong suitability for implementation in Advanced Driver Assistance Systems (ADAS) used in commercial automobiles.

Lyu et al. [16] developed a multi granularity deep network that utilized sequential behavioral cues instead of relying on single frame facial information. Their system monitored alertness continuously throughout extended driving durations and improved prediction stability during the early stages of drowsiness. Their contribution created opportunities to integrate long term behavioral context into advanced driver monitoring architectures.

A significant effort toward enhancing yawning detection accuracy involved leveraging ensemble learning techniques. Salman et al. [17] trained multiple convolutional neural networks on the YawDD dataset to better classify fatigue cues across different drivers and head poses. Their ensemble strategy yielded improved generalization and reduced misclassification rates compared to single network models. The study also stressed the need for broader datasets to widen applicability in realistic driving scenarios.

Salman et al. [17] utilized an ensemble of multiple CNN models trained on the YawDD dataset to enhance yawning detection accuracy across different drivers and head poses. Their ensemble approach improved generalization and reduced misclassification when compared with individual networks. They emphasized that expanding to broader and more diverse datasets remained essential for improving applicability in real world driving scenarios.

Gupta et al. [19] reviewed yawning detection methodologies used in driver drowsiness monitoring. Their work compared traditional mouth feature extraction techniques with recent deep learning solutions such as CNN based models and hybrid neural architectures. The authors observed that although deep learning enhanced classification performance, challenges including low visibility conditions, head pose variations, and partial facial occlusions still restricted reliability in realistic vehicle environments. They recommended the creation of stronger and more diverse datasets, along with the integration of yawning cues and physiological indicators, to achieve practical and highly accurate fatigue monitoring in intelligent transportation systems.

Pandey et al. [20] worked on a deep CNN based computer vision system to detect yawning and eye state changes in real time. Their methodology used image augmentation techniques to handle lighting variation and driver head pose changes. Their results achieved above 91% yawning recognition and around 94% eye state classification accuracy. They demonstrated that integrating a robust vision pipeline with fast alert mechanisms helped reduce fatigue related driving risks in real world transportation environments.

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

System design in this project aims to create a compact and reliable real time driver monitoring system that can accurately detect yawning as an early sign of drowsiness. This project focuses on practicality and efficient performance in real driving conditions by integrating key components such as camera input, deep learning based detection, embedded processing, and IoT alert mechanisms. Together, these elements work seamlessly to provide fast and accurate warnings, helping reduce fatigue related accidents and improving overall road safety.

3.2 DEVELOPMENT ENVIRONMENT

3.2.1 HARDWARE SPECIFICATIONS

This section outlines the hardware components required for training and fine tuning the YOLO based yawning detection model. These components support high performance computation, deep learning experimentation, and efficient processing of large scale image datasets during the development phase. The major hardware units used for both model training and embedded deployment are listed in Table 3.1 Hardware Specifications.

3.2.2 SOFTWARE SPECIFICATIONS

The software specifications are designed to support both the development and deployment of a real time yawning detection system that operates efficiently on embedded hardware while enabling accurate deep learning inference. The selected tools and libraries provide an optimal environment for training the YOLO model, processing video streams, implementing IoT based alerts, and running the system smoothly on the Raspberry Pi. These components ensure a balance between computational efficiency, portability, and ease of deployment for real world driver monitoring applications. The complete software configuration used for this system is listed in Table 3.2 Software Specifications.

Table 3.1 Hardware Specifications

Component	**Specifications**
Processor	Intel Core i7
RAM	16GB or above (DDR4 RAM)
GPU	NVIDIA T4 (DL model training)
Storage	256GB SSD
Processor frequency	2.6 GHz or above
Processor	Raspberry Pi 4 Model B
RAM	4GB / 8GB LPDDR4
Camera	Raspberry Pi Camera Module or USB HD Camera
Alerts Hardware	Buzzer / Speaker Module
Storage	32GB–128GB MicroSD Card
Processing Frequency	1.5 GHz Quad Core
Power Supply	5V, 3A USB-C Adapter

Table 3.2 Software Specifications

Programming Language	Python 3.x
Deep Learning Framework	PyTorch / TensorFlow
IDE	VS Code, Jupyter Notebook
Cloud platform	Kaggle or Google Colab
Model Development	YOLOv8
Computer Vision Library	OpenCV
Embedded Platform OS	Raspberry Pi OS (Linux based)
IoT / Alerts	Telegram Bot API, Python Requests

The selected software configuration provides a flexible and efficient environment for developing and deploying the real time yawning detection system. It supports deep learning model training, video processing, and IoT alert integration while remaining lightweight enough to run smoothly on embedded hardware and resource constrained setups.

3.3 ARCHITECTURE DESIGN

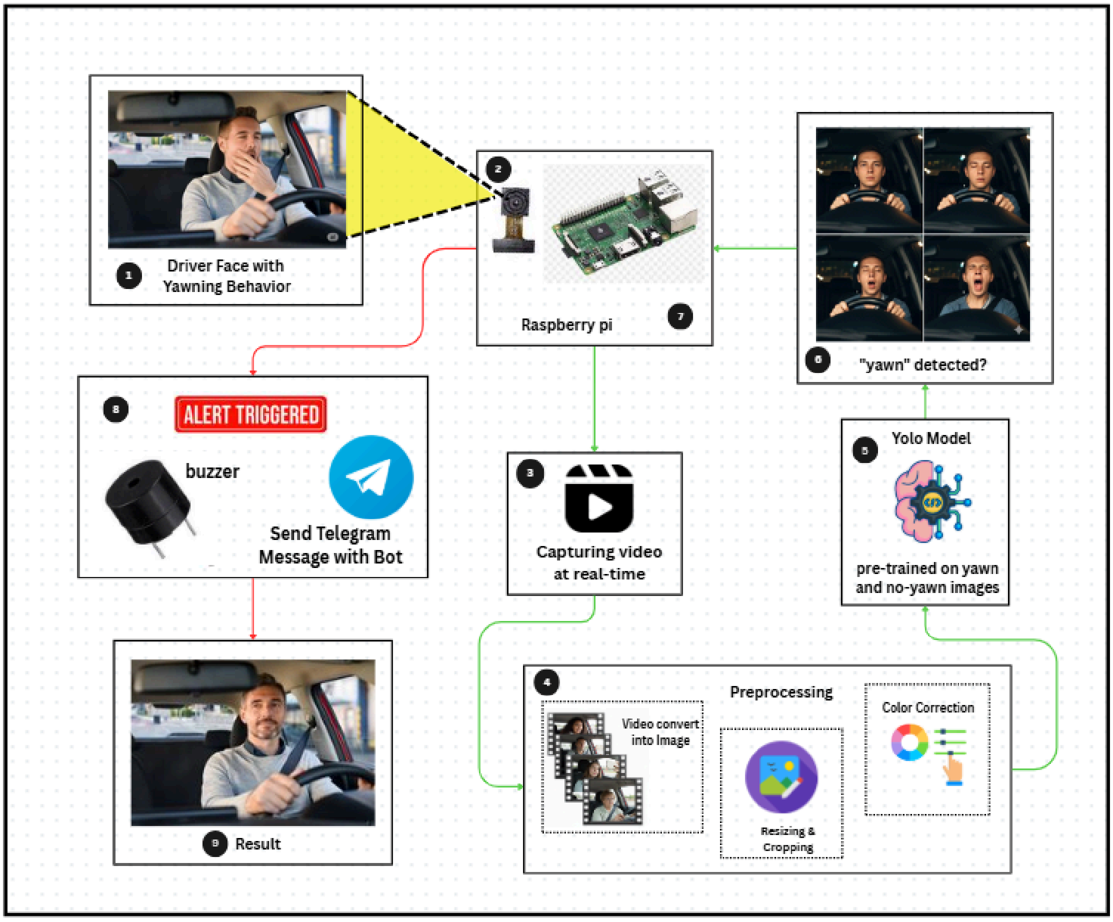

Fig 3.1 Architecture Design of the Proposed System

Figure 3.1 illustrates the proposed real time yawning detection system designed to enhance driver safety by monitoring early signs of drowsiness using deep learning and IoT based alert mechanisms. The system uses a YOLO-based model deployed on a Raspberry Pi to detect yawning behavior directly from live video captured inside the vehicle. By integrating lightweight vision algorithms with embedded processing, the system provides a practical and reliable solution for preventing fatigue related accidents, especially in long distance driving environments or regions where advanced driver assistance technologies are not widely accessible..

The proposed system consists of the following core modules:

1. **Data collection and preprocessing:** The system begins by capturing continuous video frames of the driver through a camera mounted on the dashboard. These frames are preprocessed by converting the video stream into individual images, resizing and cropping them for uniformity, and performing color correction to ensure stable detection under varying lighting conditions. This preprocessing prepares the input for consistent real time performance and helps the model handle natural variations such as shadows, brightness changes, and camera angles inside the vehicle.
2. **Real Time Video Capture Using Raspberry Pi:** The Raspberry Pi receives the video stream from the connected camera module and processes it frame by frame. Its low power consumption and compact size make it well suited for in vehicle deployment. The device continuously feeds each preprocessed frame into the YOLO detection model to quickly identify signs of yawning.
3. **YOLO-Based Yawn Detection Model:** A lightweight YOLO model, pre trained on annotated yawn and no yawn images, is used to detect whether the driver is yawning in each frame. The model examines facial regions, especially the mouth, to differentiate between normal and yawning states. Its fast inference speed and high accuracy allow it to operate smoothly on the Raspberry Pi, making it ideal for real time embedded applications.
4. **Decision and Alert Mechanism:** If the model detects a yawn, the system triggers immediate alerts to warn the driver. A buzzer connected to the Raspberry Pi provides an audible warning, helping the driver regain focus. Additionally, a Telegram Bot is integrated to send remote notifications to guardians, fleet managers, or monitoring personnel when repeated yawning is detected. This ensures both immediate and remote safety responses.

3.3.1 ACTIVITY DIAGRAM – IMAGE ENHANCEMENT AND MODEL TRAINING

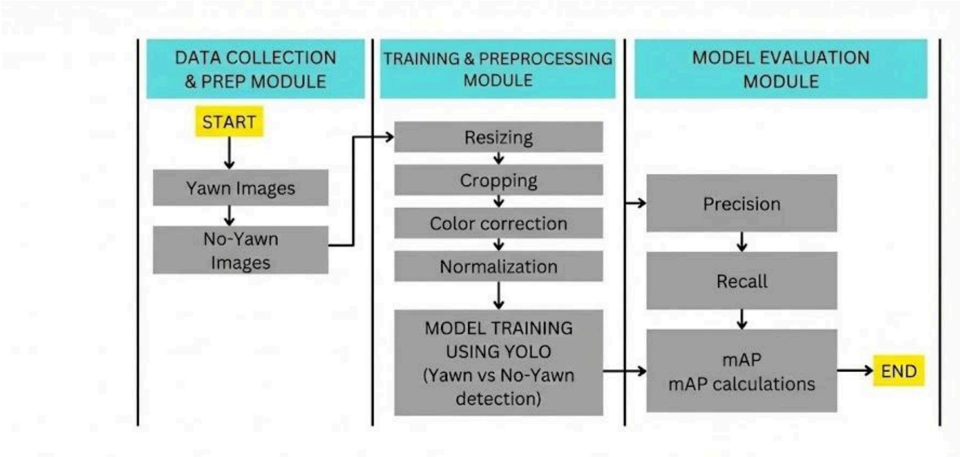

Fig 3.2 Activity Diagram – Image Enhancement and Model Training For Yawn Detection

Figure 3.2 illustrates the complete workflow from data acquisition to the final model training stage for the yawning detection system. The process begins with the collection of yawn and no yawn images from various sources, forming a diverse dataset that reflects different lighting conditions, facial orientations, and driver appearances. This variety ensures the model learns to generalize effectively in real driving environments.

Once the images are collected, the dataset undergoes preprocessing to make it suitable for training. This includes operations such as frame extraction, resizing, cropping, color correction, and normalization, ensuring that all input images are consistent in format and quality. Preprocessing helps reduce noise, standardize the inputs, and improve overall learning efficiency.

After preprocessing, the system moves to feature extraction using the YOLO model. YOLO identifies key facial regions particularly the mouth and face areas and produces bounding boxes that act as the foundation for detecting yawning behavior. This step enables the model to focus on the most relevant regions in each image, improving detection accuracy.

The extracted features are then used for model training, where the YOLO based detector learns to classify each instance as a yawn or no yawn. Throughout the training process, the model refines its ability to distinguish subtle differences in mouth openness and facial expressions that characterize yawning. The training loop progressively enhances detection precision to ensure reliable performance in real time conditions.

In the final stage, model evaluation is conducted to determine the system's effectiveness. Key performance metrics such as precision, recall, and mean Average Precision (mAP) are calculated to assess accuracy and robustness. This evaluation helps identify areas for improvement and ensures that the trained model is capable of performing reliably when deployed in actual driver monitoring scenarios.

3.3.2 ACTIVITY DIAGRAM - USER INPUT CLASSIFICATION

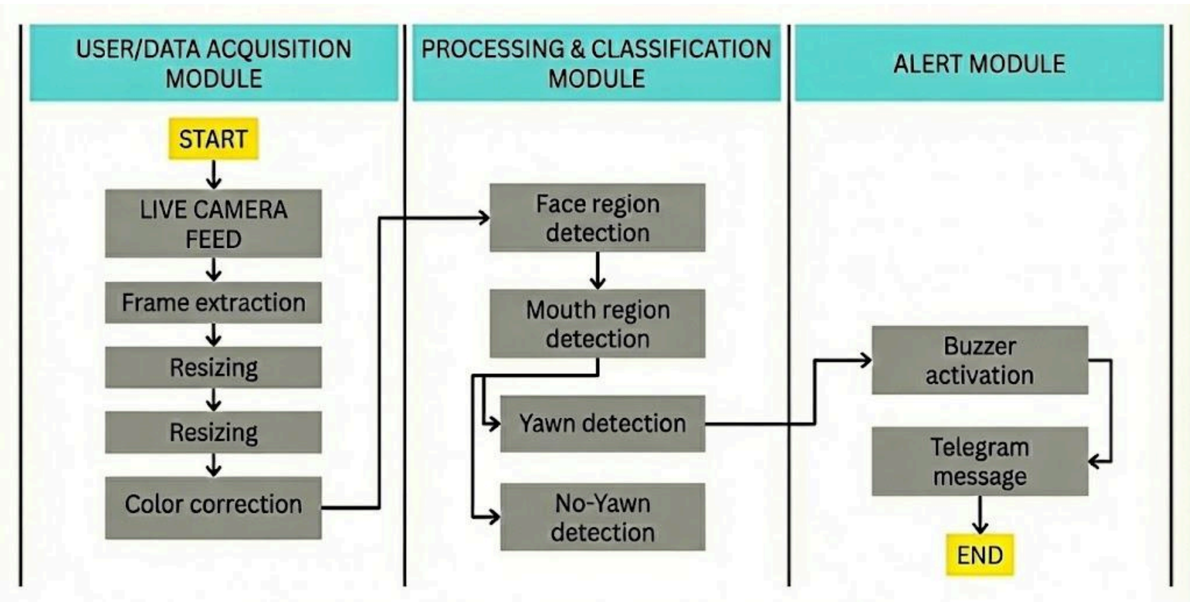

Fig 3.3 Activity Diagram - User Input Classification For Yawn Detection

Figure 3.3 illustrates the complete process beginning with the system capturing live video input from the driver through an in vehicle camera. This real time feed forms the core input through which the system continuously monitors the driver's facial behavior. Each frame extracted from the video stream is then forwarded to the detection pipeline for further analysis.

Once the input is captured, the system performs feature extraction using the YOLO model, where key facial regions particularly the mouth and surrounding areas are detected. This step allows the model to isolate relevant features that indicate yawning behavior. YOLO's real time detection capability ensures efficient processing even under varying lighting, angles, and environmental conditions inside the vehicle.

After the features are identified, the system moves into the classification stage where each frame is evaluated to determine whether the driver is yawning or not. This stage applies the trained yawn classification model to differentiate between yawn and no yawn patterns with high accuracy. The classification outcome is generated continuously as the video stream is processed.

Following the classification, the system triggers an alert mechanism when a yawn is detected. The alert can involve sounding a buzzer or automatically sending a Telegram notification to a connected device. This ensures the driver receives immediate feedback prompting them to regain alertness, thereby reducing the risk of fatigue related incidents.

Finally, the system maintains a real time monitoring loop, ensuring continuous observation of the driver's state. This streamlined, automated flow helps promote safer driving conditions by enabling fast, consistent detection of early drowsiness indicators.

CHAPTER 4

METHODOLOGY

4.1 DRIVER DROWSINESS DETECTION USING DEEP LEARNING MODELS

The initial phase of this research involved experimenting with conventional deep learning approaches, primarily CNNs, for the classification of yawning versus non yawning facial images. CNNs are widely used in facial expression analysis because they can automatically learn hierarchical features such as mouth shape, face orientation, and motion cues. In this phase, several CNN based models were tested to understand their capabilities and limitations in detecting yawns under real world driving conditions.

A custom dataset containing labeled Yawn and No Yawn images was prepared, and all images were preprocessed through resizing, normalization, frame extraction, and augmentation to simulate variations in lighting, pose, and motion blur. Hyperparameters such as learning rate, batch size, epoch count, and regularization techniques were optimized to achieve the best performance. The overall training process for the CNN based models was carried out efficiently from data preprocessing to final inference.

The following subsections describe each traditional deep learning model evaluated in this study, discussing their architecture, advantages, and performance in yawning classification.

Baseline CNN: A basic CNN model was developed as an initial benchmark, consisting of a few convolutional layers for extracting mouth region features and dense layers for binary classification (Yawn vs No Yawn). This simple architecture successfully captured broad patterns such as open mouth versus closed mouth frames. However, it struggled with complex real world conditions, including partial occlusions, changing illumination, and subtle facial movements. Although the baseline CNN achieved moderate accuracy, it served primarily as a reference point for evaluating more advanced techniques.

MobileNetV2: MobileNetV2 was included in the analysis because of its lightweight design suitable for real time inference on embedded devices such as the Raspberry Pi. Its depthwise separable convolutions significantly reduced computation while maintaining reasonable accuracy. The model performed better than the baseline CNN, especially for images captured in low light conditions or with slight motion blur. Despite this improvement, MobileNetV2 still lacked sufficient precision when detecting small mouth movements during the early stages of yawning.

VGG16: VGG16 provided deeper feature extraction capabilities due to its multiple convolutional layers. It performed noticeably better in distinguishing early signs of yawning. However, its large number of parameters resulted in slow inference speeds, making it unsuitable for use on Raspberry Pi where real time detection is required.

ResNet50: ResNet50 used residual connections to preserve gradient flow, enabling deeper feature learning. It consistently performed well even when the driver’s head was slightly tilted or partially occluded. Although ResNet50 showed high accuracy, the computational complexity made it too slow for deployment on embedded hardware.

4.2 ADVANCED YAWN DETECTION MODEL FOR DRIVER DROWSINESS

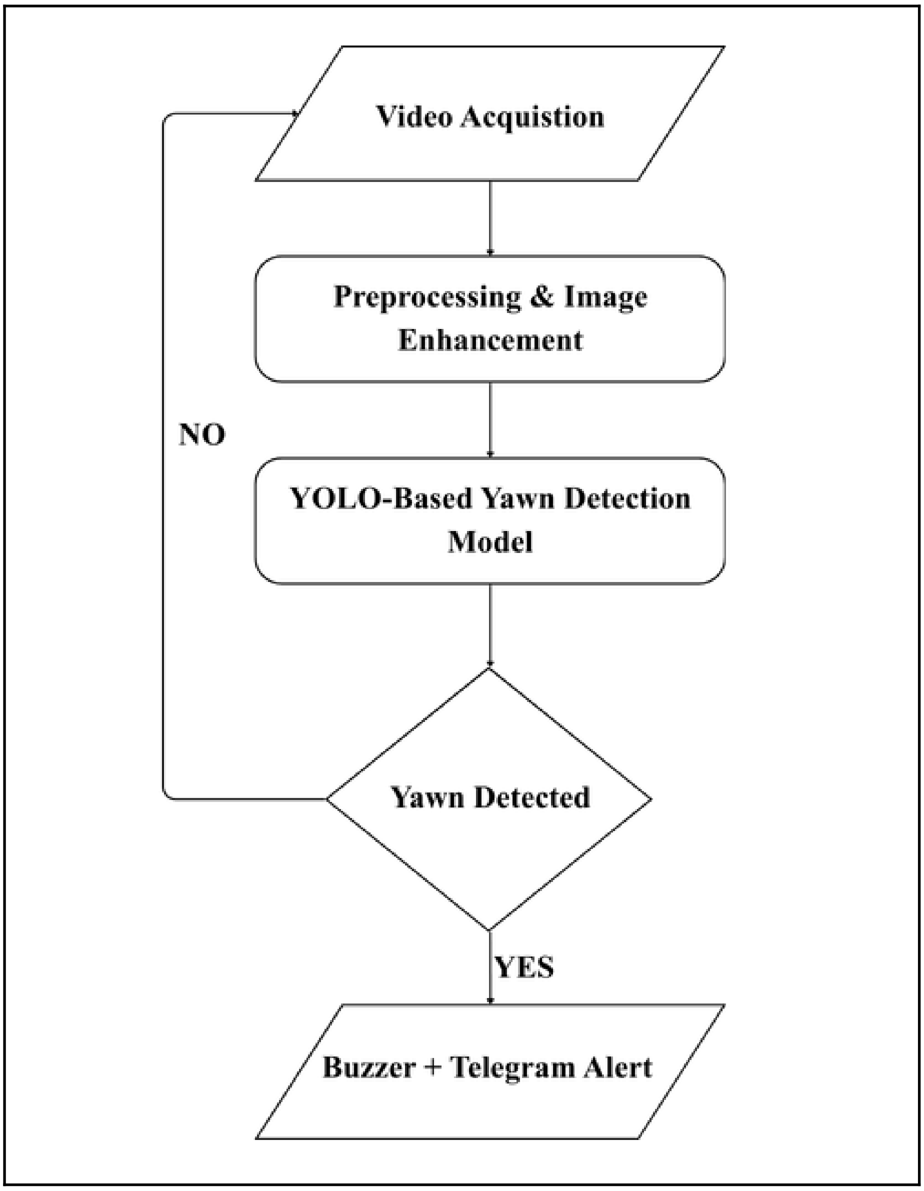

Fig 4.1 Data Flow Diagram of the overall proposed system

The advanced data driven model in yawn detection for driver drowsiness monitoring focuses on combining a YOLO based deep learning approach with embedded processing on Raspberry Pi and IoT enabled alert mechanisms. The workflow is designed to process live in vehicle video streams efficiently and convert them into meaningful safety alerts. Figure 4.1 illustrates the overall data flow, where incoming frames from the driver facing camera are preprocessed through resizing and color adjustments before being passed into the YOLO model for accurate face and mouth region detection. The model then classifies each frame as Yawn or No Yawn, enabling the system to identify early signs of fatigue. When yawning is detected, the alert module activates a buzzer and optionally sends a Telegram notification, ensuring quick intervention during drowsy driving events. This integrated approach provides a fast, reliable, and automated solution for improving road safety, especially in long distance travel and environments lacking modern monitoring technologies.

4.2.1 DATA PREPROCESSING AND AUGMENTATION

Dataset Description

The Yawn Detection dataset, publicly available through Roboflow Universe (Project ID: yay-vtkyd), is a large, curated collection of annotated facial images specifically designed for identifying yawning behavior in real world scenarios. It contains a diverse range of driver like facial expressions categorized into two classes yawn and no yawn allowing effective binary classification of alertness states. The dataset comprises over fifteen thousand samples collected under varying lighting conditions, camera angles, facial orientations, and background environments, ensuring that the model trained on it can generalize reliably to practical driving situations. Each image is accompanied by bounding box annotations focused primarily on the mouth region, providing precise localization for supervised learning and enabling detection models to distinguish yawning from other transient facial movements. The dataset's diversity, high quality labels, and broad coverage of environmental variations make it a valuable resource for training deep learning models aimed at real time fatigue detection. Owing to these characteristics, the Yawn Detection dataset is particularly well suited for developing robust, embedded yawn recognition systems that must perform consistently under the dynamic conditions of in vehicle monitoring.

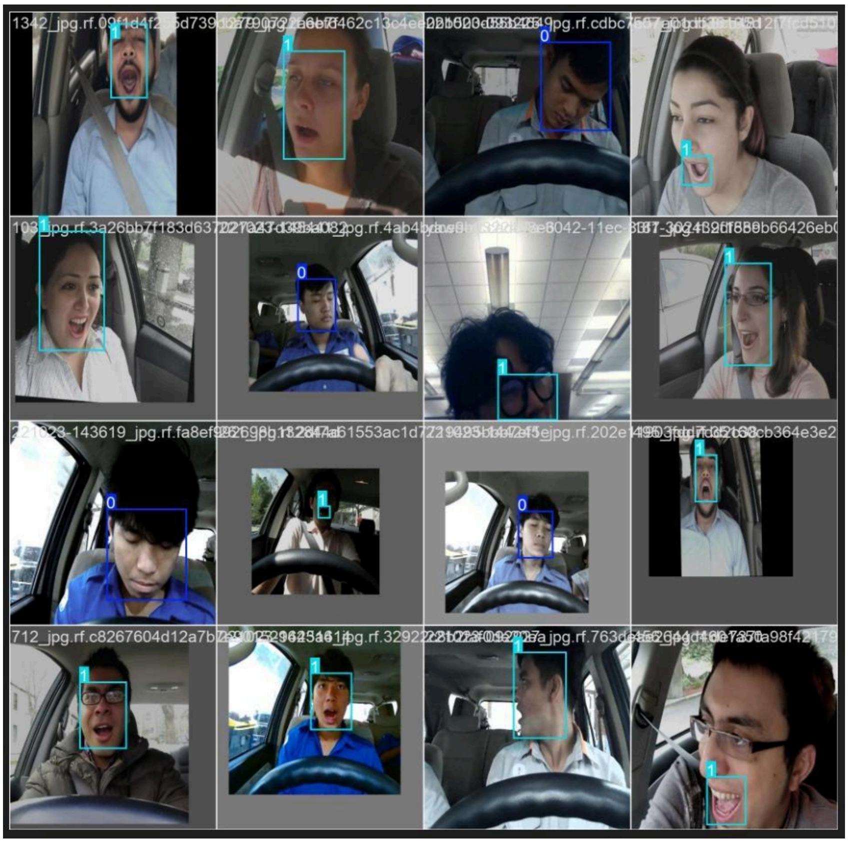

Fig 4.2 Sample of the Yawn Detection Dataset

Dataset Preparation

A diverse dataset of facial images was utilized to train the proposed yawn detection model. All images were sourced from the Roboflow Universe Yawn Detection repository and contained annotated bounding boxes for the yawn and no yawn classes. During the preparation phase, images were resized to the fixed resolution required by YOLOv8n (640 × 640), ensuring uniform compatibility across the training, validation, and testing sets. Pixel normalization was applied to stabilize feature extraction, while low quality or mislabeled samples were removed to ensure dataset consistency. This preprocessing stage ensures that the model receives standardized inputs, thereby improving the accuracy and reliability of the subsequent detection tasks.

Data Augmentation

To improve robustness and prevent overfitting, several augmentation techniques were applied to the dataset. Geometric augmentations such as horizontal flipping, rotation, and random scaling simulate realistic variations in driver head movement and camera angles. Brightness and contrast adjustments emulate different illumination conditions inside the vehicle, including shadows and nighttime driving. Noise reduction and smoothing filters were used to minimize distortions caused by low resolution sensors or vehicle motion. These enhancement techniques collectively ensure that the model can reliably detect yawning behavior under varied and challenging driving environments.

Fig 4.3 Augmentation of Yawn Dataset

4.2.2 IMAGE ENHANCEMENT AND PREPROCESSING PIPELINE

Real Time Image Enhancement

During real time monitoring, the quality of captured video frames can vary significantly due to changing lighting conditions, camera vibrations, shadows, and driver movements. To ensure optimal detection accuracy, each frame undergoes an enhancement and preprocessing pipeline before being fed into the YOLOv8n model. This enhancement pipeline improves clarity in the mouth region, helping the model distinguish between yawning and non yawning frames more reliably.

Geometric Normalization:

All incoming frames are resized to the YOLOv8n compatible resolution of 640 × 640, and pixel values are normalized to a fixed scale. This standardization reduces input variability, stabilizes the detection process, and ensures uniformity across different camera devices and lighting conditions.

Illumination and Noise Correction:

Brightness and contrast adjustments are applied to enhance mouth visibility under fluctuating in cabin lighting, including low light environments or sudden exposure to sunlight. Noise reduction filters are also used to minimize distortions caused by sensor noise or vehicle vibration, enabling the model to detect yawning patterns with improved clarity and reliability.

Yawn Detection Using YOLO:

Require: Image dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i is an image and y_i is the label (Yaw / No Yaw).

Require: Pre trained YOLO model $M_{pretrained}$,optimizer B,and loss functions L

Require: Hyperparameters: learning rate α, batch size B, number of epochs E.

Step 1: Preprocessing

1. Resize all images x_i to the required input size.
2. Normalize pixel values to range $[0, 1]$
3. Apply augmentation (brightness change, rotation, flipping, motion blur).
4. Convert video frames into image sequences if necessary.

Step 2: Data Splitting

5. Split D into training set D_{train}, validation set D_{val}, and test set D_{test}.

Step 3: Model Setup

6. Load YOLO with pretrained weights (e.g., COCO).
7. Modify the final detection layer to output two classes (Yawn, No Yawn).
8. Set anchor boxes and detection thresholds.
9. Initialize optimizer and learning rate schedule.

Step 4: Training the Model

10. For epoch e = 1 to E do

11. Forward → Loss → Backprop → Update → Repeat

12. end for

13. Evaluate $M_{pretrained}$ on D_{train} and record metrics.

14. Apply early stopping if validation performance stagnates.

Step 5: Inference

16. Import a live frame x_{live} from a camera.

17. Predict yaw status using the trained model $M_{pretrained}$.

4.2.3 YOLOv8n ARCHITECTURE FOR YAWN DETECTION

YOLOv8n is the primary model used in this project for detecting yawning behavior in real time. The model is a lightweight and fast object detection architecture designed for edge devices, making it suitable for deployment on Raspberry Pi. Unlike traditional CNN based classifiers, which only classify entire images, YOLOv8n performs both object localization and classification, allowing the system to accurately detect the mouth region and determine whether the driver is yawning. As illustrated in Figure 4.6, the YOLOv8n detection pipeline begins by taking each video frame captured by the camera and preprocessing it to the required resolution (640×640). The model then processes the image using its feature extraction backbone and detection head to generate bounding boxes along with confidence scores for the two classes used in this project yawn and no yawn. The YOLOv8n architecture internally handles all required steps such as feature extraction, prediction, and filtering to produce the final detection output. The resulting prediction is then passed to the Raspberry Pi, which interprets the outcome and triggers the appropriate alert mechanism when a yawning event is detected. Due to its optimized design, the model performs efficiently even on low power devices and remains robust under various real world driving conditions such as varying illumination, head movements, and partial occlusions.

- Varying cabin lighting
- Driver head movement
- Partial occlusions
- Motion blur while driving
- High speed and accuracy enable real time driver monitoring.

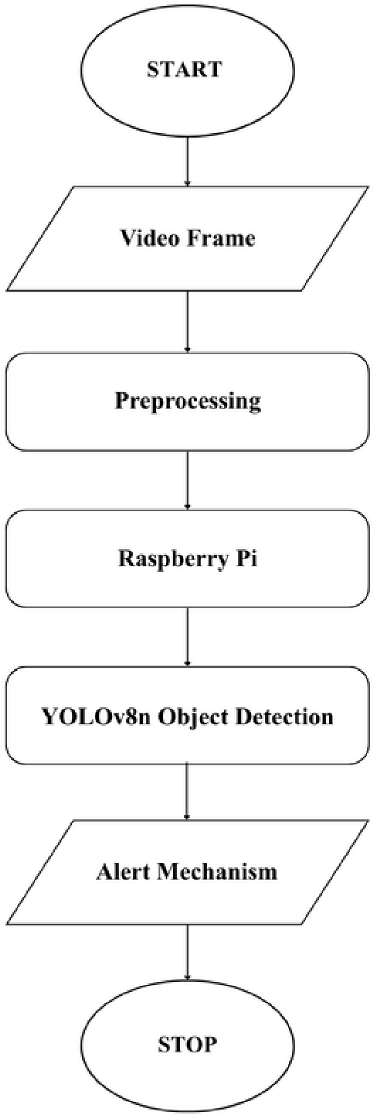

Fig 4.4 Data Flow Diagram for Yawn Detection using YOLOv8n

4.2.4 MULTI CHANNEL ALERT GENERATION

Once the YOLOv8n model identifies a yawning event with sufficient confidence, the system proceeds to activate the alert mechanism designed to notify the driver and associated stakeholders. As shown in Figure 4.7, the alert pipeline operates immediately after the detection stage and provides both local and remote notifications to ensure timely intervention during early signs of fatigue.

The system first triggers an in cabin buzzer, providing an audible warning that prompts the driver to regain focus or take a brief rest. In addition to this local alert, a Telegram notification is automatically generated through the integrated API. This message includes essential details such as the time of detection, system status, and the confidence level of the prediction. Such remote alerts are especially useful in fleet monitoring, logistics operations, and supervised environments where real time driver condition tracking is required.

The multi channel alert system enhances the accessibility and effectiveness of the fatigue detection process by ensuring that critical warnings are communicated promptly and reliably. This dual alert strategy supports proactive safety intervention and contributes to reducing the risk of fatigue related driving incidents.

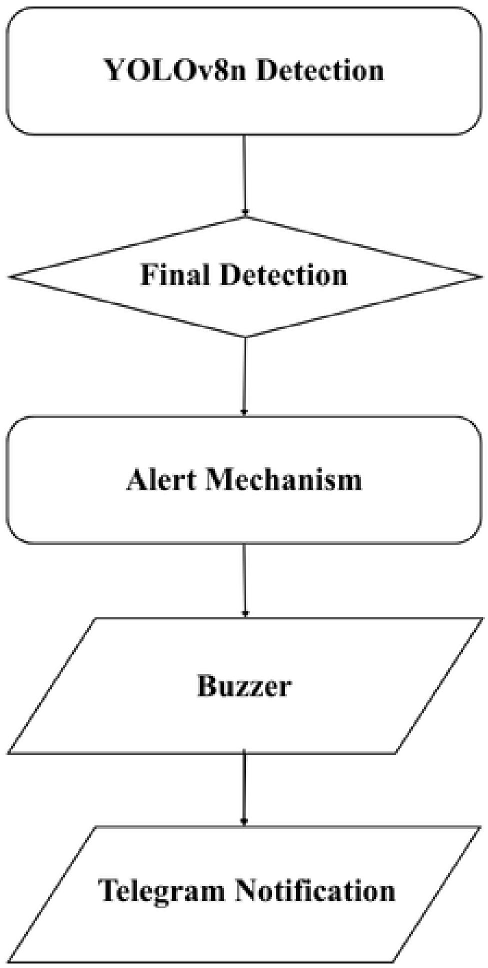

Fig 4.5 Flow Diagram for Yawn Detection using YOLOv8n

CHAPTER 5

RESULTS

5.1 PERFORMANCE ANALYSIS OF THE YOLO BASED YAWN DETECTION SYSTEM

In this project, the YOLO-based deep learning model is the primary and only architecture used for yawn detection. Since the goal is real time deployment on Raspberry Pi, no traditional CNN models were trained. Instead, the entire evaluation focuses on the performance, stability, and real time responsiveness of the YOLO model trained on the custom Roboflow Yawn/No Yawn dataset.

The dataset was preprocessed with resizing, normalization, and augmentation techniques such as brightness variation, motion blur simulation, rotation, and contrast enhancement. These steps ensured the model could handle real world driving conditions, including low lighting, face tilt, and partial occlusions. YOLO was trained on annotated images for both Yawn and No Yawn classes and evaluated based on its classification performance and inference behavior

Table 5.1 YOLO Model Evaluation Metrics

Class	**Precision**	**Recall**	**mAP@50**
No Yawn	0.999	1.0	0.995
Yawn	0.765	0.775	0.84
Overall	0.854	0.893	0.913

The YOLOv8n model demonstrated strong overall performance, with exceptionally high scores for the no yawn class and competitive performance for the yawn class which is more visually complex due to partial mouth openings, talking movements, and lighting variations. The overall mAP@50 of 0.913 confirms reliable detection behavior suitable for real time driver monitoring.

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	1197	1197	0.882	0.878	0.918	0.825
no_yawn	469	469	0.999	1	0.995	0.963
yawn	728	728	0.765	0.755	0.84	0.687

Figure 5.1 YOLO Model Validation Metric

5.2 REAL TIME PERFORMANCE OF YOLO MODEL

To assess the real time capability of the yawn detection system, YOLOv8n inference performance was tested on both a laptop/PC and a Raspberry Pi 4 using the same trained model. This comparison validates that the proposed system maintains consistent detection accuracy across different hardware platforms. As shown in Table 5.2, the Raspberry Pi 4 achieves 32 FPS with a latency of 148 ms, which is very close to the performance observed on the laptop (30 FPS and 140 ms latency). Importantly, there is no reduction in detection accuracy, demonstrating that the Raspberry Pi 4 can reliably support real time in vehicle monitoring and timely alert generation during continuous driver observation.

Table 5.2 YOLOv8n Real Time Inference Comparison on Different Devices

Platform	**FPS**	**Latency (ms)**	**Detection Accuracy**	**Notes**
Laptop / PC	30	140	Same as trained model	Fast, smooth inference
Raspberry Pi 4	32	148	Same as trained model	Stable real time performance

5.3 TRAINING AND VALIDATION PERFORMANCE OF YOLOv8n

The YOLOv8n model was trained on the curated Yawn or No Yawn dataset using a composite loss function that combines localization, classification, and distribution focal components. The total loss used during optimization is expressed as:

$$L_{total} = \lambda_1 L_{box} + \lambda_2 L_{cls} + \lambda 3_3 L_{dfl} \qquad (5.1)$$

The Box Loss measures the geometric overlap between predicted and ground truth bounding boxes:

$$L_{box} = 1 - GIoU(B_p, B_t) \qquad (5.2)$$

The Class Loss is computed using the cross-entropy formula:

$$L_{cls} = -\frac{1}{C \sum Y_c \, log(y^c)} \qquad (5.3)$$

The Distribution Focal Loss (DFL) refines bounding box regression by modeling offsets as discrete distributions:

$$L_{dfl} = \sum (i = 1 \, to \, n) \, w_i \mid p_i - g_i \mid \qquad (5.4)$$

The YOLOv8n model was trained on the curated Yawn or No Yawn dataset using its standard composite loss function. The learning behavior during training was evaluated by analyzing the loss and validation curves over 50 epochs. The results show that the model converged smoothly with no signs of overfitting or instability.

As presented in Figure 5.2 and Figure 5.3, the training and validation loss curves for all components Box Loss, Class Loss, and Distribution Focal Loss (DFL) decrease consistently across epochs. The validation losses closely follow the training losses without any sharp spikes or divergence, indicating strong generalization. The continuous reduction in Box Loss reflects improved bounding box localization accuracy, the rapid decline in Class Loss confirms stable discrimination between the yawn and no yawn classes, and the gradual drop in DFL demonstrates increasing precision in fine localization of the mouth region.

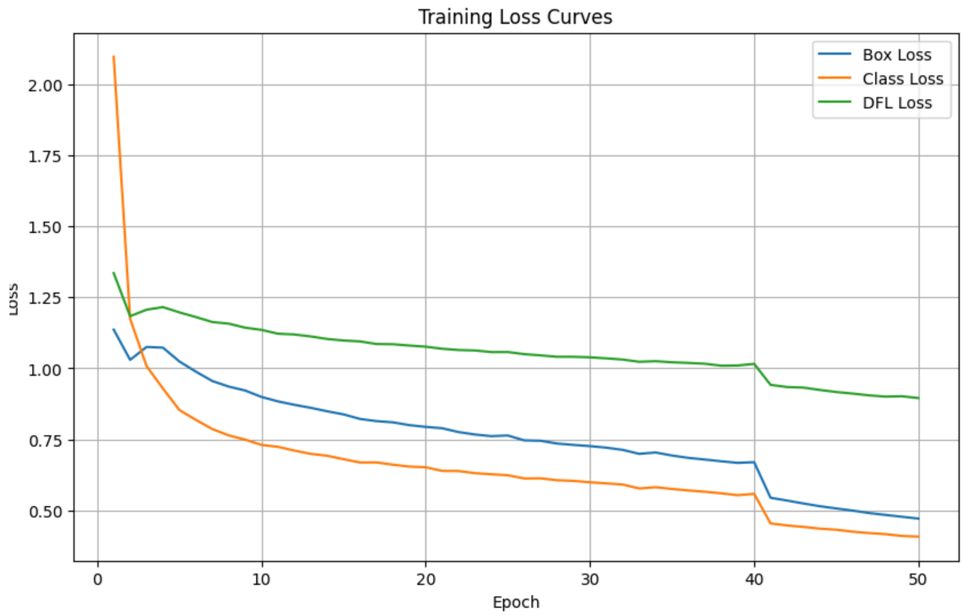

Fig 5.2 Training Loss Curves

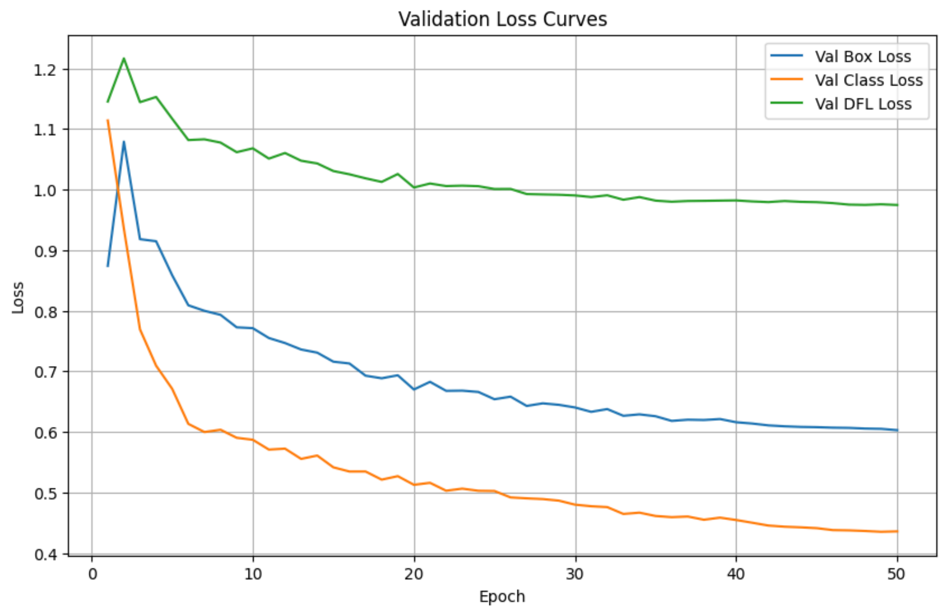

Fig 5.3 Validation Loss Curves

In addition to loss trends, the evolution of accuracy related metrics provides deeper insight into the performance of the YOLOv8n model. As shown in Figure 5.4, Precision stabilizes between 0.87 and 0.89, Recall increases toward 0.90, mAP@50 reaches approximately 0.91, and the stricter mAP@50–95 metric rises to around 0.82. These validation results are further supported by the model's overall validation accuracy of 87.55%, as reported during the YOLO evaluation stage. To complement the validation metrics, an additional image wise test evaluation was performed on a separate dataset, where the model achieved an exceptionally high accuracy of 99.84%, correctly predicting 627 out of 628 test images. This strong performance across both validation and test sets confirms that augmentation techniques such as rotation, brightness adjustment, motion blur, and contrast variation effectively reduced overfitting and enabled the model to generalize well under diverse real world driving conditions.

1. Precision

Precision represents the proportion of correctly detected yawns among all predicted yawns. This metric helps reduce false detections of normal mouth openings as yawns.

$$Precision = \frac{TP}{TP+FP} \tag{5.5}$$

2. Recall

Recall represents the proportion of actual yawns that were successfully detected. This metric helps ensure that genuine yawns are not missed during monitoring.

$$Recall = \frac{TP}{TP+FN} \quad (5.6)$$

3. Intersection over Union (IoU)

IoU measures the overlap between the predicted mouth bounding box and the ground truth bounding box. A higher IoU value indicates a more accurate localization of the mouth region.

$$IoU = \frac{Area(B_{pred} \cap B_{gt})}{Area(B_{pred} \cup B_{gt})} \quad (5.7)$$

4. Average Precision (AP)

AP summarizes the precision recall relationship into a single numerical score. This metric provides an overall measure of detection performance for the yawn class.

$$AP = \int_0^1 Precision(Recall)\, d(Recall) \quad (5.8)$$

5. mAP@50

mAP@50 calculates the mean Average Precision across all classes at an IoU threshold of 0.50. This metric provides a standard benchmark for evaluating detection quality in the yawn/no yawn model.

$$mAP@50 = \frac{1}{N} \Sigma\,(i = 1\ to\ N) AP_i^{(IOU = 0.5)} \quad (5.9)$$

5. Accuracy

Accuracy indicates the proportion of correct predictions made by the model across all classes. In the yawn detection project, this metric shows how well the system classifies frames as Yawn or No Yawn.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5.10}$$

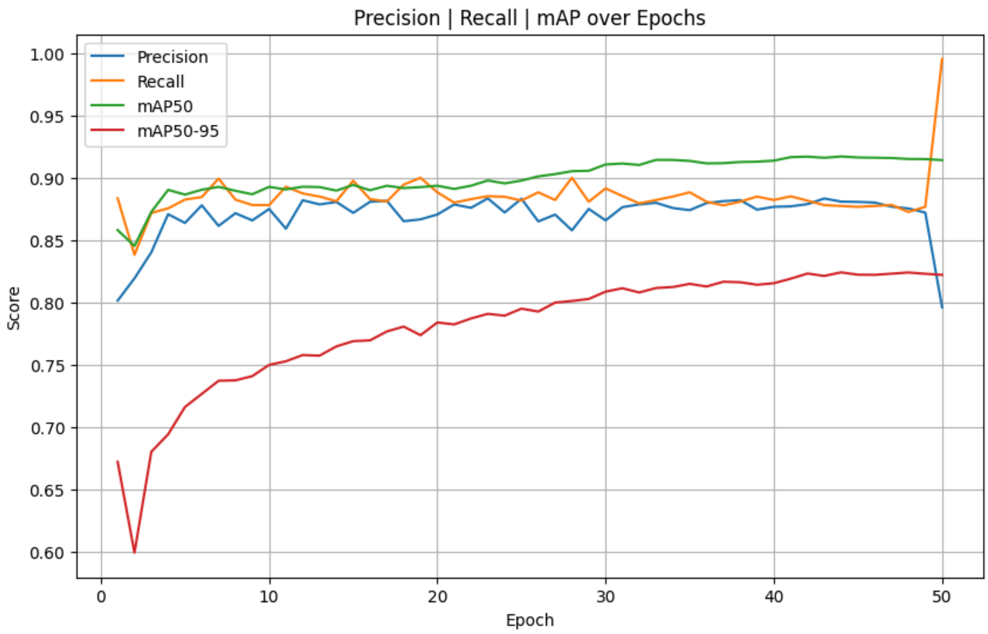

Fig 5.4 Precision, Recall, and mAP Trends Across Epochs

Overall, the training and validation performance clearly demonstrate that YOLOv8n successfully learned robust and discriminative features for detecting yawning behavior and maintained stable, reliable behavior suitable for embedded real time deployment on Raspberry Pi.

5.4 CONFUSION MATRIX EVALUATION

The performance of the YOLOv8n model was evaluated using a normalized confusion matrix, as shown in Figure 5.5, which visually illustrates how effectively the model distinguishes between the yawn, no yawn, and background classes. The matrix displays very high diagonal values for all categories, indicating strong classification consistency. The no yawn class achieves perfect prediction accuracy, while the yawn class also shows excellent recognition with only minimal misclassification, mainly occurring during partial mouth openings, speaking movements, or low light conditions. Only a very small fraction of yawn samples were incorrectly categorized as background. Overall, the results confirm extremely high true positive rates for both yawn and no yawn classes, minimal false positives and false negatives, and strong generalization across variations in lighting, facial orientation, and motion making the YOLOv8n model highly reliable and well suited for real time driver monitoring applications.

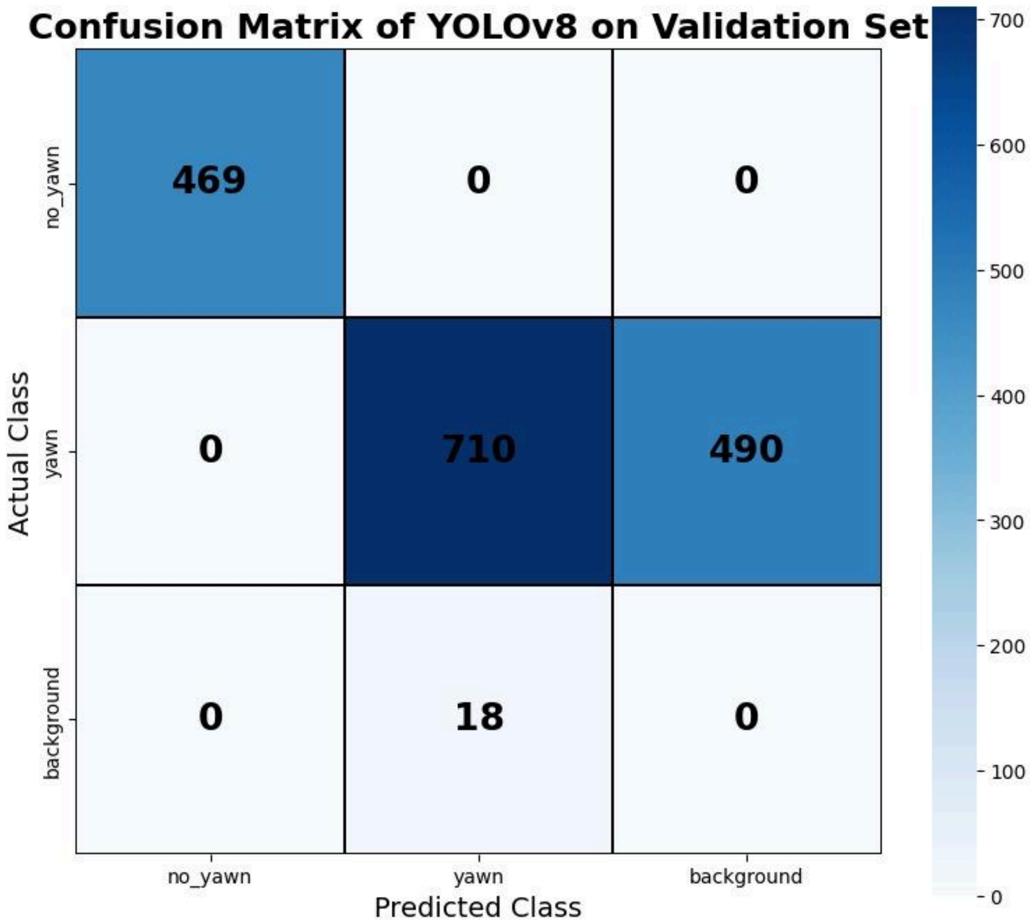

Fig 5.5 Confusion Matrix of YOLOv8n on Validation Set

5.5 CONFIDENCE AND LABEL DISTRIBUTION ANALYSIS

To gain deeper insight into model behavior, confidence score distribution analysis was performed for each class, as illustrated in Figure 5.6. The confidence distribution reveals a clear separation between the two classes, with the no yawn class exhibiting consistently high confidence values (median ≈ 0.93), indicating the model's strong certainty in identifying non yawning faces. In contrast, yawn predictions display a broader confidence range (typically 0.55-0.75) due to natural variations in mouth opening, facial posture, and lighting conditions. This wider spread is expected, as yawning gestures differ significantly across individuals. Overall, the confidence score analysis validates the robustness of the YOLOv8n model and demonstrates reliable detection performance even in challenging real world driving scenarios.

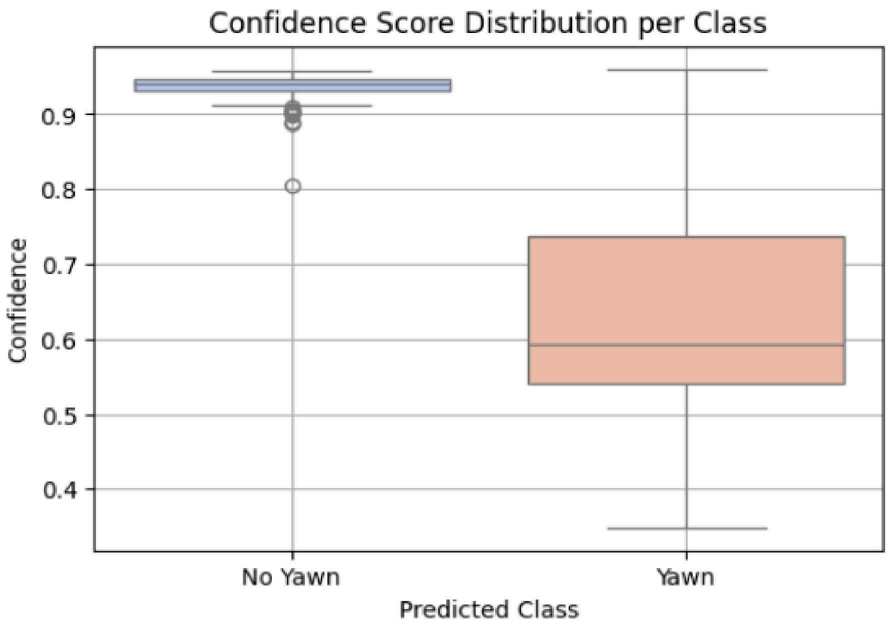

Fig 5.6 Confidence Score Distribution per Class

5.6 YAWN DETECTION USING TRAINED YOLOV8N MODEL

During the testing phase, the trained YOLOv8n model is evaluated on recorded facial video data to identify yawning instances with high accuracy. The model processes each frame and predicts bounding boxes around the mouth region along with yawn confidence scores. Detection results are visualized in two formats: a multi frame grid layout that showcases both yawn and non yawn predictions under different conditions as shown in Figure 5.7, and individual single frame detection outputs that display precise yawn localization with confidence values as shown in Figure 5.8. These visual results enable validation of the model's classification performance under varying lighting, facial posture, and visibility conditions. These observations ensure that the model has learned reliable yawn related visual features required for accurate driver drowsiness detection.

These images show multiple frames processed by the YOLOv8n model, demonstrating the system's ability to classify both yawning and non yawning states across different positions and lighting conditions.The grid layout highlights consistent detection accuracy across varied scenarios.

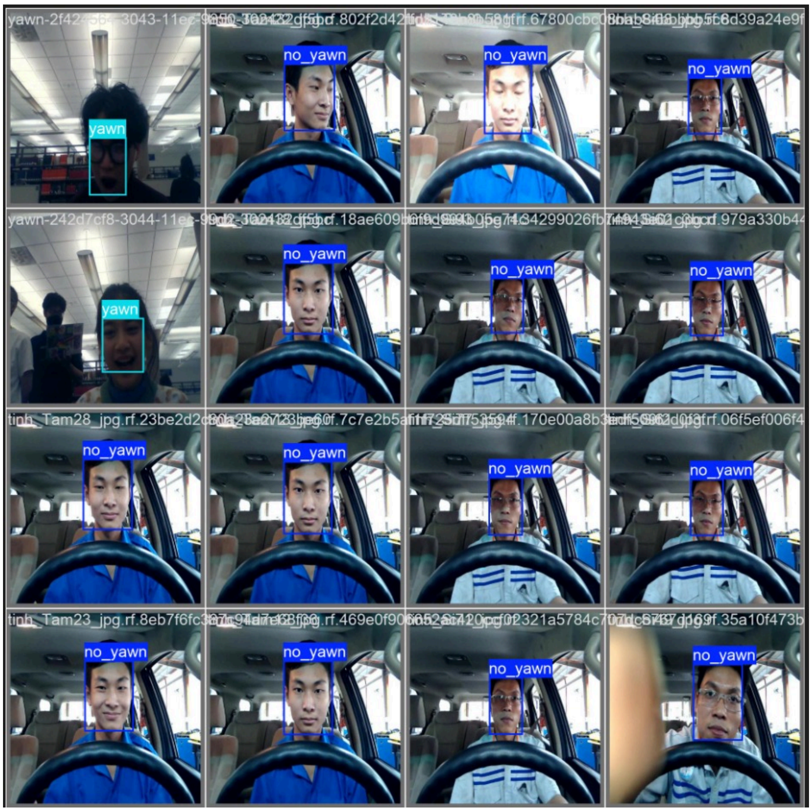

Fig 5.7 Grid Based Yawn/No Yawn Detection Results

This output illustrates a clear single frame detection where the model identifies a yawning event with a high confidence score. The example demonstrates precise localization of the mouth region during real time inference.

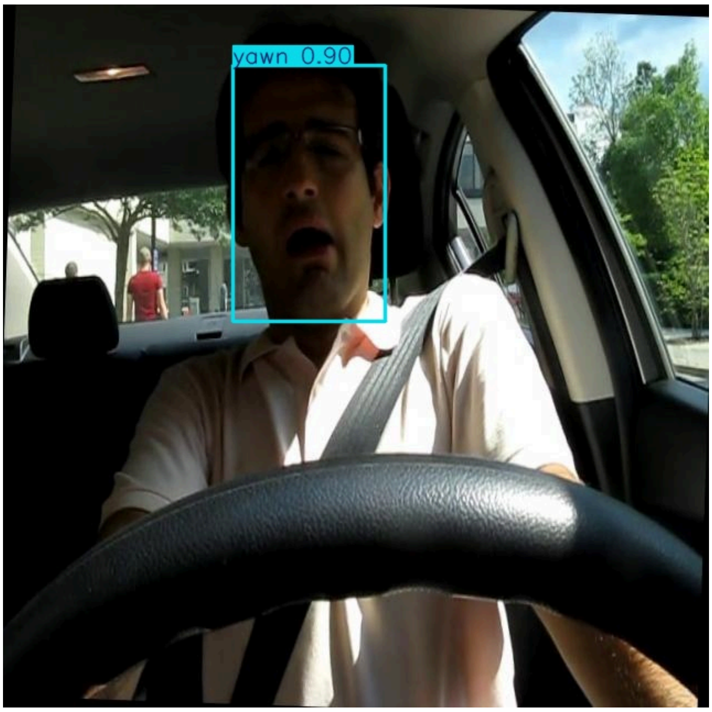

Figure 5.8 Real Time Single Frame Yawn Detection

5.7 REAL TIME YAWN DETECTION AND ALERT MECHANISM

In real time deployment on the Raspberry Pi, the system continuously captures the driver's facial video stream and performs YOLOv8n inference frame by frame, as demonstrated in Figure 5.9. When yawning is detected for multiple consecutive frames with high confidence, a dual alert mechanism is immediately activated to prevent fatigue induced accidents. First, a loud audible buzzer is triggered via GPIO to refocus the driver's attention. Second, the detected frame containing a yawn bounding box, along with a timestamp and the message "Driver is Yawning!", is automatically sent to a designated Telegram group for remote supervision, as shown in Figure 5.10 and Figure 5.12.

Additionally, the Raspberry Pi hardware setup and execution logs validate the real time functioning of both the detection and alert modules, as illustrated in Figure 5.11. This end to end alert system operates with minimal latency and ensures timely intervention, making it highly suitable for smart transportation, fleet supervision, and intelligent driver safety applications.

These frames represent real time detection results captured during continuous monitoring, showing how the model tracks facial changes during yawning. The consistent bounding boxes confirm stable performance across sequential frames.

Fig 5.9 Real Time Yawn Detection

This image displays the Raspberry Pi hardware setup along with the terminal logs generated during yawn detection. The output highlights real time inference, alert triggering, and message sending activity on the embedded device.

Fig 5.10 Real Time Yawn Detection on Raspberry Pi with Telegram Alert Log

This screenshot shows the automated alert notification sent to the driver monitoring group through Telegram when a yawn is detected. The alert mechanism provides immediate remote visibility during critical fatigue events.

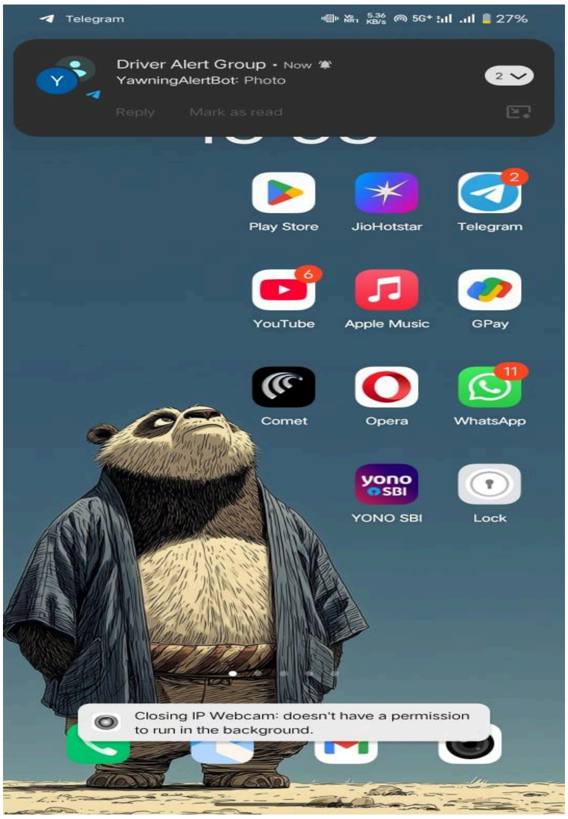

Fig 5.11 Real Time Yawn Telegram Alert

This screenshot shows multiple alert messages received in Telegram, each containing the detected yawning frame. The repeated notifications demonstrate the system's ability to capture and report consecutive yawning events during monitoring.

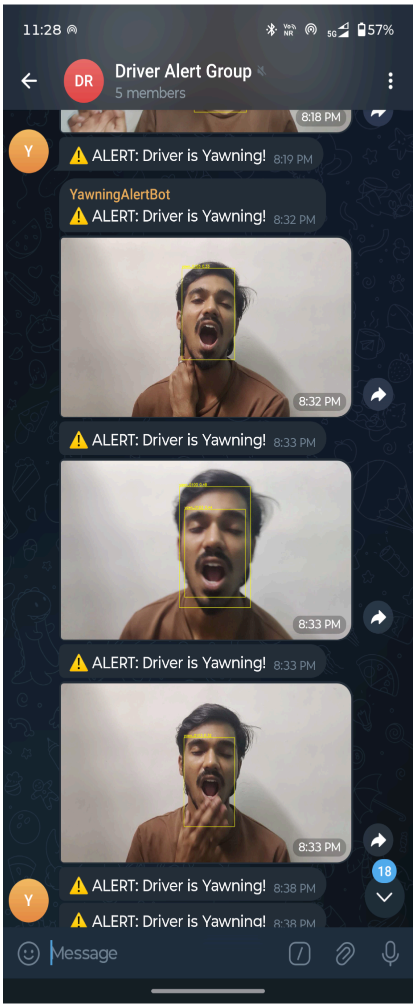

Fig 5.12 Real Time Yawn Detection and Telegram Alert Message

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

The progress in artificial intelligence has greatly influenced intelligent transportation systems, particularly in enhancing real time driver monitoring and road safety. This project leveraged advanced AI methodologies to detect yawning behavior with high precision, enabling early identification of fatigue in vehicular environments. By integrating an improved image preprocessing pipeline with the lightweight YOLOv8n model, the system was capable of efficiently analyzing facial cues that indicate drowsiness. The deployment of the model on a Raspberry Pi platform further demonstrated the practicality of edge based inference, ensuring rapid processing with minimal computational overhead. Additionally, multi channel alert mechanisms, including onboard buzzer warnings and IoT driven Telegram notifications, were implemented to increase accessibility and responsiveness across different driving scenarios. The overall findings highlight the significance of AI based visual monitoring in reducing fatigue related risks and demonstrate the potential of intelligent, real time solutions in promoting safer and smarter transportation.

6.1 CONCLUSION

This work focuses on a comprehensive real time driver fatigue detection system by integrating an image enhancement pipeline with the YOLOv8n detection framework. Preprocessing operations such as normalization, illumination adjustment, and noise suppression significantly improved the clarity of the mouth region and strengthened the overall yawning detection performance. The model achieved strong evaluation results, including Precision of 0.881, Recall of 0.878, mAP@50 of 0.918, mAP@50-95 of 0.825, and an overall YOLO validation accuracy of 0.8755. Further testing on a separate dataset demonstrated exceptional reliability, achieving a test accuracy of 0.9984 with 627 correct predictions out of 628 images, indicating excellent generalization under varied lighting conditions, camera positions, and head orientations.

The YOLOv8n model achieved fast inference, averaging 5.6 ms per frame on embedded hardware, confirming the feasibility of real time deployment without reliance on cloud services. The detection pipeline, combined with dual alert mechanisms through buzzer activation and Telegram notifications, ensured timely and reliable warnings for both drivers and monitoring authorities. Overall, the results validate the effectiveness of lightweight, edge based deep learning models for continuous in vehicle safety monitoring, providing a robust and efficient solution for fatigue related risk prevention.

6.2 FUTURE ENHANCEMENTS

Future enhancements for the fatigue detection system can focus on integrating Infrared (IR) cameras to ensure reliable monitoring during low light and nighttime conditions, adding real time mobile usage detection to identify distractions caused by phone handling while driving, and expanding the alert mechanisms with options such as voice assisted warnings, haptic feedback, or seat vibration for more intuitive driver notifications. Incorporating limited multi modal sensing such as blink rate, eye closure duration, and head pose tracking can further strengthen the accuracy of drowsiness assessment without making the system overly complex. These targeted improvements would significantly enhance reliability, safety, and real world usability while keeping the system lightweight and efficient for deployment in smart transportation environments.

REFERENCES

[1] M. Ngxande, J.-R. Tapamo, and M. Burke, “Bias remediation in driver drowsiness detection systems using generative adversarial networks,” in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, IEEE, pp. 309–316, 2020.

[2] N. S. Karuppusamy and B.-Y. Kang, “Multimodal system to detect driver fatigue using EEG, gyroscope, and image processing,” in *Proc. IEEE Int. Conf. Artificial Intelligence and Signal Processing (AISP)*, 2020.

[3] A. Altameem, A. Kumar, R. C. Poonia, S. Kumar, and A. K. J. Saudagar, “Early identification and detection of driver drowsiness by hybrid machine learning,” in *Proc. Int. Conf. Computing and Communication Systems (I3CS)*, pp. 1–6, 2021.

[4] L. Zhang, H. Saito, L. Yang, and J. Wu, “Privacy-preserving federated transfer learning for driver drowsiness detection,” *IEEE Access*, vol. 10, pp. 12857–12869, 2022.

[5] M. A. Khan, T. Nawaz, U. S. Khan, A. Hamza, and N. Rashid, “IoT-based non-intrusive automated driver drowsiness monitoring framework for logistics and public transportation,” *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10524–10535, 2023.

[6] M. Venkateswarlu and V. R. R. Ch, “DrowsyDetectNet: Driver drowsiness detection using lightweight CNN with limited training data,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 2, pp. 87–94, 2024.

[7] J. Alguindigue, A. Singh, A. Narayan, and S. Samuel, “Biosignals monitoring for driver drowsiness detection using deep neural networks,” *Sensors*, vol. 24, no. 2, pp. 305–319, 2024.

[8] A. Jarndal, H. Tawfik, A. I. Siam, I. Alsyouf, and A. Cheaitou, “A real-time vision transformers-based system for enhanced driver drowsiness detection and vehicle safety,” *Computers in Biology and Medicine*, vol. 165, p. 107901, 2024.

[9] S. V. Dixith, S. Jadhav, Y. Kim, and N. Jayakumar, “Deep learning-based drowsiness detection system for driver’s safety,” *IEEE Access*, vol. 13, pp. 12845–12857, 2025.

[10] D. Dixith, A. Kumar, and Y. Kim, “Hybrid CNN-SVM-based deep learning system for driver drowsiness detection using eye and mouth features,” *Journal of Intelligent Transportation Systems*, vol. 29, no. 4, pp. 452–463, 2025.

[11] S. Abtahi, B. Hariri, and S. Shirmohammadi,"Driver drowsiness monitoring based on yawning detection,"Proc. University of Ottawa Research Repository, 2015.

[12] F. Makhmudov, D. Turimov, M. Xamidov, F. Nazarov, and Y.-I. Cho,"Real-time fatigue detection algorithms using machine learning for yawning and eye state,"Sensors, 2024.

[13] P. Kielty, M. Sefidgar Dilmaghani, C. Ryan, J. Lemley, and P. Corcoran,"Neuromorphic sensing for yawn detection in driver drowsiness,"arXiv preprint arXiv:2305.02888, 2023.

[14] S. B. Birajdar and N. Ahmed, "Drowsiness and Yawn Detection System," International Journal of Engineering Development and Research, vol. 10, no. 2, pp. 169–174, 2022.

[15] S. Essahraui, M. A. El Hassani, Y. Oulhadj, and M. Azizi,"Real-time driver drowsiness detection using facial analysis and machine learning techniques," Sensors, vol. 25, no. 3, p. 812, 2025.

[16] M. Lyu, Z. Yuan, and D. Chen,"Long-term multi-granularity deep framework for driver drowsiness detection,"arXiv preprint arXiv:1801.02325, 2018.

[17] R. M. Salman, M. Rashid, R. Roy, M. M. Ahsan, and Z. Siddique,"Driver drowsiness detection using ensemble convolutional neural networks on YawDD,"arXiv preprint arXiv:2112.10298, 2021.

[18] W. Lindskog, V. Spannagl, and C. Prehofer, "Federated learning for drowsiness detection in connected vehicles,"arXiv preprint arXiv:2405.03311, 2024.

[19] R. Gupta and V. Bari, "Review paper on yawning detection prediction system for driver drowsiness," in *Proceedings of the International Conference on Computational Intelligence and Data Science (ICCIDS)*, 2021.

[20] S. Pandey, A. Kumar, R. Sharma, and P. Jain, "Reliable drowsy driver detection using machine vision and deep learning," *IEEE Access*, 2025.