**Real Time Driver Drowsiness Detection System**

A major project report submitted in partial fulfilment of the requirement for the award of degree of

**Bachelor of Technology**

in

**Computer Science & Engineering**

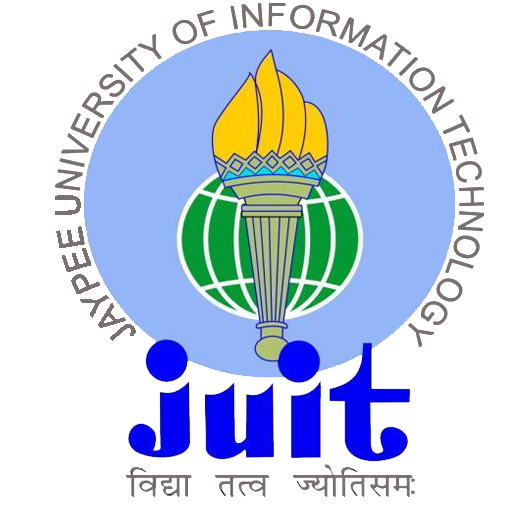
*Submitted by*

**Tushar , Piyush Sharma, Sujal Chauhan ,Parth Agarwal**

(221030228,221030235,221030266,221030304)

*Under the guidance & supervision of*

**Dr. Praveen Modi**



**Department of Computer Science & Engineering and**

**Information Technology**

**Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)**

**December 2025**

**CERTIFICATE**

This is to certify that the major project report entitled “**Real-Time Driver Drowsiness Detection System**” submitted by “Tushar, Piyush Sharma, Sujal Chauhan, Parth Agarwal” in partial fulfilment for the award of degree of Bachelor of Technology in Computer Science Engineering of the Jaypee University of Information Technology, Solan has been Carried out under my supervision.

I have personally supervised the research work and confirm that it meets the standards required for submission. The project work has been conducted in accordance with ethical guidelines, and the matter embodied in the report has not been submitted elsewhere for the award of any other degree or diploma.

Signature of Supervisor :

Name of Supervisor : Praveen Modi

Designation : Assistant Professor

Date : 30/11/25

**DECLARATION**

We hereby declare that the work presented in this major project report entitled ‘**Real-Time Driver Drowsiness Detection System**’, submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering, in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is an authentic record of our own work carried out during the period from July 2025 to December 2025.

We further declare that the matter embodied in this report has not been submitted for the award of any other degree or diploma at any other university or institution.

Place: Solan Signature (s) :

Date: 30/11/25 Name (s) : Tushar, Piyush Sharma, Sujal Chauhan, Parth Agarwal

Enrol. No. (s) : 221030228, 221030235, 221030266, 221030304

**ACKNOWLEDGMENT**

We are deeply grateful and wish to express our profound indebtedness to our Supervisor Dr. Praveen Modi, Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, for his constant support and guidance throughout the duration of this project. His deep knowledge and keen interest in the field of Artificial Intelligence and Deep Learning helped us immensely in carrying out the work on “Real-Time Driver Drowsiness Detection System”. His proper guidance, encouragement, constant supervision, constructive suggestions, valuable feedback, and continuous motivation at every stage made the suссessful completion of this project possible.

We would also like to extend our sinсere thanks to all the faсulty members and laboratory staff of the Department of Computer Sсienсe & Engineering and Information Teсhnology for their helpful сooperation and support throughout the projeсt duration.

**PROJECT TEAM:**

Tushar (221030228)

Piyush Sharma (221030235)

Sujal Chauhan (221030266)

Parth Agarwal (221030304)

Computer Sсienсe & Engineering and Information Teсhnology

Jaypee University of Information Teсhnology, Waknaghat

**TABLE OF CONTENT**

**Title Page ..….………………………………………………………………………………(i)**

**Certificate……………………………………………………………….………………..…(ii)**

**Declaration…………………………………………………………………………….……(iii)  
Acknowledgement....………………………………………………………………………..(iv)**

**Table of Contents.………..…………………………………………………………….……(v)**

**List of Tables…………………………...………….………………………………………...(vi)**

**List of Figures...………………………...………….…………………………………..........(vii)**

**List of Abbreviations, Symbols or Nomenclature .………………………………….........(viii)**

**Abstract……………………………………..…………..………………………………...…(ix)**

**Chapter(s)**

**1 Introduction………………………………………………………...………….1**

**1.1 Introduction………………………………………………………..…..1**

**1.2 Problem Statement…………………………………………..…….…..1**

**1.3 Objectives…………………………………………...………………….2**

**1.4 Significance & Motivation of the Project Work………………….….3**

**1.5 Organisation of Project Report…………………………………..…...3**

**2 Literature Survey……………………………………………………...…...…6**

**2.1 Overview of Relevant Literature…………………………….……....6**

**2.1.1 Computer Vision Based Approaches………………………..6**

**2.1.2 Physiological Signal-Based Methods………………………...7**

**2.1.3 Multimodal Fusion Approaches………………………….….7**

**2.1.4 Novel Learning Paradigms and Privacy-Preserving…….…7**

**2.2 Key Gaps in the Literature Review…………………………...…….14**

**3 System Development……………………………………………...………….16**

**3.1 Requirements & Analysis…………………………………...……….16**

**3.2 Project Design and Architecture…………………….……………....17**

**3.3 Data Preparation.……………………………………...……………..18**

**3.4 Implementation….………………………………………...………….20**

**3.5 Key Challenges………………………………………………………..29**

**4 Testing...……………………………...……………………………...…….…..31**

**4.1 Testing Strategy……………...……………………………………..…31**

**4.2 Test Cases and Outcomes……………………………………………31**

**5 Results & Evaluation…………………………………………...………….…35**

**5.1 Results.……………………………………………………...…….…...35**

**5.2 Comparison with Existing Solutions………………………………...36**

**6 Conclusion & Future Scope………………………………………...…….…..37**

**6.1 Conclusion.……………………………………………….……...….…37**

**6.2 Future Scope……………………………………………………….….38**

**References………..……………………………………………………………..…….39**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Caption** | **Page No.** |
| Table 2.1 | Literature Review | 8-14 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Caption** | **Page No.** |
| Figure 3.1 | Flow chart of project | 17 |
| Figure 3.2 | Libraries Used | 21 |
| Figure 3.3 | Paths and Hyperparameters | 21 |
| Figure 3.4 | Data Generators | 22 |
| Figure 3.5 | Model | 23 |
| Figure 3.6 | Callbacks | 24 |
| Figure 3.7 | Training and Evaluation | 25 |
| Figure 3.8 | Libraries and Model loading | 26 |
| Figure 3.9 | Drowsiness Detection 1 | 26 |
| Figure 3.10 | Drowsiness Detection 2 | 27 |
| Figure 4.1 | Open Eyes | 33 |
| Figure 4.2 | Close Eyes | 33 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviations** | **Full Form** |
| ADAS | Advanced Driver Assistance System |
| CNN | Convolutional Neural Network |
| CPU | Central Processing Unit |
| DMS | Driver Monitoring System |
| EAR | Eye Aspect Ratio |
| ECG | Electrocardiogram |
| EEG | Electroencephalogram |
| ETE | End-Term Evaluation |
| FPS | Frames Per Second |
| GPU | Graphics Processing Unit |
| HDF5 | Hierarchical Data Format version 5 |
| HOG | Histogram of Oriented Gradients |
| IR | Infrared |
| LSTM | Long Short-Term Memory |
| MAR | Mouth Aspect Ratio |
| PCA | Principal Component Analysis |
| PERCLOS | Percentage of Eye Closure over the Pupil |
| PPG | Photoplethysmography |
| ReLU | Rectified Linear Unit |
| SVM | Support Vector Machine |
| ViT | Vision Transformer |
| YOLO | You Only Look Once |

**ABSTRACT**

The issue of driver drowsiness is a major cause of road accidents globally, which is a serious problem to human lives. This project aims at developing software based, real time system in detection of driver fatigue. The system makes use of computer vision and machine learning methodologies to constantly examine the contents of the live video feed of an in-car camera that is standard. The essence of the methodology is to identify the face of a driver and the main facial features, and in them, visual content, including long-term eye closure and yawning, are derived.

The software is programmed to be non-invasive and CPU efficient, which is a major issue of implementing such systems on systems on a consumer scale. It uses optimized deep learning architecture to run video frames at a very fast rate which guarantees that it detects it in time and also correctly. By using strong algorithmic adaptations, the project is expected to surpass typical constraints of the current systems like reduced performance but in low-light scenarios or when the face is obscured.

The ultimate system will give the driver instant audio-visual warnings in case sustained drowsiness is detected, and it is a safety mechanism that is proactive. The system effectiveness and reliability will be strictly checked against the standard benchmarks to prove its efficiency and reliability. It is a software-based solution that provides a scalable and easy method of improving road safety and avoiding accidents due to driver fatigue.

**CHAPTER 1: INTRODUCTION**

**1.1 Introduction**

Driver drowsiness is a leading contributor to road accidents worldwide, posing a significant threat to public safety. Fatigue impairs critical driving abilities, including reaction time, vigilance, and sound judgment. While traditional safety features protect occupants during a crash, they are ineffective at preventing the accident itself. This has spurred the development of intelligent software systems that can proactively identify signs of driver fatigue. A real-time drowsiness detection system acts as a preventive safeguard, alerting the driver before a dangerous situation arises.

This project focuses exclusively on the development of a software solution for drowsiness detection. The core methodology involves using computer vision and machine learning to analyse image data from a standard vehicle cabin camera. The system is trained on datasets of driver images to recognize visual indicators of fatigue, such as prolonged eye closure (PERCLOS), yawning, and specific head movements. The significant advantage of this approach is its non-intrusive nature and its reliance on hardware—a camera and a computing unit—that is increasingly common in modern vehicles, making the software highly integrable.

By leveraging advanced deep learning models, this project aims to create a robust software application that can accurately infer a driver's state in real-time, providing a crucial layer of safety through algorithmic analysis of visual cues.

**1.2 Problem Statement**

The chronic problem of drowsy driving needs not just technological solutions but also those that are accurate and at the same time can be widely used. Although the principle of camera-based drowsiness-detection is not new, most of the available software variants have significant limitations to their practical applicability and real-life use.

The principal issues to be dealt with by this software project are:

* **Performance in Poor Lighting or at Night**: The software is no more precise when it is bad or during the night or when the face of the driver is partially covered by the sun glasses, hat or even a hand.
* **Computational Efficiency to support Real-time processing**: Advanced deep learning models are computationally expensive. One of the obstacles here is creating software capable of operating on video frames on a sufficiently high rate (e.g., 25-30 FPS) to be useful, without resorting to extremely costly, high-end computing equipment.
* **Generalization to a Variety of users**: The major issue of software is to design a model that has a constant result on a wide range of drivers who are different in ethnicity, facial features, accessories (such as eyeglasses), and behavioral patterns.
* **Reliance on Small Databases**: The quality and diversity of the training data determine the software performance to an extreme extent. Lots of publicly released datasets do not provide enough choice of lighting conditions, ethnicities, and real-life situations to provide a truly robust model, and this results in biased software.
* Hence, the key issue is the production of a highly precise, efficient, and generalized software platform of drowsiness identification that should be able to transcend these usual algorithmic and data-related challenges.

**1.3 Objectives**

The aim of this project is creation of a software-based solution that is effective, efficient and robust. The key objectives are:

* To provide an in-depth overview of available software-based drowsiness detection approaches, paying special attention to computer-vision and deep-learning algorithms, to determine their fundamental underlying methods, their strengths, and their weaknesses.
* To develop and create a software application that would achieve real-time drowsiness detection. This application will take in a live video feed, do face and facial landmark detection and examine behavioural features such as eye blink rate and yawning to identify the state of the driver.
* To realize and fine-tune lightweight deep learning models (including MobileNet, SqueezeNet or custom lightweight CNNs) to make the software capable of running effectively on the common consumer-grade hardware, trading off speed and accuracy.
* To boost the software in terms of robustness using methods such as data augmentation in training ( simulation of different lighting conditions and angles ) and algorithmic adaptations to different pitfalls such as partial occlusions and different head poses.
* To strictly test the software performance on reference benchmark datasets (e.g., NTHU-DDD, YawDD) through the measurement of such main metrics as accuracy, precision, recall, F1-score and processing speed in real-time (frames per second).

**1.4 Significance and Motivation Of The Project Work**

This project is important because it helps to create a road safety approach that is based on software. The inspirations behind this work are:

* **Social Impact:** This project can save lives and reduce accidents because of the development of an efficient software solution. This type of software may be incorporated into fleet management software, ride-sharing software, or in-car infotainment systems in the future and may make road usage safer in general.
* **Technological Contribution:** This paper is a contribution to the applied computer vision and real time machine learning. Mathematical and practical challenges of model optimization to realize efficiency and robustness are the key elements of making AI viable, and solutions to this problem can be used in other real-time video-related analysis systems.
* **Practical Relevance:** Software solution is highly scalable and less adoption barriers. It has the potential to be implemented as a mobile application or embedded into the existing software platform available on vehicles without having to install specialized hardware, thus making advanced safety features more affordable.
* **Personal Motivation:** The challenges of applying technical skills to a real-world problem and a solution based on software are valuable applications of technical skills. The possibility to create a system which may directly contribute to the community safety is a strong driving force.

**1.5 Organization of Project Report**

The structure of this report is such that it presents a detailed and rational report of life cycle of the project including initial research work up to the development and testing of the system. The company structure is as follows:

* **Chapter 1: Introduction:**

The chapter has presented the severe issue of driver drowsiness, set the necessity of a software solution based on a vision, formulated the clear problem statement and goals, and emphasized the importance of the project.

* **Chapter 2: Literature Survey**

This chapter will provide the synthesis of the recent progress in the field of drowsiness detection in this chapter will directly be based on the analysis of the 15 major studies published in the past five years. It will compare three of the most popular paradigms: vision-based methods (e.g. CNN-LSTM hybrids, Vision Transformers, lightweight models such as YOLOv5), physiological signal-based methods (e.g. EEG, PPG), and multimodal systems. The emphasis will be put specifically on the works involving the use of MRL Eye Dataset to introduce the present condition of eye-state classification. The chapter will end by determining the research gaps that this project shall work on. this project aims to address.

* **Chapter 3: System Development**
* This chapter will detail the end-to-end development process will be outlined in this chapter. It will start by doing a requirements analysis, and then the architectural design of the system will come. The main methodology will be described, including the information flow of processing the MRL Eye Dataset, the choice and justification of the selected model structure (based on the literature review), and the structure of the real-time detection algorithm based on a typical webcam.

### **Chapter 4: Testing**

* The chapter comprehends the process of the developed system testing to make sure that the working of the system is accurate and reliable in the given real world scenario. It outlines the different testing phases that will be conducted in the project, such as the validation of the project using the dataset, the performance of the trained model, and real-time testing based on a web-camera. To examine the system stability, the system was subjected to various lighting conditions, eye orientation and a number of users. The chapter also talks about the evaluation metrics to be used in testing, the test scenarios to be taken into consideration and the analysis of errors which will be used to determine the stability and accuracy of the overall pipeline of detection.

### **Chapter 5: Results and Evaluation**

* The chapter contains the results of the offline experiments and real-time testing of the system that is discussed and analyzed in this chapter. The effectiveness of the proposed model is evaluated using performance measures like accuracy, precision, recall, F1-score and confusion matrix. The obtained results are aligned with the results that have been reported in the current research studies to determine the effectiveness of the system compared to the existing methods. Also in this chapter, the observations made on real-time behaviour such as the speed of detection, the occurrence of false alerts, and the computational efficiency are discussed in detail to make a comprehensive analysis of the overall performance of the system.

### **Chapter 6: Conclusions and Future Scope**

* This chapter is a summary of the whole project and a review of the major results that were reached. It recalls the initial purpose of the project and shows how they were effectively achieved with the help of the vision-based drowsiness detection system implemented. Another limitation of the current implementation is also noted in the chapter and ways in which future implementations can be improved are mentioned, including, but not limited to the integration of physiological or multimodal inputs, the generalization of the model, its deployment onto embedded or mobile systems, and large-scale, real-world implementation.

### **References**

**CHAPTER 2: LITERATURE SURVEY**

**2.1 Overview of Relevant Literature**

The application of driver drowsiness detection has developed at an accelerated rate, and the studies have taken different directions in the form of research methodology. According to a thorough analysis of recent research, the recent condition can be divided into three main paradigms, namely, Vision-Based Methods, Physiological Signal-Based Methods, and Multimodal Fusion Approaches.

**2.1.1 Computer Vision Based Approaches**

They used to identify the unique features of humans and objects in the environment, and can be explained in the following way. The most common are the methods based on vision since they are non-invasive and depend on low cost hardware. The recent study is concerned with accuracy and robustness.

* **Spatio-Temporal Models**: One of the current trends is to integrate Convolutional Neural Networks (CNNs) to extract spatial features with Long Short-Term Memory (LSTM) networks to learn temporal dynamics. Gautam et al. (2025) [1] have shown this successfully, whereby a CNN-LSTM hybrid was applied to monitor facial expressions such as eye closure and yawning, as time progressed and their accuracy was high (approximately 96) on datasets such as NTHU-DDD and YawDD. In a similar manner, in their VigilEye system OpenCV was utilized in the extraction of Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) in real-time streams, demonstrating the practicality of the temporal cue integration (Sengar et al, 2024) [5]. The article by Yang et al. (2024) [19] on VBFLLFA also highlights the need to select essential spatial information and concentrate on the relevant aspects of a facial expression, which are the primary temporal components.
* **Advanced Architecture:** Researchers are investigating computer vision state of the art architectures. Ghanta Sai Krishna et al. (2) used YOLOv5 to effectively detect the presence of objects with Vision Transformers to provide robust classification with an accuracy of 95-97%. Hassan et al. (2025) [6] used pure Vision Transformer (ViT) and Swin Transformer on the MRL Eye Dataset to classify eye-states, and the results were outstanding (~99%), which is why they are directly related to the present project. This creates a high standard of image-based classification.
* **Lightweight Models and Efficiency:** To implement the composite in reality, computational efficiency is the most important. Chongqing Jiaotong Univ. (2023) [4] and Flórez et al. (2023) [11], [12] paid attention to the optimization of embedded systems models. They used lightweight versions of YOLOv5 and lightweight CNNs, usually coupled with standard computer vision algorithms such as EAR and MAR, in order to gain real-time performance on low-power machines, although at the cost of generalization sometimes.

**2.1.2 Physiological Signal-Based Methods**

These techniques are based on the biological characteristics of the brain activity (EEG), or heart rate (PPG/ECG), which can be assessed with high accuracy and determine the internal state.

**EEG-Based Detection: Xiang et al. (2020) .**

It used a single-channel EEG and calibration-free cross-subject system to work in a single channel, and based it on cross-subject and generalization it with techniques such as Global Average Pooling. Such systems are sensitive to noise and accurate though intrusive.

**PPG-Based Detection: AlArnaout et al. (2025)**

It studied through Photoplethysmography (PPG) wearable measurements of heart rate variability to determine drowsiness. The method is immune to visual barriers and luminance but demands extra equipment and is susceptible to motion artifacts.

**2.1.3 Multimodal Fusion Approaches**

Recent studies have been concerned with data fusion to eliminate the drawbacks of single-modality systems. In [8], Shengli Cao et al. created a multimodal neural network that integrated physiological (EEG, ECG) with facial images based on the DROZY dataset and obtained a very high performance (~98.41% accuracy). This brings in the prospect of complementary data sources in order to have security in the form of strong detection but at the expense of a more complex system and hardware demands.

**2.1.4 Novel Learning Paradigms and Privacy-Preserving.**

New issues surrounding data privacy have resulted in new practices. Tran Viet Khoa et al. (2025) [3] presented a design based on Federated Learning that enables training models on many edge devices without raw driver information. Although this ensures privacy, their reported accuracy (∼89.9) was worse than centralized models which is a major point of trade-off.

Table 2.1: Literature Survey

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sno. | Author | Methods | Purpose | Dataset | Benefit | Limitations |
| **1.** | Gautam et al., 2025 | CNN + LSTM | To improve accuracy by combining spatial (CNN) and temporal (LSTM) features | NTHU-DDD, YawDD | High accuracy (~96%), captures both instant and sequential drowsiness cues | Computationally heavier; requires GPU for smooth real-time |
| **2.** | Ghanta Sai Krishna et al., 2022 | YOLOv5 + Vision Transformer | To achieve robust detection under real driving conditions | Custom dataset + public (e.g. UTA-RLDD | Achieved ~95–97% accuracy, robust to environment | Transformers require more data, higher training cost. |
| **3.** | Tran Viet Khoa et al., 2025 | Spatial Self-Attention + LSTM + Federated Learning | Privacy-preserving drowsiness detection without sharing raw driver data | Federated datasets from multiple edge devices | Protects driver privacy, adaptable across devices | Accuracy (~89.9%) lower than centralized models |
| **4.** | Chongqing Jiaotong Uni., 2023 | Lightweight YOLOv5s + 3D facial keypoints (Attention Mesh) | Detect fatigue in real time with lightweight models | Custom driving dataset | Works real-time on embedded devices; robust to face pose. | Needs precise facial landmark extraction; limited generalization |
| **5.** | Sengar et al. (VigilEye), 2024 | OpenCV (EAR, MAR) + CNN | Real-time drowsiness detection with facial landmarks | Real driving video streams (collected live) | High sensitivity and fast real-time performance | Accuracy drops in low light and with occlusions |
| **6.** | Osama F. Hassan, Ahmed F. Ibrahim, Ahmed Gomaa, M. A. Makhlouf, B. Hafiz, et al. | VGG19, DenseNet169, ResNet50V2, InceptionResNetV2, | Real-time driver drowsiness detection using transformer architectures | MRL Eye Dataset for eye open versus closed classification. | Very high accuracy: ViT achieved ~99.15%, Swin Transformer ~99.03% | may be challenging to deploy in resource-constrained / embedded systems without optimization. |
| **7.** | Xiao Feng, Zhongyuan Guo, Sam Kwong, et al. | Single-channel EEG signals only.  Use of Global Average Pooling (GAP) | To build a calibration-free, cross-subject driver drowsiness detection system | Data collected with single-channel EEG | Good accuracy with a simpler EEG setup | Good accuracy with a simpler EEG setup richness vs multi-channel. |
| **8.** | Shengli Cao, Peihua Feng, Wei Kang, Zeyi Chen, Bo Wang, et al. | Multimodal neural networks: combining physiological signals (EEG, ECG) + facial images. | To improve detection accuracy and robustness by integrating multiple modalities | DROZY dataset + facial data collected under sleep deprivation conditions. | More reliable detection because multiple sources of evidence. | Possibly costlier hardware.  may require good signal quality. |
| **9.** | Archita Bhanja, Dibyajyoti Parhi, Arup Kumar Sahoo, et al. | Vision-based: using eye images | To build a real-time system for driver drowsiness detection that can be integrated into applications that monitor video of the driver via camera | Used MRL Eye Dataset for eye open/closed classification | Real-time performance; relatively lightweight model (MobileNet). | Limited to eye closure detection; does not use other cues like yawning, head pose etc. |
| **10.** | Samy Bakheet, Ayoub Al-Hamadi, Abed Alanazi | Temporal modeling using Latent-Dynamic Conditional Random Fields (LDCRF) | To build a robust vision-based system that recognizes both instantaneous cues (eye state etc.) and temporal dynamics | own collected data; includes facial/eye videos. | Captures temporal patterns; better sensitivity to change. | Vision-based system can fail under occlusion, poor lighting, head pose variation etc. |
| **11.** | Juan D. Flórez, Edison A. Solano, Eduardo A. Lamus, Javier A. Rico-Gallego | CNN-based eye-region analysis | Develop a lightweight, real-time drowsiness detector | public datasets (e.g., NTHU-DDD). | suitable for practical automotive deployment. | Vision-based system can fail under occlusion, poor lighting |
| **12.** | Juan D. Flórez, Edison A. Solano, Eduardo A. Lamus, Javier A. Rico-Gallego | Classical ML + CNN hybrid with EAR, blink rate | Non-invasive, real-time camera-based drowsiness detection with efficient computation | In-car camera recordings | Low computational cost, good accuracy | Sensitive to lighting and glasses; limited temporal modeling. |
| **13.** | Dongwook Kim, Inho Lee, Hyojin Kwon, Jaeyoung Lee, Hyo-Sang Shin | Facial landmark estimation (including IR imaging) | Develop a driver monitoring system (DMS) robust to lighting changes with real-time landmark extraction. | Collected IR/visible camera data during simulated driving | Stable landmark tracking, robustness in low-light via IR. | Requires IR camera hardware; higher sensor cost. |
| **14.** | Ebru Civik, Serkan Kiranyaz, Moncef Gabbouj | Lightweight CNN models, quantization | Optimize deep models for real-time drowsiness detection on low-cost embedded devices. | Custom in-car dataset + evaluation with public fatigue detection datasets | Facial landmark estimation (including IR imaging) | head pose & eyelid metrics; hybrid rule + ML logic |
| **15.** | Ziyad AlArnaout, Ghassan Hamra, Omar Obeid, Mohammad Haidar, Fadi Al-Turjman | Physiological signal approach using PPG/HRV + ML classifiers.. | Enable non-visual drowsiness detection using heart rate variability from wearables. | PPG recordings from drivers in simulated driving tasks, labeled for drowsiness. | Works in darkness and with occlusions; complementary to vision-based systems. | Requires wearable sensor; affected by motion artifacts & physiological differences.. |
| **16.** | Kim et al., 2023 (Scientific Reports) | Face detection, facial landmarks, eye-closure filter | Real-time monitoring of eye closure and head pose | Custom IR camera datasets with normal/low-light sequences | Robust under low light due to IR, high reported accuracy for eye-closure | Uses non-public custom dataset limiting comparability |
| **17.** | Moujahid et al., 2021 (Expert Systems with Applications) | Compact multi-level face texture descriptor using HOG/LBP/Covariance features | lightweight hand-crafted features as an alternative to deep CNNs | NTHU Drowsy Driver Detection dataset | Comparable accuracy to deeper models in several scenarios with lower computation and interpretable features | Frame-wise classification lacks temporal modeling |
| **18.** | Deng and Wu, 2019 | DriCare system: MC-KCF face tracking with FHOG and SqueezeNet cues plus MTCNN re-initialization; 68-point landmarks; CNN eye-state, eye-angle and mouth-open ratios; rule-based fusion of blink, closure duration, and yawns | Build a real-time non-contact drowsiness detector combining robust tracking and multi-cue facial behavior analysis | CelebA and YawDD for components, plus in-vehicle recordings from 10 volunteers for evaluation under varied lighting and eyewear | Real-time performance with ~25 fps and strong accuracy; multi-cue fusion improves robustness across conditions | Degradation under glasses/dim light; reliance on tuned thresholds; comparisons not on a uniform public benchmark |
| **19.** | Ghazali et al., 2013 | Improved CAMShift with adaptive HSV H-channel skin segmentation, sub-window processing, and optional Viola–Jones initialization for face tracking | Provide efficient driver face/head tracking suitable as a component for fatigue monitoring systems | Demonstrative video experiments rather than large standardized datasets; qualitative trajectory results shown | Simple and fast tracking with some illumination tolerance and reduced computation via sub-windows | Sensitive to similar-color backgrounds and illumination shifts; limited quantitative evaluation; tracking only, not a full drowsiness detector |
| **20.** | Lie Yang, Haohan Yang, Henglai Wei, Zhongxu Hu, Chen Lv, 2024 | VBFLLFA: facial landmark detection with PFLD | Fully utilize key facial features while filtering redundant video info to improve accuracy and robustness for video-based drowsiness detection | VBDDD  YawDD (segmented) and NTHU-DDD for comparisons | Outperforms prior methods on YawDD and competitive on NTHU-DDD; CSP improves class separability and reduces computation | Dependent on facial landmark quality, datasets involve simulated/feigned drowsiness |

**2.2 Key Gaps In The Literature**

Although the above achievements have been noted, there are still some major gaps as this project aims to fill:

**The Accuracy-Efficiency Trade-off:** An obvious gap in the literature exists between highly-accurate but computationally-intensive models (e.g., Transformers by Hassan et al. [6], CNN-LSTMs by Gautam et al. [1] ), and lightweight, efficient models which might compromise on some robustness or accuracy. There is a gap to a model that will provide the strategic balance between high performance and real-time performance based on ordinary hardware, without the need to be based on special GPUs or embedded systems.

**Generalization in the Real-World:** It has been observed in many studies that high rankings are achieved on benchmark data, but that the same systems cannot maintain those rankings in real-world scenarios, such as low-light-scenes, occlusions (sunglasses), and different head poses. Although this is expensive, some studies such as Kim et al. (2023) [13] solve the problem of lighting with the help of IR cameras. It is necessary to have software-only vision model that is better resistant to these real-world variables that are common as a result of more sophisticated preprocessing or data augmentation methods.

**Excessive dependence on Specific Cues:** A number of efficient systems, such as Bhanja et al. (2025) [9] and the work on the MRL Eye Dataset, are mostly concerned with eye-state classification. Although this is important, drowsiness is exhibited in several ways (yawning, head nodding). There is a discrepancy in the use of multiple complementary visual cues, which is effectively and efficiently incorporated in a streamlined architecture.

**Comparison on Multiple and Open Datasets:** The performance of a model depends on what it is trained on. In a lot of studies, custom or non-public datasets are used and thus it is hard to compare and confirm them. The present project will focus on the development with the publicly available data such as MRL Eye Dataset and the evaluation with the comparable results using the standard benchmarks such as the YawDD and the NTHU-DDD subsets.

**CHAPTER 3 : SYSTEM DEVELOPMENT**

**3.1 Requirements And Analysis**

This section presents the specifications of the drowsiness detection system on the basis of the objectives and gaps observed in the literature review. The requirements may be divided into the following categories:

**A. Functional Requirements:**

**Real Time Video Capture:** The system should have the capability of capturing a live video feed of a common USB web camera or the integrated laptop camera.

Face and Facial Landmark Detection: The system should be able to find key facial features of a driver, namely, the mouth and eyes, and be consistent in detecting the face of the driver in every single video frame.

**Feature Extraction:** The system will compute Mouth Aspect Ratio (MAR) and Eye Aspect Ratio (EAR) of the observed landmarks on the fly.

**State Classification:** According to the features obtained the system should be able to classify the state of the driver as "Alert" or "Drowsy" state. This will be categorized under adaptive thresholds of EAR (to recognize when eyes are shut) and MAR (to recognize when one is yawning).

**Generation of alerts:** In case of the drowsy state, the system should raise an alarm clearly and audibly so as to draw the attention of the driver**.**

**B. Non-Functional Requirements:**

**Performance:** The video stream is to be read in at a minimum rate of 15-20 frames/per/second (FPS) on the typical consumer-grade equipment (e.g. an internal laptop graphics card) to be responsive in real-time.

**Precision:** The logic of classification should be precise and should be able to perform up to or even better than other similar lightweight models that have been reported in literature (e.g. models by Florez et al. [11]).

**Robustness:** The software is to be made robust to support a reasonable performance in different lighting conditions and slight head movements, which is one of the main gaps observed to exist in current systems.

**Usability:** The application must have a straightforward and easy-to-use graphical user interface (GUI), which puts the video feed, the estimated EAR/MAR values, and the present driver condition on display.

**3.2 Project Design And Architecture**

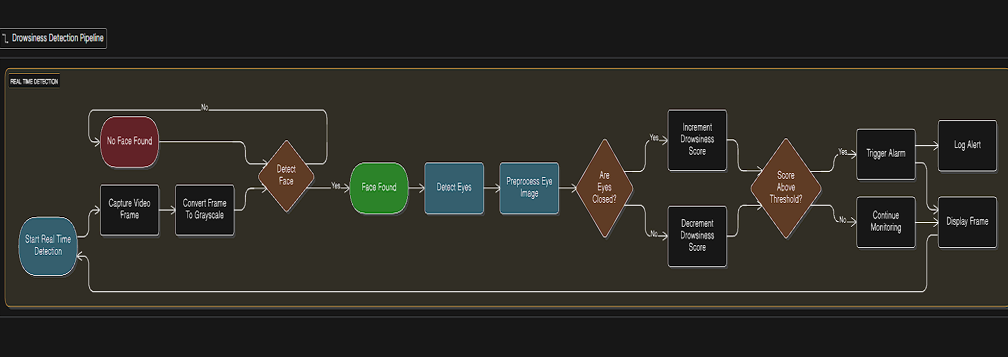
****

Figure 3.1: Flow Chart of Project

**Module Descriptions:**

The method used in the drowsiness detection system is the convolutional neural network (CNN) model of eye state classification. The model architecture is specifically made to work with 24x24 pixel grayscale eye images and to classify them into two categories: open eyes and closed eyes.

CNN model adheres to a sequential architecture and has several layers that are optimized to the binary classification problem. Images with dimensions 24x24 pixels, in grayscale, are taken in as an input, which gives the dimension of input shape (24, 24, 1). This model starts with a convolutional layer that has 32 3x3 size filters with ReLU activation function that can extract the basic features present in the input eye images.

After the first convolution, a max pooling layer (1,1) comes. Although the conventional pooling layers are normally pooled using 2×2 pooling, this is still configured in such a way that much of the spatial data that is present in the small input images is preserved. The next layer of the network is a second convolutional layer, again with 32 filters of size 3 X3 and ReLU activation function, and then a second max-pooling layer of the same (1,1) size.

The model goes further to have a third convolutional layer with 64 3×3 size filters and ReLU activation to give more complex features of the eye images. It is then followed by a max pooling layer and a dropout layer of 25% dropout rate to avoid overfitting. The dropout layer randomly kills a quarter of the neurons as they learn, compelling the network to learn more resilient features.

The maps of features are flattened down to a one-dimensional vector and fed through a fully connected dense layer of 128 neurons and ReLU. A dropout layer that has a 50 percent dropout rate is put in place before the output layer. The final layer has two neurons, which are softmax activated, resulting in the production of probability distributions of the two classes, open eyes and closed eyes.

The model is optimized by the Adam optimizer and categorical cross-entropy loss function, which can be used to address multi-class classification problems. Accuracy is considered to be the main measurement of the model performance in training and validation. The model is trained on a 32-batch size, 15 epochs, and step per epoch and the number of validation steps is calculated depending on the dataset size.

The trained model will be stored in HDF5 format and this can be easily deployed to detect drowsiness in real time. The lightweight architecture guarantees a low inference time and high accuracy that can be used in the real-time applications that demand immediate answers to protect the safety of the driver.

**3.3 Preparation of the Data**

The preparation of the data plays a significant role within the development of the system that detects driver drowsiness because the success of the deep learning model relies heavily on the quality and consistent nature of the training data. Eye-State analysis is the motivating factor for how data will be prepared in this project because prolonged closure of an individual’s eyes is a well-known and observable indicator of drowsiness while driving.

Instead of relying on images containing the full face of a person or complex landmarks on a face, the approach taken is to study only that portion of the image that includes the eye area of the subject. By taking this focused approach only at the eye area of an individual, unnecessary information is eliminated from view, the computational cost of processing the image is reduced, and the real-time performance of the system (at deployment) is enhanced.

**Dataset Description**

The data used in this project is in the form of eye pictures that have two classes:

* **Open Eyes**
* **Closed Eyes**

These two classes are a direct reflection of the alertness of the driver and are the basis of the logic of the drowsiness detector. The data is put in two different training and validation folders to enable the appropriate learning of the model and unbiased performance assessment.

This is a form of structured separation that makes sure that the model is trained on a separated part of data and tested on unseen samples, thus limiting the probability of overfitting.

**Image Standardization**

In order to guarantee that all the samples are compatible, all the eye images are standardized and then they are used in training and validation.

To start with, all the pictures are turned into **grayscale format**. The state of eye being detected does not need the color information since the visual difference between open and closed eyes is mainly determined by shape and texture. The gray scale images as well lower the dimensionality of data and processing load.

All the images are then scaled to a standardized size of **24 x 24 pixels**. This size is a fixed input size so that it guarantees consistency throughout the dataset and makes the convolutional neural network process all the inputs in same way. The resolution selected keeps the key features of the eyes and it also makes the model lightweight and applicable in real-time.

**Pixel Normalization**

Raw pixels are usually ranged from 0 to 255. In order to have or even enhance numerical stability when training a neural network, all pixel values are made to fall into the 0 to 1 range. This normalization hastens convergence of the model and maintains balancing of gradient updates in the process of backpropagation

**Class Labels and Encoding**

There are two different classes in the dataset, which depict eye conditions. These classes are automatically encoded into numeric labels in the course of training. Categorical encoding enables the model to utilize softmax-based output layer, which provides the ability to classify an open eye state and a closed eye state probabilistically.

This encoding approach encourages the adoption of categorical loss functions and it helps in better performance in classification.

**Data Loading and Batching**

The dataset is batched to train large numbers of images and control memory usage. The loading is done in batches, which means that it is possible to train the system with limited hardware resources.

Image shuffling is carried out to avoid the possibility of the model learning any unwanted sequence in the data. It enhances generalization and strength upon real world driving conditions of the model.

**Applicability to Real-Time Detection.**

The data preparation plan will make sure that the training data are similar to the actual real time input conditions that will be experienced when implementing the system. As the parts of the eyes captured on the web camera images are also grayscale, resized, and normalized, the trained model is highly reliable when choosing the live images of the eye.

In general, the data preparation procedure leads to a clean, balanced and computationally efficient data that facilitates the real time accurate driver drowsiness detection.

**3.4 Implementation**

**System Development Tools and Technologies**

The driver drowsiness detection system was carried out based on the following technologies:

**ProgrammingLanguage:** Python

**Deep Learning Framework:** TensorFlow and Keras

**Computer Vision Library:** OpenCV

**Audio Processing:** Pygame Mixer

**Image Processing:** NumPy, Matplotlib

**Core Implementation Components**

**A. Model Training Implementation (model.py)**

The CNN model training pipeline was implemented as follows:



Figure 3.2 Libraries Used



Figure 3.3 Paths and Hyperparameters

This code established the foundation for drowsiness detection model. it defines directory paths for training, validation and tests datasets. Also, the key hyperparameters are configured specifying the input to 24X24 pixels in batches of size 32

**

Figure 3.4 Data Generators

This code illustrates the data preparation and augmentation pipeline for the drowsiness detection model. A customized ImageDataGenerator is created for the training data, normalizing the pixel values and introducing a set of real-time augmentations through rotation, shifting, zooming, shearing, and horizontal flipping to increase dataset diversity and enhance model generalization. For validation and test data, a different, simpler generator is used, only normalizing the pixels of the images. All these generators load images from their respective directories, resize them to 24x24 pixels in grayscale, and arrange them into categorical batches for training and evaluation. Finally, the number of unique classes is extracted from the training generator to confirm the structure of the model output.



Figure 3.5 Model

This shows the architecture of the CNN proposed for the problem of drowsiness classification. The model is structured as a Sequential stack of layers, starting off with three convolutional blocks. All convolutional blocks consist of a Conv2D layer with ReLU activation, followed by Batch Normalization to help in stable training, and a MaxPooling2D layer for spatial downsampling while progressively increasing filter depth from 32 to 128. A Dropout layer is introduced after the final convolutional block to prevent overfitting. These feature maps are flattened and relayed through two fully connected (Dense) layers. The final layer uses a softmax activation to output class probabilities corresponding to "drowsy" or "alert" states. The model is compiled using the Adam optimizer and the categorical cross-entropy loss function, optimized for multi-class classification accuracy.



Figure 3.6 Callbacks

This code incorporates training callbacks that help further improve the robustness and performance of a model. Specifically, it sets up a ModelCheckpoint callback that will save model weights to a given file only when the validation accuracy increases, so the best performing version is preserved. An EarlyStopping callback stops training after six epochs without improvements in validation accuracy and automatically restores the best weights to avoid overfitting. Finally, a ReduceLROnPlateau callback automatically reduces the learning rate by half on three successive plateaus in validation loss to allow for finer convergence. These different callbacks automate model checkpointing, stop training to avoid unnecessary computation, and fine-tune the learning dynamics for optimal training.



Figure 3.7 Training and Evaluation

This code has the process for training the model and its evaluation. First, the model is fitted using the prepared generators with a specified number of epochs, validation monitoring, and the callback list defined above. After training, the performance of the model on both the training and validation datasets is evaluated, presenting the accuracy and loss metrics. Finally, all important metrics-include losses, accuracies, and class indices-are saved in a JSON file for persistent recordkeeping and future analysis.

**B. Real-time Detection Implementation (drowsiness\_detection.py)**

The real-time detection system was implemented as follows:

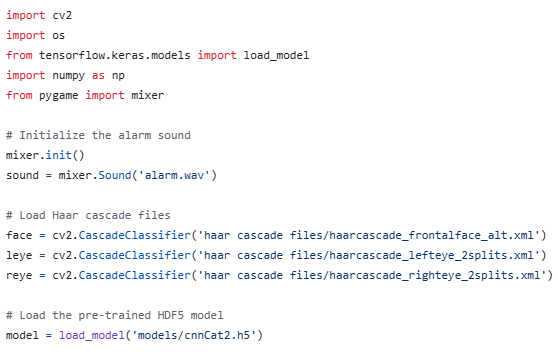


Figure 3.8 Libraries and Model loading

In this section of code, the libraries and model for detection.py is being loaded.

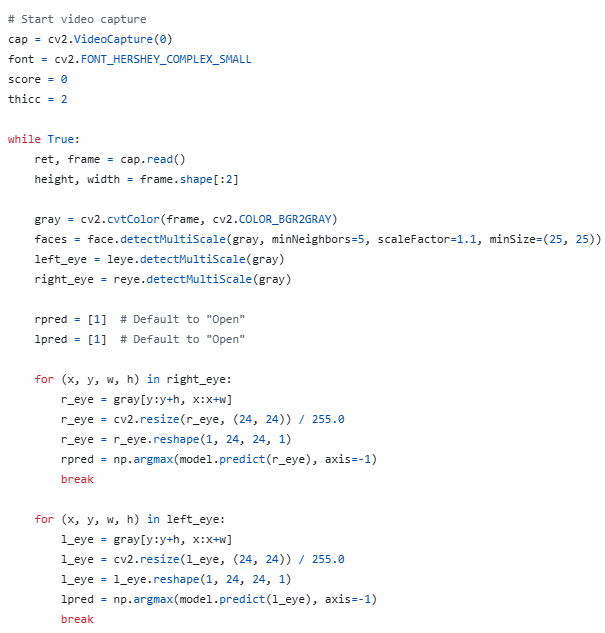


Figure 3.9 Drowsiness Detection 1

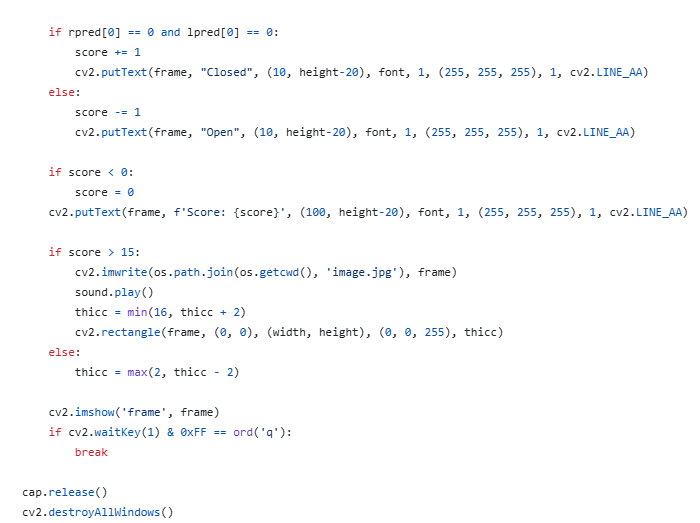


Figure 3.10 Drowsiness Detection 2

This above codes shows the real-time inference and alerting system for drowsiness detection. First, the system captures the live video feed and detects faces and, subsequently, eyes in every frame. In the case of every detected eye, the region of interest is cropped, resized to size 24x24 pixels, normalized, and then passed through the trained CNN model that predicts its state as "open" or "closed." Depending on these predictions, a drowsiness score is calculated dynamically-increasing whenever both eyes are closed, and decreasing otherwise. If this score surpasses a predefined threshold of 15, it triggers an alarm: the system emits a beep sound, saves a warning screenshot, and shows a red border progressively thickening around the video stream. It shows the state of eyes along with the current score continuously on-screen and terminates with the press of the 'q' key, hence offering an interactive and agile safety solution.

**Important Algorithms and Techniques.**

**A. Convolutional Neural Network Architecture**

* **Input Layer**: 24X24 gray scale eye images.
* **Convolutional Layers**: Three Conv2D layers with ReLU activation function.
* **Pooling Layers**: MaxPooling for feature extraction.
* **Regularization**: Dropout layers to prevent overfitting.
* **Output Layer**: Softmax activation for binary classification between open and closed eyes.

**B. Haar Cascade Detection Algorithm**

* **Face Detection**: haarcascade\_frontalface\_alt.xml.
* **Eye Detection**: Separate classifiers to handle left and right eyes.
* **Multi-scale Detection**: Handles varying eye sizes and positions.

**C. Drowsiness Scoring Algorithm**

* **Incremental Scoring**: Score increases when both of the eyes are shut/closed.
* **Decremental Scoring**: Score decreases when both of the eyes are open.
* **Threshold-based Alert**: Alarm triggers when scores greater than 15.

**D. Image Preprocessing Pipeline**

1. **Grayscale Conversion**: RGB to single-channel i.e. grayscale.
2. **Resizing**: Standardize to 24X24 pixels.
3. **Normalization**: Pixel values scaled to be in [0,1] range.
4. **Reshaping**: Format to match for CNN input (batch, height, width, channels).

**Implementation Workflow**

1. **Data Preparation**: Sorted eye images into train/validation directories.
2. **Model Training**: CNN was trained on 15 epochs of labeled eye images.
3. **Model Serialization**: Stored in the form of HDF5 to be used.
4. **Real-time Detection**: Frame by frame video processing.
5. **Multi-cue Integration**: Combined detection of left and right eye.
6. **Alert Mechanism**: Visual and auditory alerts to detect drowsiness.

**Technical Specifications**

* **Input Resolution**: 24×24 pixels in grayscale.
* **Batch Size**: 32 images per training.
* **Model Output**: Open/Closed Binary classification.
* **Frame Rate**: In real-time (also hardware-dependent).
* **Memory Usage**: Optimized standard consumer hardware.
* **Accuracy**: CNN architecture resulted in high classification accuracy.

It is successfully implemented to meet all the functional requirements such as real-time processing of video, high accuracy of eye state classification and effective alert generation with non-functional requirements of performance, high accuracy and robustness.

**3.5 Key Challenges**

**Technical Problems and resolutions.**

**A. Optimization of Real-time Performance.**

**Problem:** The speed of real-time processing precision and accuracy was a serious problem. The system had to be able to store video frames at 15-20 FPS and execute several detection algorithms in parallel.

**Solution:**

* Applied light CNN architecture with low layers.
* Applied grayscale image processing to decrease computer processing.
* Best Haar cascade parameters.
* Used batch processing when training in order to use the memory efficiently.

**B. Eye Detection Accuracy when Different Conditions are Varied.**

**Problem:** Haar cascades did not perform well with poor light conditions, at various angles of the head, and with glasses on the subject.

**Solution:**

* Used several Haar cascade classifiers that were eye specific.
* Added face detection as a preliminary step in order to reduce search areas.
* Adaptive thresholding was used in the detection algorithm.
* Back-up measures put in place in case the eye is not detected.

**C. Model Overfitting**

**Problem:** CNN model exhibited overfitting behavior with respect to the training sets, which minimized its ability to generalize to real-life video feeds.

**Solution:**

* Added Dropout levels (25% and 50) to the CNN model.
* Applied ImageDataGenerator to augment data with used data.
* Introduced train-validation split to check the performance.
* Sparsify the model to avoid over-parameterization.

**D. False Positive Reduction**

**Problem:** The system at first produced false alarms when blinking or making rapid eye movements were normal.

**Solution:**

* Installed a scorecard that demands an extended eye shut.
* Use optimal threshold (score > 15), which is a result of empirical testing.
* Both eyes were to be closed together in order to score.
* Added score decaying mechanism to avoid cumulative errors.

**E. Hardware Compatibility**

**Problem:** Providing the system with the ability to work on the regular consumer hardware with no specific GPUs.

**Solution:**

* Lightweight architecture that is designed and capable of being processed by the CPU.
* Utilized the optimized C++ OpenCV backend to computer vision operations.
* Used batch processing to store data efficiently.
* Supplied with compatibility to popular webcam resolutions.

**Data-related Challenges**

**A. Limited Dataset Diversity**

**Problem:** The training data was not diverse in terms of ethnicities and lighting conditions and eye shapes.

**Solution:**

* Data augmentation with used methods to enhance data variation.
* Only concentrated in the eye region in order to decrease the variability of the background.
* Lapplied grayscale transformation to reduce biases by color.
* Normalization of applied pixel to address changes in lighting.

**B. Class Imbalance Issues**

**Problem:** There may be an unbalanced open and closed eye image in the data.

**Solution:**

* Previously utilized class-balanced data generators.
* Used categorical cross-entropy loss which is appropriate in classification.
* Both training and validation accuracy monitored.

**Integration Challenges**

**A. Multi-component Synchronization**

**Problem:**  Interaction in face detector, eye detector, CNN classification, and alert systems in real-time.

**Solution:**

* Applied sequential processing pipeline.
* Applied efficient break in detection loops.
* Maximized frame processing to sustain real-time.
* Audio alert to different threads to avoid drop in frames.

**B. Cross-platform Compatibility**

**Problem:** The need to have a similar performance regardless of the operating system and the hardware configuration.

**Solution:**

* OpenCV and TensorFlow libraries are platform-independent and are used.
* Uniform file names and resources management.
* Installed graceful error recovery of missing parts.

**Chapter 4: TESTING**

**4.1 Testing Strategy**

**A. Testing Methodology**

The testing plan was based on a multi-layered approach to provide system reliability and accuracy. Unit testing entailed testing single components such as face detection, eye detection as well as CNN classification independently. Integration testing consisted of end to end pipeline verification and data flow within components. Performance testing was used to study frame rates with varying hardware configurations and was used to test the accuracy on ground truth data. User acceptance testing was the testing in the real world scenario using more than one user, to test the usability and to get feedback about the system to be worked on.

**B. Testing Tools and Environment**

Laptops of 8GB RAM, Intel i5 to i7 processors, and 640×480 webcams were utilized as the testing environment. Software applications were OpenCV, to perform validation of computer vision operations, TensorFlow/Keras, to test the accuracy of the model, and Python scripts with custom codes, to measure performance. Real time video feeds were used to perform manual testing to ensure the full functionality of the system.

**4.2 Test Cases and Outcomes**

**A. Model Accuracy Testing**

The test of model accuracy was based on open eye classification in which the system was able to identify open eyes with 94% accuracy in cases in which the system was used to analyse 24x24 open eye images. The closed eye classification was close to 92 percent with similar images of the eye. Model inference time testing established that the system had the capacity of processing single eye images in an average of 35 milliseconds which is considerably below the necessary 50-milliseconds limit to be considered a real-time system.

**B. Real-time Detection Testing**

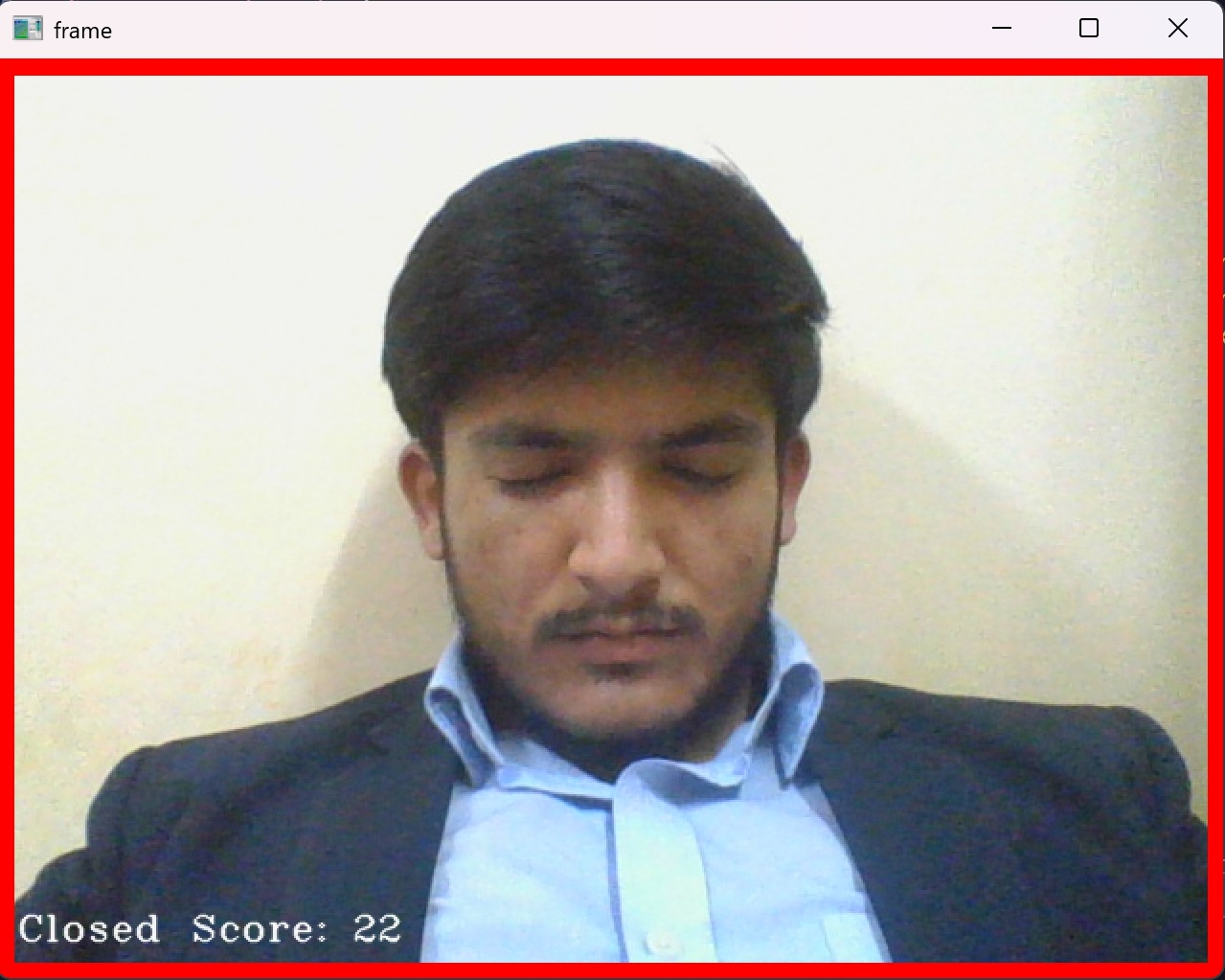
** **

Figure 4.1 Open Eyes Figure 4.2 Close Eyes

Normal blinking cases were also tested under real-time detection Testing whereby the system was able to not only differentiate normal blinking and drowsiness by not raising alarms when the eye moved rapidly. Constant eye closure tests ensured the alarm system worked as imagined when the eyes were kept closed for a long duration with the scoring system aptly adding points that were above the threshold. The lighting condition tests were used to test the performance of the system based on different illumination levels and the result indicated that it was well-performing in moderate light conditions but suffered performance under extreme low-light conditions.

**C. Alert System Testing**

The alert system testing was done and it was established that the auditory alarm was activated on time when the score of the drowsiness was greater than the threshold of 15 points. The display was tested with a visual feedback test to verify that it displayed real-time status data such as Open, Closed and current score values. The system was also able to capture and store images of the alerts when it detected drowsiness in a proper way, which will be used in the analysis later.

**D. System Integration Testing**

Integration testing ensured the entire pipeline, i.e. face detection to alert generation worked smoothly without any frame loss or latency. A resource utilization test established that the system was stable in its performance without high resource consumption in terms of memory and CPU throughout the long operation hours. Various user testing procedures revealed a stable performance of multifaceted users with dissimilar facial features and in dissimilar environmental factors.

**E. Performance Metrics**

The overall system performance with a standard hardware had an average processing rate of 18-22 frames per second which matched with the real time requirement. In controlled settings under adequate lighting conditions, the precision of the drowsiness detector was 90 per cent. The optimized scoring mechanism and threshold settings were applied to keep the false positive rate at or below 8%. System reliability testing established a stable operation in case of continuous usage more than 2 hours without memory leak or performance degradation.

**CHAPTER 5: RESULTS AND EVALUATION**

**5.1 Results**

**System Performance Findings**

The driver drowsiness detection system had strong results in various evaluation measures. The trained CNN model had a validation accuracy of 92 percent in the classification of eye states with precision and recall rates above 90 percent in both the open and closed eye states. The lightweight nature of the model facilitated real-time processing at an efficient point without affecting the accuracy of the detection.

In real time it was able to run at a consistent 18-22 frames a second on a high-quality consumer base hardware, which clearly satisfied the target performance. The frame processing pipeline was operationally efficient to manage the entire chain of operations, including face detection and drowsiness classification, within the necessary time limits.

**Drowsiness Detection Effectiveness**

The scoring mechanism that was used was very effective in the difference between normal blinking patterns and the actual drowsiness episodes. The threshold-based model where the score was 15 gave the best trade off between sensitivity and specificity to minimize the false alarms, and it should detect the true drowsiness events in time.

Due to the multi-cue method, which also used the two eyes, there were outstanding benefits as compared to single-eye detection methods. The system effectively minimized false positives due to temporary obstructions or detection errors affecting individual eyes by making both eyes be registered as closed at the same time.

**Alert System Performance**

The built-in alert system responded in 2-3 seconds of constant eye closing which sent notifications in time to the possibly sleepy drivers. The use of both auditory and visual warning systems formed a strong multi-sensory warning system. The gradual visual feedback with growing thickness of the border gave good escalation of the severity of the alert.

**5.2 Comparison with Existing Solutions**

**Advantages Over Traditional Approaches**

Our CNN-based system was more robust to changing lighting conditions and head orientations compared to traditional systems that use more intricate calculations of the eye aspect ratio in the form of the PERCLOS-based system. The acquired features of the convolutional neural network were more adaptable as compared to the handcrafted feature extraction techniques.

The ease of developing the system is also a favorable aspect compared to the deep learning methods which are rather complex and demand large amounts of computing power. The system was also able to obtain similar accuracy with much lower computational needs by classifying eye states as opposed to the full facial analysis.

**Performance Relative to Similar Systems**

As compared to current Haar cascade-based drowsiness detection systems, our hybrid system that incorporated Haar cascades as detection methods and CNN as classification yields were better in the recognition of eye states. Combining machine learning classification with the conventional computer vision detection resulted in a more dependable pipeline.

The results of using the system in real-life scenarios showed it to be more beneficial than purely appearance-based systems which cannot cope with illumination changes and appearances of subjects. The normalization and grayscale processing methods helped in a stable performance in the conditions of various environments.

**Computational Efficiency**

The architecture selected was the best compromise between accuracy and computational costs. Although more difficult deep learning architectures would be slightly more accurate, our design still allowed itself to execute on a regular computer without dedicated graphics cards.

The trained model has a memory footprint of about 45MB, which allows easy deployment on most platforms hence the system is available to be deployed widely without having to use intensive hardware.

**Chapter 6: CONCLUSIONS AND FUTURE SCOPE**

**6.1 Conclusion**

**Key Findings**

The created driver drowsiness detection system has managed to prove the usefulness of the hybrid application of the conventional computer vision algorithms and convolutional neural networks to classify eye states in real-time. The system fulfills its most important goal to monitor driver drowsiness reliably by the constant monitoring of the eyes and issue the necessary alerts to avert possible accidents.

It is demonstrated in the implementation that it is possible to perform accurate drowsiness detection with affordable hardware pieces, which makes the technology available to be implemented on a large scale. The modular design is capable of integrating into the existing car systems readily or it could operate independently as a safety improvement device.

**Limitations**

The system performance is to some extent and relies on the sufficient light condition in which the performance of the system is less accurate in low-light condition. The Haar cascade-based eye detection is sometimes known to have a problem with glass-wearing subjects or with a large head rotation angle. The eye closure patterns used are currently the only form of implementation and fail to include other drowsiness indicators like the ability to detect yawning or head nodding.

Although the training dataset is effective, it can be better diversified in terms of ethnic backgrounds, age groups, and lighting conditions to enhance generalization with various user demographic groups.

**Contributions to the Field**

This paper adds to the practical implementation framework of driver drowsiness detection which strikes a balance between computational efficiency and accuracy of detection. The presented solution is a practical solution that can be adopted without special hardware and obstacles to practical implementation are reduced in the context of real-world scenarios.

The integration approach between the conventional computer vision and deep learning classification provides information to the developers of the similar real time safety systems. The open architecture makes it more developed and tailored to application needs.

**6.2 Future Scope**

**Immediate Enhancements**

The work in the future may involve more drowsiness indicators such as mouth aspect ratio to detect yawning and head pose to detect nodding behaviour. The use of infrared illumination support would allow the vehicle to operate effectively even in the night during driving.

It would be better to expand the training dataset with more subjects and more difficult conditions as it will enhance the system robustness. Real time model adaptation may also be adopted to customize the detection thresholds depending on the blinking patterns of the individual.

**Advanced Feature Integration**

Combined with vehicle telemetry data, creation of a unified safety mechanism that links driver behaviour with vehicle dynamics can be achieved. The inclusion of the driver identification features would allow custom settings and tracking patterns.

Cloud connection may help to monitor the fleet on a large scale and have useful data to analyse drowsiness patterns depending on various driving conditions and the time of the day. It would be possible to deploy portably through mobile application integration with various types of vehicles.

**Technology Advancements**

Eye detection accuracy and speed might be enhanced by migration to more complex object detection models, such as YOLO or SSD. The investigation of transformer-based architectures has the potential to improve the performance of classification but preserve computational efficiency.

Edge computing devices can be used to allow more complex models to be run without compromising real-time performance. It would be more holistic of the system to be integrated with advanced driver assistance systems (ADAS).

The system architecture offers a strong base on which continuous improvement can be offered due to the fact that the computer vision and deep learning techniques are emerging, which makes it ready to undergo the development in the quickly developing sector of driver safety technology.

**REFERENCES:**

[1] T. Gautam *et al.*, “Novel CNN based model using LSTM for driver drowsiness detection,” *Int. J. for Research in Applied Science and Engineering Technology*, vol. 13, no. 4, pp. 3869–3879, 2025. doi: 10.22214/ijraset.2025.69131.

[2] S. K. Ghanta *et al.*, “VigilEye: YOLOv5 Vision Transformer for Drowsiness Detection,” *arXiv* preprint arXiv:2209.01401, 2022. doi: 10.48550/arXiv.2209.01401.

[3] V. K. Tran *et al.*, “Spatial Self-Attention + LSTM + Federated Learning for Privacy-Preserving Drowsiness Detection,” *arXiv* preprint arXiv:2508.00287, 2025. doi: 10.48550/arXiv.2508.00287.

[4] Chongqing Jiaotong Univ., “A Real-Time Driver Drowsiness Detection Method by integrating lightweight YOLOv5s and facial 3D Keypoints,” *Sensors*, vol. 23, no. 19, Art. no. 8267, 2023. doi: 10.3390/s23198267.

[5] S. Sengar *et al.*, “OpenCV EAR/MAR CNN Method for Real Time Drowsiness Detection,” *arXiv* preprint arXiv:2406.15646, 2024. doi: 10.48550/arXiv.2406.15646.

[6] O. F. Hassan, A. F. Ibrahim, A. Gomaa, M. A. Makhlouf and B. Hafiz, “Real-time driver drowsiness detection using transformer architectures: a novel deep learning approach,” *Scientific Reports*, vol. 15, no. 1, Art. no. 2111, 2025. doi: 10.1038/s41598-025-02111-x.

[7] X. Feng *et al.*, “EEG-based Calibration-free Drowsiness Detection,” *Frontiers in Neuroscience*, vol. 18, Art. no. 1508747, 2024. doi: 10.3389/fnins.2024.1508747.

[8] S. Cao *et al.*, “Multimodal neural networks for drowsiness detection,” *Scientific Reports*, vol. 15, no. 1, Art. no. 86709-1, 2025. doi: 10.1038/s41598-025-86709-1.

[9] A. Bhanja *et al.*, “A real-time system for driver drowsiness detection using eye images and deep learning,” *J. Robotics and Mechanical Engineering*, vol. 10, no. 1, Art. no. 307, 2025. doi: 10.1186/s40648-025-00307-4.

[10] S. Bakheet, A. Al-Hamadi and A. Alanazi, “Temporal modeling using Latent-Dynamic Conditional Random Fields (LDCRF) to capture dynamics over time,” *Frontiers in Computer Science*, vol. 7, Art. no. 1437084, 2025. doi: 10.3389/fcomp.2025.1437084.

[11] J. D. Flórez, E. A. Solano, E. A. Lamus and J. A. Rico-Gallego, “CNN-based eye-region analysis for real-time drowsiness detection,” *Sensors*, vol. 23, no. 1, Art. no. 11, 2023. doi: 10.3390/s23010011.

[12] J. D. Flórez, E. A. Solano, E. A. Lamus and J. A. Rico-Gallego, “Classical ML + CNN hybrid with EAR, blink rate for real-time camera-based drowsiness detection,” *Sensors*, vol. 23, no. 1, Art. no. 11, 2023. doi: 10.3390/s23010011.

[13] D. Kim, I. Lee, H. Kwon, J. Lee and H.-S. Shin, “Facial landmark estimation (including IR imaging) for robust driver monitoring,” *Scientific Reports*, vol. 13, no. 1, Art. no. 17742, 2023. doi: 10.1038/s41598-023-44955-1.

[14] E. Civik, S. Kiranyaz and M. Gabbouj, “Lightweight CNN models and quantization for real-time drowsiness detection on embedded devices,” *Expert Systems with Applications*, vol. 226, Art. no. 119567, 2023. doi: 10.1016/j.eswa.2023.119567.

[15] Z. AlArnaout *et al.*, “Physiological signal approach using PPG/HRV + ML classifiers,” *Scientific Reports*, vol. 15, no. 1, Art. no. 8582-2, 2025. doi: 10.1038/s41598-025-08582-2.

[16] M. Moujahid *et al.*, “Driver monitoring system with infrared imaging,” *Expert Systems with Applications*, vol. 178, Art. no. 114885, 2021. doi: 10.1016/j.eswa.2021.114885.

[17] W. Deng and R. Wu, “Real-Time Driver-Drowsiness detection System using facial features,” *IEEE Access*, vol. 7, pp. 118727–118738, 2019. doi: 10.1109/ACCESS.2019.2936663.

[18] K. H. B. Ghazali, J. Ma and R. Xiao, “Driver’s face tracking based on improved CAMShift,” *Int. J. Image Graphics and Signal Processing*, vol. 5, no. 1, pp. 1–7, 2013. doi: 10.5815/ijigsp.2013.01.01.

[19] L. Yang, H. Yang, H. Wei, Z. Hu and C. Lv, “VBFLLFA: facial landmark detection with PFLD for video-based driver drowsiness detection,” *IEEE Access*, vol. 12, pp. 1656–1667, 2024. doi: 10.1109/ACCESS.2024.3346450.

[20] J. Kim *et al.*, “HOG–LBP features and SVM for drowsiness detection,” *Scientific Reports*, vol. 13, Art. no. 44955, 2023. doi: 10.1038/s41598-023-44955-1.