Faculty of Engineering

Department of Computer Engineering and Software Systems

CSE491: Computer & Systems Engineering Graduation Project (1)

DRIVER DROWSINESS DETECTION USING DEEP LEARNING

Prepared by:

- Seif Eldin Amr Mostafa Yassine (20P2006)

Submitted to:

- Prof. Mahmoud Khalil

Abstract

Driver drowsiness is a major contributor to road accidents and fatalities worldwide. Existing research identifies four main categories of drowsiness detection methods: physiological measures (e.g., EEG), vehicle-based measures (e.g., steering behavior), subjective self-reporting, and behavioral measures (e.g., facial features). Among these, behavioral approaches offer a non-intrusive and practical solution for real-time monitoring.

This project presents a drowsiness detection system based on eye state analysis using deep learning. Three desktop-based implementations were developed: OpenCV with Haar cascades, Dlib with facial landmarks, and MTCNN for robust face and eye detection. A convolutional neural network was trained to classify eye states and later enhanced using transfer learning and data augmentation to improve accuracy and generalization.

The final model was deployed in a Flutter-based mobile application using TensorFlow Lite for on-device inference. The system processes live camera input, monitors the eyes, and calculates a rolling score to trigger an alarm when drowsiness is detected. It operates entirely offline and is optimized for low-latency mobile environments. This project demonstrates an effective combination of machine learning, computer vision, and mobile development to deliver a real-time, accessible driver monitoring system that enhances road safety.

Keywords: Drowsiness Detection, Deep Learning, Eye State Classification, TensorFlow Lite, Mobile Application, Transfer Learning, Behavioral Measures

Table Of Contents

List Of Figures:

List Of Tables:

CHAPTER ONE: INTRODUCTION

1.1 Motivation

1.1.1 Human Psychology with Current Technology

Humans have always invented machines and devised techniques to ease and protect their lives, whether for mundane activities like traveling to work or for more ambitious purposes like aircraft travel. With advancements in technology, modes of transportation kept evolving, and our dependency on them began increasing exponentially. Transportation has greatly influenced our lives as we know them. Today, we can travel to places at a pace that even our grandparents wouldn't have thought possible.

In modern times, almost everyone uses some form of transportation daily. Some people are fortunate enough to own vehicles, while others rely on public transportation. However, there are universal rules and codes of conduct for those who drive, regardless of social status. One critical rule is to stay alert and active while driving. Neglecting this responsibility has led to countless tragedies being associated with this otherwise wonderful invention every year.

While it may seem trivial to some, following traffic rules is of utmost importance. On the road, an automobile wields significant power, and in irresponsible hands, it can become destructive. This carelessness can harm not just the driver but also innocent pedestrians and other road users. A common yet dangerous form of negligence is driving while fatigued.

To monitor and prevent destructive outcomes from such negligence, researchers have extensively studied driver drowsiness detection systems. However, at times, these systems yield observations and results that are not entirely accurate. To address these shortcomings, provide fresh data, and explore another perspective on the issue, this project has been undertaken.

1.1.2 Facts & Statistics

Current statistics reveal alarming numbers. In 2015 alone, 148,707 people in India lost their lives in car-related accidents. Of these, at least 21% were attributed to fatigue-induced errors by drivers. This percentage might be an underestimate, as the role of fatigue is often overlooked among the many causes of accidents.

Fatigue, combined with poor infrastructure in developing countries like India, creates a recipe for disaster. Unlike alcohol or drugs, fatigue is difficult to measure or observe since it lacks clear indicators and readily available tests. Addressing this issue requires raising awareness about fatigue-related accidents and encouraging drivers to acknowledge when they are fatigued. However, this is a complex challenge.

Driving long hours is often lucrative, particularly in jobs requiring overnight transport. The financial incentives motivate drivers to make unwise decisions, like driving through fatigue, unaware of the immense risks involved. While some countries have imposed restrictions on the number of consecutive hours a driver can work, implementation remains challenging and costly.

1.2 Problem Statement

Despite advancements in automotive safety systems, there is a lack of accessible and efficient drowsiness detection solutions for everyday drivers. Existing systems are either too costly or inefficient in detecting drowsiness early enough to prevent accidents.

1.3 Outline

This document outlines the key components of the driver drowsiness detection system, including literature review, neural network architecture, solution design, software requirements, and prototype results.

CHAPTER TWO: LITERATURE REVIEW (Shaik, 2023)

The purpose of the literature review is to provide a comprehensive understanding of existing research and advancements in the field of driver drowsiness detection systems. It aims to examine the methodologies, technologies, and findings from previous studies to identify strengths, limitations, and gaps in the current body of knowledge. This review will establish a foundation for the proposed solution by analyzing behavioral, physiological, and vehicle-based detection techniques, highlighting their effectiveness and challenges. Ultimately, it seeks to contextualize the project within the broader field, justify its significance, and guide the development of an optimized system to enhance road safety.

2.1 Existing Research on Driver Drowsiness Detection (Md. Ebrahim Shaik, 2021)

Substantial efforts are needed to reduce the consequences of drowsiness, as it plays a significant role in road crash fatalities and injuries. Road accidents account for an estimated 1.35 million fatalities and 20–50 million injuries annually, averaging 27.5 fatalities per 100,000 people worldwide (Shaik et al., 2021). Driving while drowsy poses a severe risk to road safety. According to the National Highway Traffic Safety Administration (NHTSA), drowsy driving caused 4,121 fatal crashes and 3,662 injuries between 2011 and 2015. This represents 2.4% of all fatal crashes and 2.5% of all crash fatalities in the USA during that period.

Drowsy driving accidents, particularly on high-speed expressways, tend to be more severe due to the nature of the roads. Drivers often become tired after prolonged driving, which significantly increases the likelihood of fatal crashes. Additionally, insufficient sleep among drivers is a major contributing factor to these accidents (Zhu et al., 2021). The data underscores the critical need to address driver fatigue as a leading cause of traffic accidents. Advances in technology capable of warning drivers about their drowsy state could significantly reduce fatalities and injuries related to traffic accidents.

Driver drowsiness jeopardizes the safety of drivers, passengers, and pedestrians. Its detection is challenging, making it essential to develop reliable methods for identifying and predicting drowsy driving to enhance transportation safety. Various features can be leveraged in driver drowsiness detection systems. Behavioral data, such as eye, face, and head movements, physiological parameters like ECG, EEG, EOG, and heart rate, and vehicle-based data including steering wheel movement, vehicle speed, braking style, and lane position variation, are commonly used.

2.2 Methods for measures driver drowsiness

Different techniques have been employed by researchers to gauge driver drowsiness. The process of detection can be carried out by using behavioral data, physiological characteristics, subjective measurements, and data collected from the vehicle. In addition to these three, researchers have also employed subjective methods, in which drowsy driving is assessed directly or by completing a questionnaire by the driver. The various measurements and the associated parameters are discussed in this section. The brief classification of driver drowsiness measures is displayed as follows.

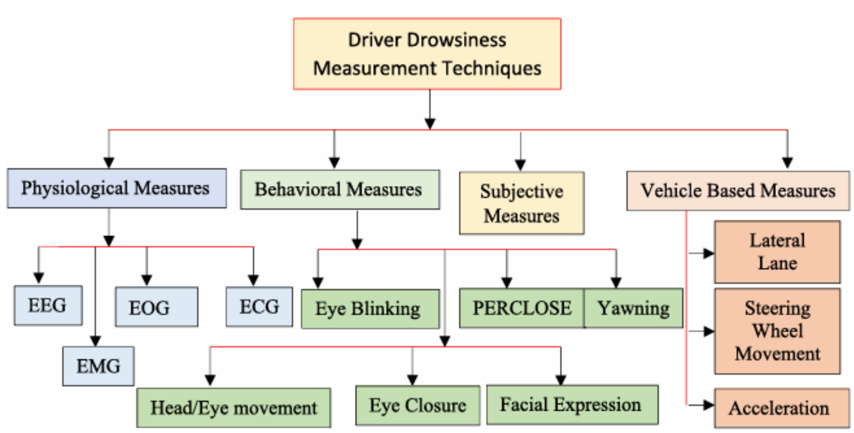

Figure 1 Major Approaches to measure DD

2.2.1 Physiological measures

Physiological methods for driver drowsiness detection involve monitoring key bodilysignals using sensors and electronic devices attached to the driver. These methods focus on physiological variables such as heart rate, pulse rate, brain activity, and body temperature. The primary signals used in this approach are:

I. **Electroencephalography (EEG):**
   - Measures brainwaves to detect states of arousal, including wakefulness, drowsiness, and sleep.
   - Widely used for diagnosing sleep disorders and epilepsy due to its high accuracy and reliability.

II. **Electrooculography (EOG):**
   - Tracks the cornea-retinal potential to monitor eye movement.

III. **Electrocardiography (ECG):**
   - Records the electrical activity of the heart to assess heart rate variations, which differ significantly between alertness and drowsiness.

Research Findings:

- **Hybrid Techniques:** Studies show that combining multiple traits yields better accuracy than singular approaches (Hasan et al., 2021).
- **Non-Contact Sensors:** A hybrid system using non-contact sensors demonstrated early and accurate detection of mild drowsiness (Gwak et al., 2020).
- **Wearable EEG-Based CNN Systems:** Recent studies propose wearable EEG systems integrated with convolutional neural networks (CNN) for detecting drowsiness in moving vehicles, showing high effectiveness (Zhu et al., 2021).

Summary of previous studies on driver drowsiness detection using physiological measures.

Author	Classification Method	Signal	Subjects/ Participants	Performance
Gao et al. (2019) (Gao et al., 2019)	RN-CNN	EEG	10 right-handed healthy students, 8 males, 2 females, average age: 23.3 years	Accuracy = 92.95% SD = 4.39
Cui and Wu, (2017) (Cui and Wu)	CNN	EEG	16 healthy subjects, normal vision	RMSE = 0.2347
Hajinoroozi et al. (2015) (Hajinoroozi et al., 2015)	CCNN, CNN-R	EEG	70 sessions, 37 subjects	Accuracy = 82.94%
Abbas, (2020)(Abbas, 2020)	CNN, DBN	ECG	56 people, 5880 images	Accuracy = 94.50%
Wang et al. (2015) (Wang et al., 2015)	Pulse coupled neural network (PCNN)	EEG	20 healthy male subjects, mean age: 37.5 years	
Chaabene et al. (2021) (Chaabene et al., 2021)	CNN	EEG	14 active electrodes, 2 reference electrodes	Accuracy = 90.42%
Ma et al. (2016) (Ma et al., 2016)	NN, Fuzzy logic, SVM, ARIMA	EOG	0.5-second-ahead EOG signal behavior	0.5 s, Ahead Prediction
Liu et al. (2016) (Eskandarian and Mortazavi, 2017)	RSEFNN	EEG	10 healthy young adults participants, average age: 24.2 ± 3.7 years	RMSE = 0.0840 ± 0.0285
Zhu et al. (2021) (Zhu et al., 2021)	CNN	EEG	69,054 samples, awake period = 33,035; drowsiness period = 36,019	Accuracy = 94.68%
Gao et al. (2019) (Gao et al., 2019)	Spatio–Temporal CNN	EEG	800 undergraduates (5 males, Females, age: 19 to 26	Accuracy = 97.37%
Kim and Shin, (2019) (Kim and Shin, 2019)	SVM, KNN, RF	HRV	37 recordings, 6 subjects, 5 males, 1 female, ages: 25 to 35.	AUC = 0.95
Becerra-Sánchez et al. (2019) (Becerra-Sánchez et al., 2019)	SVM, KNN, LR	EEG		Accuracy = 93.00%

Figure 2 Summary of previous physiological measures studies

2.2.2 Behavioral Measures (Bajaj, 2023)

Behavioral measures involve analyzing a driver's observable physical actions, such as facial expressions, eye movements, and head posture, to detect signs of drowsiness. These measures rely on capturing visual and motion-related data using cameras and sensors.

Key Behavioral Indicators (Safarov, 2023)

1. **Eye-Related Features:**
    - Eye closure, blinking rate, and duration of eye closure are significant indicators of drowsiness.
    - Monitoring eye movements using visual techniques or eye-tracking devices helps assess driver alertness.
2. **Facial Expressions:**
    - Yawning, drooping eyelids, and other facial cues are commonly analyzed to identify fatigue.
    - Techniques such as facial landmark detection and emotion recognition are applied.
3. **Head Movements:**
    - Changes in head posture, such as nodding or leaning, often signal fatigue.

Research Findings:

- **Accuracy and Effectiveness:**
    - Behavioral measures are effective in detecting early signs of drowsiness, particularly when combined with other modalities.
    - They are non-intrusive and provide real-time monitoring without physical contact.

- **Challenges:**
  - External factors like lighting conditions, obstructions (e.g., sunglasses), and individual differences in behavior can reduce accuracy.
  - Behavioral data can be sensitive to noise, making it less reliable in certain scenarios compared to physiological methods.

Applications and Advancements:

- Modern systems use machine learning models to improve the robustness of behavioral detection.
- Real-time video analysis with deep learning algorithms has shown promise in accurately identifying drowsiness.

Authors	Classification Methods	Drowsiness Measures	Dataset	Performance
Guo et al. (2016) (Guo et al., 2016)	Bayesian Network (BN)	Heart rate, pulse rate, eyelid movement, gaze, head movement	21 participants car simulator for 110 min	Accuracy = 79.50%
Bakheet ang Al Hamadi, (2021), (Bakheet and Al-Hamadi, 2021)	Histogram of Oriented Gradient (HOG) features	Driver image, eye pair region	NTHU-DDD dataset	Accuracy = 85.62% F1-Score = 87.84%
Boyraz et al. (2008) (Boyraz et al., 2008)	Fuzzy inference system (FIS) and ANN	Eye closure, pupil area, gaze vector, head motion	Data set of 30 pairs, 1.5 h highway simulation	Accuracy = 98.00%
Park et al. (2017) (Park et al)	Deep Networks	Driver image	NTHU-driver drowsiness detection benchmark video dataset	Accuracy = 73.06%
Huynh et al. (2017) (Huynh et al)	3D CNN	Eye-closing, nodding and yawning	5 different scenarios, Bare Face, Glasses, Night Bare Face, Night Glasses, and Sunglasses cases	Accuracy = 87.46% F1-Score = 87.97%
Weng et al. (2017) (Weng et al., 2017)	Hierarchicaltemporal Deep Belief Network (HTDBN)	Yawning, blink rate, falling asleep	Large dataset, genders, lighting conditions and driving scenarios	Accuracy = 84.82% F1-Score = 85.39%
Bamidele et al. (2019) (Bamidele et al., 2019)	KNN, SVM, Logistic Regression, ANN	Percentage ofeyelid closure (PERCLOS), blink frequency (BF), Maximum Closure Duration (MCD)	Video data, various facial characteristics, different ethnicities, 5 different scenarios	Accuracy = 72.25% Sensitivity = 83.06%
De Naurois et al. (2018) (De Naurois et al., 2018)	ANN	Driving performance, eyelid and head movements	21 participants car simulator for 110 min	Prediction = 40%, Detection = 80%
Vijayan and Sherly, (2019) (Vijayan and Elizabeth, 2019)	CNN	Eye blinking, yawning, head swaying	68 attributes RGB video input of a driver	Accuracy = 78.61%
Chen et al. (2021) (Chen et al., 2021)	LSTM, CNN	Eye, face area	THU-DDD dataset	Accuracy = 93.30%
Han et al. (2015) (Han et al., 2015)	vision based, feature extraction	PERCLOS and blink rate	Drivers face video data, 8 subjects, 30 min. driving simulator	Accuracy = 90.45% Correlation Coefficient = 0.91
Zhang et al. (2012) (Zhang et al., 2012)	Computer Vision Technology	Eyelid closure, blink frequency, opening and closing velocity of the eyes	06 participants, driving simulator experiments	Accuracy = 86.00%
Flores et al. (2010) (Flores et al., 2010)	Condensation algorithm (CA), NN	Face and eye image	Real-time data, grayscale images	Accuracy = 98.00%

Figure 3 Summary of previous behavioural measures studies

2.2.3 Vehicle-Based Measures

The way a drowsy driver operates a vehicle may differ significantly from the way a regular driver operates a vehicle. Any change in these measurements that exceeds a certain threshold denotes a greatly increased likelihood that the driver is sleepy. These metrics include deviations from lane position, steering wheel movement, pressure on the accelerator, etc. (Sahayadhas et al., 2012).

Key Metrics and Techniques:

I. **Lateral Lane Position:**

- A critical metric often used in detecting driver drowsiness.
- External cameras are commonly employed to track the vehicle's lateral position during field trials.
- Studies:
  - Wang and Xu (2016)
  - Zhang et al. (2020)
  - Gwak et al. (2020)
  - Forsman et al. (2013)

II. **Steering Wheel Movement:**

- A widely utilized vehicle-based measurement.
- Assessed using steering angle sensors positioned on the steering column to monitor the driver's steering behavior.
- Studies:
  - Guo et al. (2016)
  - Boyraz et al. (2008)
  - Arefnezhad et al. (2020)

- Zhang et al. (2020)
- Wang and Xu (2016)
- Gwak et al. (2020)
- Dehzangi and Masilamani (2018

Observations:

- Studies have shown that vehicle-based metrics do not always reliably predict drowsiness-related performance errors.
- Drowsiness is not explicitly defined in these metrics, leading to challenges in standardizing detection techniques.
- However, advancements like unobtrusive vehicle-generated data evaluation have improved diagnostic capabilities.

Summary of previous studies on driver drowsiness detection using Vehicle based measures.

Authors	Classification Methods	Input Parameters	Data collecting technique	Performance
Guo et al. (2016) (Guo et al., 2016)	Bayesian Network (BN)	Lane deviations, steering movements	PSO-based feature selection approach, driving measurements	Accuracy = 95.50%
Boyraz et al. (2008) (Boyraz et al., 2008)	Fuzzy inference system (FIS) and ANN	Steering wheel angle, vehicle speed	Standard highway simulation for 1.5 h, data set 30 pair	Accuracy = 98.00% F1 score = 98%
Arefnezhad et al. (2020) (Arefnezhad et al., 2020)	CNN, RNN	Lateral deviation and acceleration, steering wheel angle	44 sessions fixed-base driving simulator simulating monotonous night-time highway drives	Accuracy = 96.00%
Zhang et al. (2020) (Zhang et al., 2020)	Mixed-effect ordered logit (MOL) model	Vehicle speed, lateral position, steering wheel movement	Karolinska Sleepiness Scale (KSS)	Accuracy = 62.84%
Wang and Xu, (2016) (Wang and Xu, 2016)	Multilevel ordered logit (MOL) model, ordered logit model, ANN	Vehicle speed, lateral position, steering wheel angle	High fidelity motion based driving simulator	Accuracy = 64.15%-68.40%
Quddus et al. (2021) (Quddus et al., 2021)	Long short-term memory (LSTM), CNN	Vehicle driving dynamics	38 subjects, simulated driving experimen	Accuracy = 95%–97%
Gwak et al. (2020) (Gwak et al., 2020)	Support Vector Machine (SVM), K-Nearest Neighbor(KNN)	Vehicle velocity, longitudinal acceleration, lateral position, steering wheel acceleration	driving simulator and driver monitoring system	Accuracy = 95.4%
Dehzangi and Masilamani (2018) (Dehzangi and Masilamani, 2018)	Decision-Tree algorithms	Acceleration, braking, steering wheel	KSS (Karolinska Sleepiness Scale)	Accuracy = 99.10% RMSE = 0.31 MAE = 0.14
McDonald et al. (2018) (McDonalda et al., 2018)	Dynamic Bayesian Network	Speed and acceleration	72 participants driving the National Advanced Driving Simulator	False positive rates < 15%
Forsman et al. (2013) (Forsman et al., 2013)	principal component analysis (PCA)	Lateral lane position, steering wheel angle, driving speed	Two laboratory-based, high-fidelity driving	Correlation r = 0.88

Figure 4 Summary of previous vehicle based measures studies

2.2.4 Subjective Measures

Subjective measures assess the degree of drowsiness based on the driver's self-assessment. These evaluations are converted into quantifiable measures of driver drowsiness using various scales and techniques.

Common Scales:

1. **Stanford Sleepiness Scale (SSS):**
    - A 7-point scale with numerical ratings corresponding to specific verbal descriptions of drowsiness.
2. **Karolinska Sleepiness Scale (KSS):**
    - A 9-point scale where numerical ratings indicate:
        - 1: Extremely alert
        - 9: Extremely sleepy, requiring significant effort to stay awake (Sahayadhas et al., 2012; Lenné and Jacobs, 2016).
    - Widely used in sleepiness studies due to its association with physiological measurements and psychomotor vigilance test (PVT) results (Åkerstedt et al., 2014; Kaida et al., 2006).

Procedure:

- Participants complete brief questionnaires and rate their current state using the scales provided (Mashko, 2017).
- KSS scores have been used to identify individuals who slept extensively or insufficiently.
- Higher KSS scores are observed during simulated night driving compared to on-road night driving (Hallvig et al., 2013).

Key Observations:

1. Clustered KSS Categories (Anund et al., 2013):

- KSS 1–5: Represents alertness.
- KSS 6–7: Indicates the onset of drowsiness.
- KSS 8–9: Represents extreme drowsiness.

2. Performance Based on KSS Ratings:
   - Participants self-rated as:
     - Less drowsy (KSS < 3): Better performance.
     - More drowsy (KSS > 8): Poorer performance (Hallvig et al., 2014).

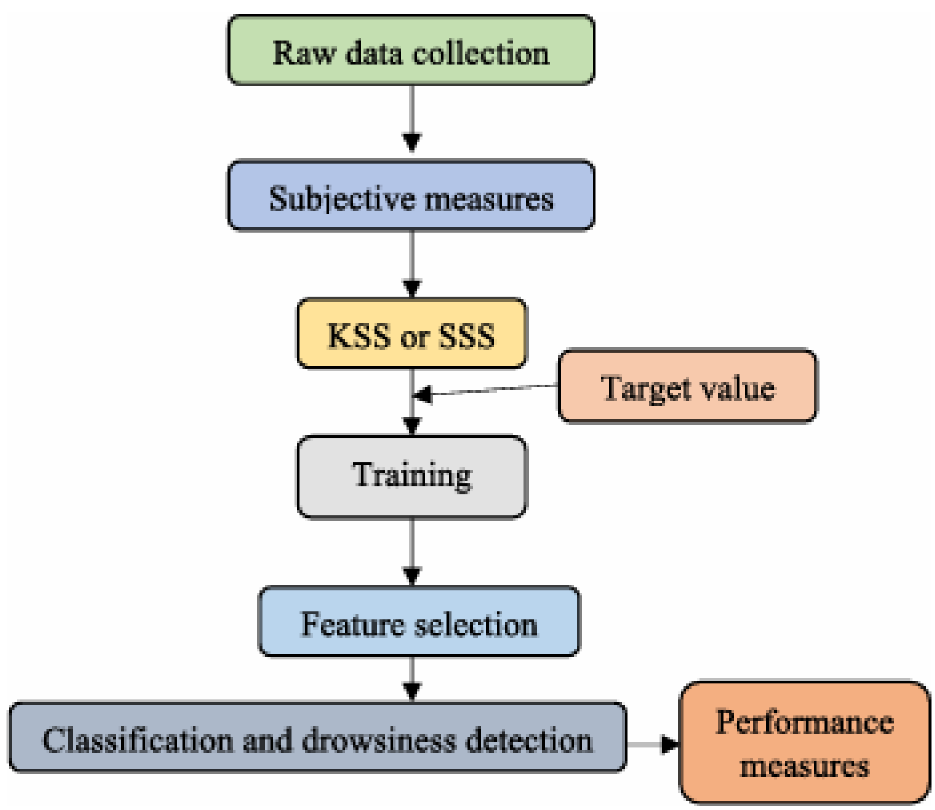

Figure 5 subjective measures activity diagram

CHAPTER THREE: NEURAL NETWORK ARCHITECTURE REVIEW

3.1 Overview of the Model

This project adopts a transfer learning approach for eye state classification, a critical component of the drowsiness detection system. The model is based on a pre-trained Convolutional Neural Network (CNN), chosen for its proven efficiency in visual recognition tasks and adapted to classify eye states (open or closed). Enhancements such as data augmentation, dropout, and batch normalization were employed to improve model generalization and robustness across real-world scenarios.

3.2 Model Architecture

I. **Input Layer**:
   - Input shape: (64, 64, 3) to handle preprocessed RGB images.

II. **Feature Extraction Base:**
   - **MobileNetV2** backbone (excluding the top classification layers), pre-trained on ImageNet.
   - Initially frozen, then partially unfrozen starting from layer 100 during fine-tuning.

III. **Custom Classifier Head:**
   - Global Average Pooling 2D layer
   - Dropout: 30% for regularization
   - Dense Layer: 128 units, ReLU activation
   - Batch Normalization
   - Output Layer: Dense (1 unit), sigmoid activation for binary classification

Table 1 CNN model architecture

Layer Type	**Output Shape**	**Parameters**
Input Layer	(64, 64, 3)	0
MobileNetV2 (frozen)	(None, 2, 2, 1280)	~2.2M (non-trainable)
GlobalAveragePooling2D	(None, 1280)	0
Dropout (0.3)	(None, 128)	0
Dense (128, ReLU)	(None, 128)	~164K
BatchNormalization()	(None, 128)	512
Dense (1, Sigmoid)	(None, 1)	129

Total params: 6,474,181 (24.70 MB)

Trainable params: 2,025,793 (7.73 MB)

Non-trainable params: 396,800 (1.51 MB)

Optimizer params: 4,051,588 (15.46 MB)

3.3 Implementation Details

a) **Preprocessing**:

   - Images resized to 64x64 and normalized to scale pixel values to [0,1].
   - Dataset split into training, validation and testing(64%, 16%, 20%)
   - Augmentations:
     - height_shift_range=0.2
     - shear_range=10
     - rotation_range=20
     - zoom_range=0.5
     - brightness_range=[0.8, 1.2]

b) **Training & Fine-Tuning:**

   - Optimizer: Adam
   - Loss Function: Binary Crossentroy
   - Batch size: 32.
   - Epochs: 50 initial + 30 fine-tuning
   - Callbacks: EarlyStopping, ModelCheckpoint

c) **Callbacks**:

   - Model weights saved using ModelCheckpoint to ensure the best-performing model is retained.

d) **Evaluation**:

   - Confusion matrix and classification report are used for performance analysis using test data.

3.4 Strengths

- Efficient CNN Backbone: Pre-trained MobileNetV2 ensures strong feature extraction with minimal training data.
- **Customizability**: Adjustable hyperparameters like the number of filters, kernel size, and dropout rates allow tuning for better performance.
- **Data Augmentation:** Increases robustness to lighting and pose variations.
- **Integration with Preprocessing**: Model seamlessly integrates with the preprocessing pipeline for RGB image normalization and augmentation.
- **Real-Time Suitability:** Lightweight design enables on-device inference via TensorFlow Lite.

3.5 Summary

This updated CNN architecture based on transfer learning provides a reliable and mobile-compatible solution for detecting driver drowsiness through eye state classification. The integration of data augmentation, regularization techniques, and evaluation metrics ensures the model performs well in practical scenarios and is suitable for real-time applications.

CHAPTER FOUR: SOLUTION ARCHITECTURE

4.1 Overview of the System

The drowsiness detection system is designed to monitor a driver's eye state (open or closed) in real time using computer vision and deep learning. The project evolved through a multi-phase implementation process, starting with three desktop-based systems developed in PyCharm for experimentation and performance comparison, followed by deployment in a mobile application. These implementations detect eye closure and trigger alerts when drowsiness is detected, integrating real-time frame capture, deep learning inference, and user feedback mechanisms.

4.2 System Components

4.2.1 Desktop Implementations

- **System 1: Haar Cascades**
  - Eye and face detection using OpenCV's Haar classifiers.
  - Frame extraction, grayscale conversion, cropping of eyes.
  - Custom CNN used for classification.
- **System 2: Dlib**
  - Facial landmark detection using Dlib's 68-point shape predictor.
  - Precise eye region extraction using landmark points.
  - More accurate and stable than Haar, especially under varied angles.
- **System 3: MTCNN**
  - Uses Multi-task Cascaded Convolutional Networks for robust face and eye detection.
  - Handles occlusions and variable lighting better.
  - Eye region fed into the CNN for classification.

4.2.2 Mobile Deployment (Flutter App)

The final solution was deployed in a **Flutter-based mobile application**, integrating real-time camera processing and on-device inference using **TensorFlow Lite**.

- **Camera Feed**: Captures live video through the front camera.
- **Preprocessing**: Resizes image to 64×64, normalizes pixel values.
- **Model Inference**:
    - MobileNetV2 (fine-tuned) classifies the eye state.
    - A score is computed using a rolling history to detect drowsiness.
- **Alarm Mechanism**: Plays an alarm via phone speaker using the just_audio package.

4.2.3 Communication:

- **User Interface (UI)**: The system includes a graphical user interface or a that shows the status of the system (e.g., "Drowsy" or "Alert"). The UI can also display relevant information such as the time the alert was triggered or the confidence level of the eye-state prediction.
- **External Devices**: The alert system may communicate with external devices, such as speakers or buzzers, to deliver an audible or visual alert to the driver.

4.3 Systems Flow

- **Desktop Systems:**

1. Live camera frame capture via OpenCV.
2. Eye detection using Haar / Dlib / MTCNN.
3. Preprocessing: resize to 64×64, RGB, normalize.
4. CNN predicts eye state.

5. Eye status history evaluated.
6. Alert is triggered if eyes stay closed for a threshold duration.

- **Mobile System:**

1. Frame captured via Flutter camera plugin.
2. Detect faces and extract ladmarks using MLKIT
3. Frames are converted from YUV to RGB format
4. Passed through TensorFlow Lite model.
5. Prediction score updated using sliding window logic.
6. Alarm triggered in case of prolonged closure.

4.4 System Integration

- **Experimental Desktop Pipeline**:
    - Developed in Python using PyCharm.
    - Multiple detection methods benchmarked for accuracy and stability.
    - Used for training/testing CNN and tuning eye-detection pipelines.
- **Mobile Deployment Pipeline**:
    - Model converted to .tflite and integrated with Flutter.
    - Real-time camera processing, eyes detection and alert system added.
    - App runs offline and efficiently on mobile devices.

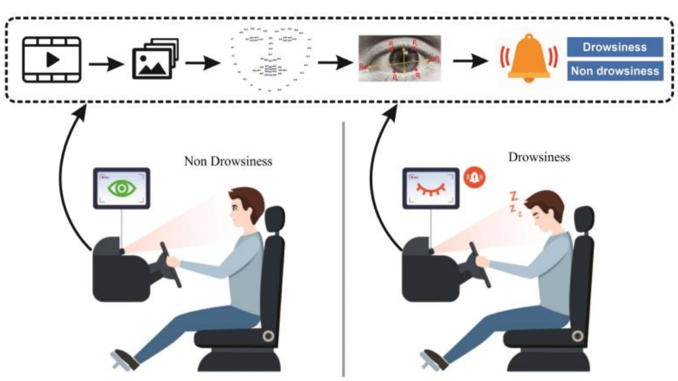

Figure 6 Systems Integration

4.5 Hardware and Software Requirements

- **Hardware**:
  - **Desktop**: Webcam, sufficient CPU/GPU for real-time testing.
  - **Mobile**: Android device with front camera and basic compute capability.
- **Software:**
  - **Python (Desktop):** OpenCV, Dlib, MTCNN, Keras/TensorFlow, PyCharm IDE
  - **Flutter (Mobile):** TensorFlow Lite, just_audio, camera plugin, Dart SDK

4.6 Strengths of Solution Architecture

- **Multi-Stage Development**: Three desktop implementations allowed for iterative refinement and performance testing.
- **Flexible Deployment**: Supports both desktop and mobile use cases.
- **Optimized Model**: Lightweight CNN (via MobileNetV2) works efficiently on mobile devices.
- **Robust Eye Detection**: Evaluated different detection methods (Haar, Dlib, MTCNN) for best performance.

4.7 Areas for Improvement

- **Model Optimization:** Further reduce inference time and model size.
- **Multi-modal Input:** Add head pose, yawning detection, or heart rate sensors.
- **Alert Customization:** Let users select vibration or spoken alerts.
- **Cross-Platform Expansion:** Add support for iOS and embedded platforms.

4.8 Project Timeline

Table 2 Project timeline

Phase	Description	Timeline	Deliverables
Phase 1	Project Initiation	Week 1	- Problem statement- Initial project plan
Phase 2	Data Collection and Preprocessing	Week 1	- Collected dataset- Preprocessed images
Phase 3	Pretrained Model Implementation	Week 1	- Integration of a pretrained CNN model- Initial training results
Phase 4	Functionality Validation	Week 1	- Model evaluation on test data- Metrics and error analysis
Phase 5	Integration with Real-Time Video and Object Recognition Techniques	Week 2	- Real-time video processing pipeline- Object recognition enhancements
Phase 6	Integration of Alert System	Week 2	- Buzzer or speaker alerts triggered by predictions

Phase 7	**Hyperparameter Tuning and Advanced Optimization**	**Week 3**	**- Fine-tuned model- Optimized performance metrics**
Phase 8	**Transfer Learning to Change Model Architecture**	**Week 6**	**- Implementation of models like MobileNet- Comparison and selection**
Phase 9	**Mobile App Deployment**	**Week 9**	**- Model converted to TensorFlow Lite or onnx**

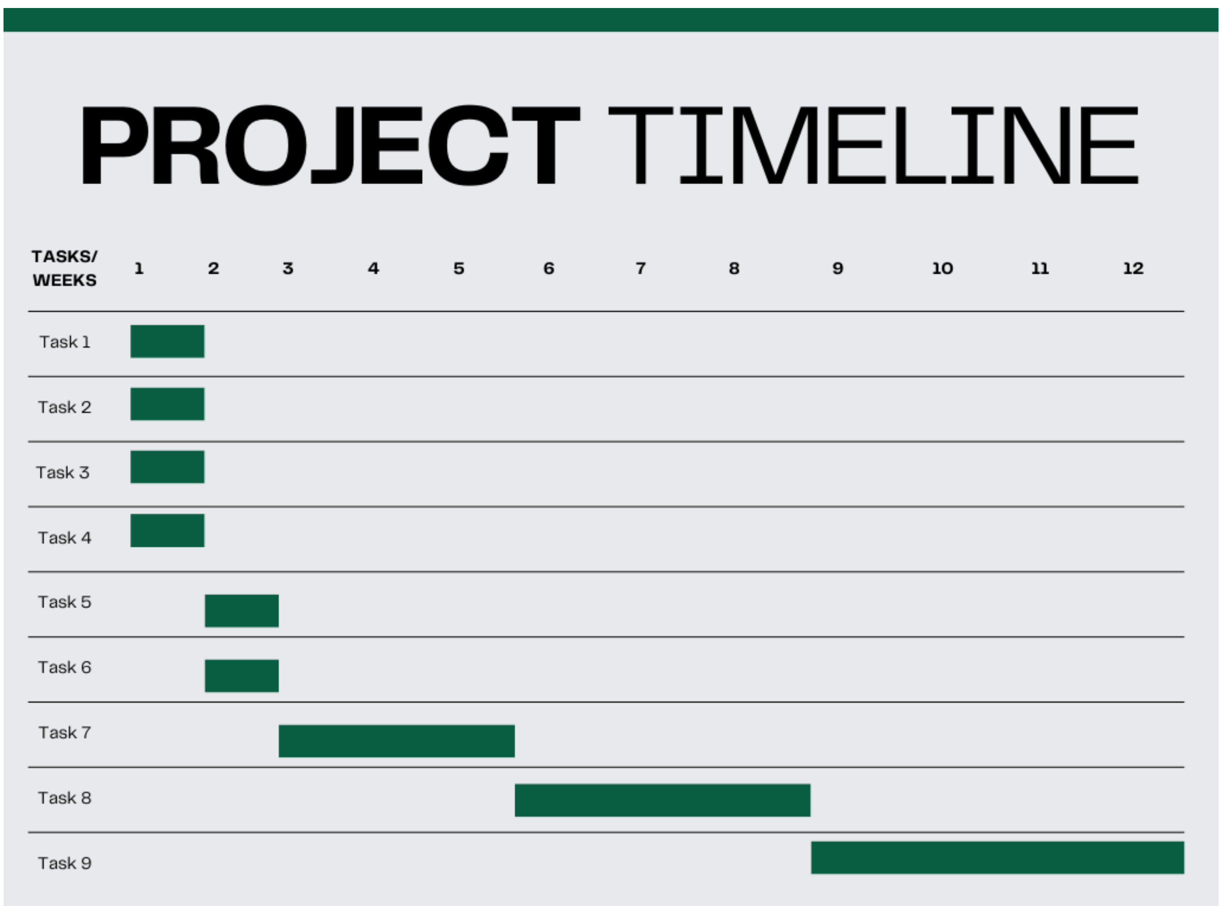

Figure 7 Project Gantt Chart

Chapter Five: Software Requirements Specification

5.1 Functional Requirements

I. **Input Data Processing:**

- The system should accept real-time video streams from the mobile device camera.
- Preprocess video frames to standard dimensions.

II. **Drowsiness Detection:**

- Classify eye states (open or closed) using a Convolutional Neural Network (CNN).
- Trigger an alert for prolonged eye closures exceeding a configurable threshold.

III. **Alert System:**

- Generate a loud sound, vibration, or both as an alert.
- Allow users to customize alert modes and sensitivity.

IV. **User Interface:**

- Provide buttons to start/stop detection.
- Display real-time status (e.g., "Monitoring", "Drowsiness Detected").

V. **Mobile Integration:**

- Deploy the trained model using TensorFlow Lite for mobile compatibility.
- Ensure smooth app performance and low memory usage.

5.2 Non-Functional Requirements

I. **Accuracy:**

   - Maintain a classification accuracy of 90% or higher in desktop applications and 85% or higher in standard conditions.

II. **Usability:**

   - Intuitive and responsive UI designed for non-technical users.
   - Minimal user intervention required during operation.

III. **Reliability:**

   - Continuous operation for up to 8 hours without crashes.
   - Handle varying lighting conditions with minimal performance degradation.

IV. **Portability:**

   - Support Android (11.0 and above)
   - App size should not exceed 100 MB for easier downloads.

5.3 System Models

5.3.1 Use Case:

- **Actors:** User (driver), System (mobile app with CNN model), Alert System.
- **Scenarios:**
  - User starts monitoring via the app.
  - The system processes video frames and predicts eye states.
- If drowsiness is detected, an alert is triggered.
- Use case diagram is represented as follows:

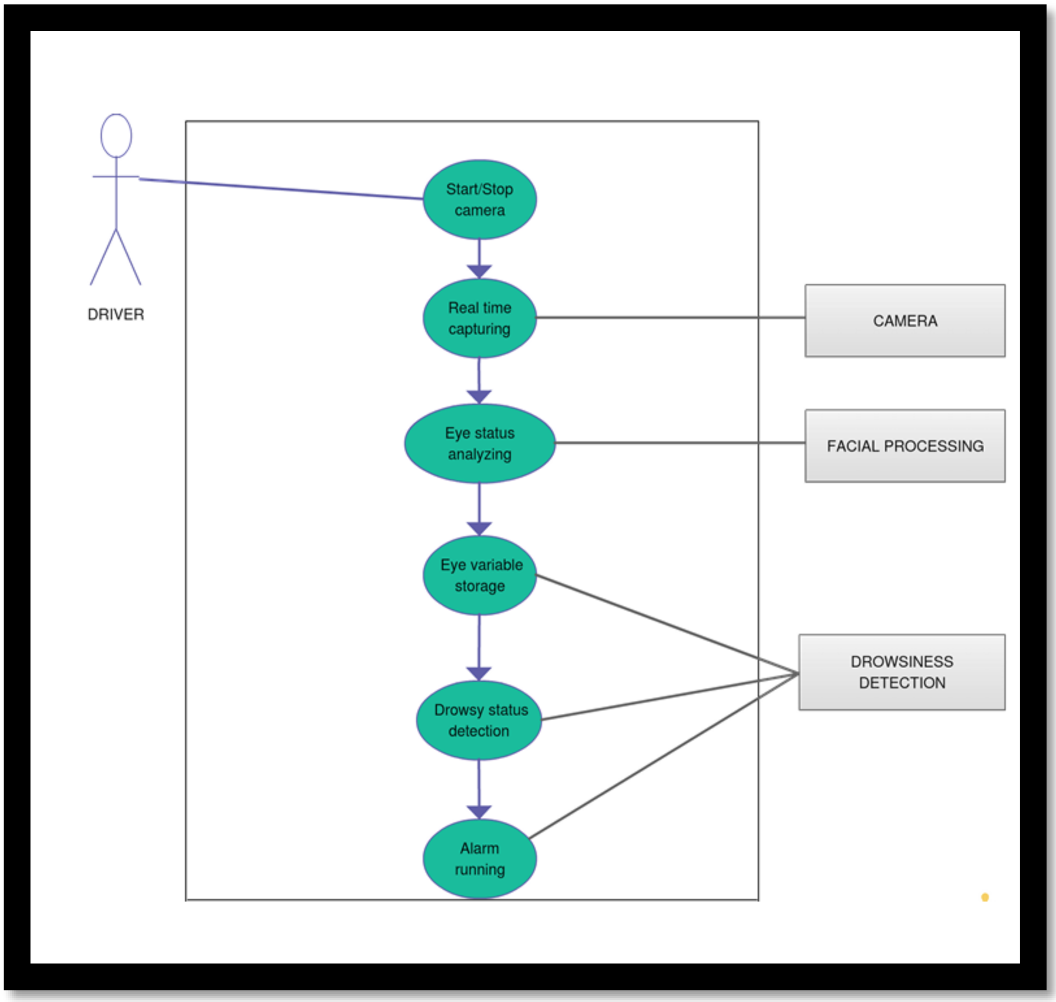

Figure 8 Use Case Diagra

5.3.2 Activity Diagram:

- User initiates video monitoring via the app.
- Frames are preprocessed and passed to the CNN model.
- The model classifies the eye state.
- If drowsiness criteria are met, an alert is triggered.
- Activity diagram is represented as follows:

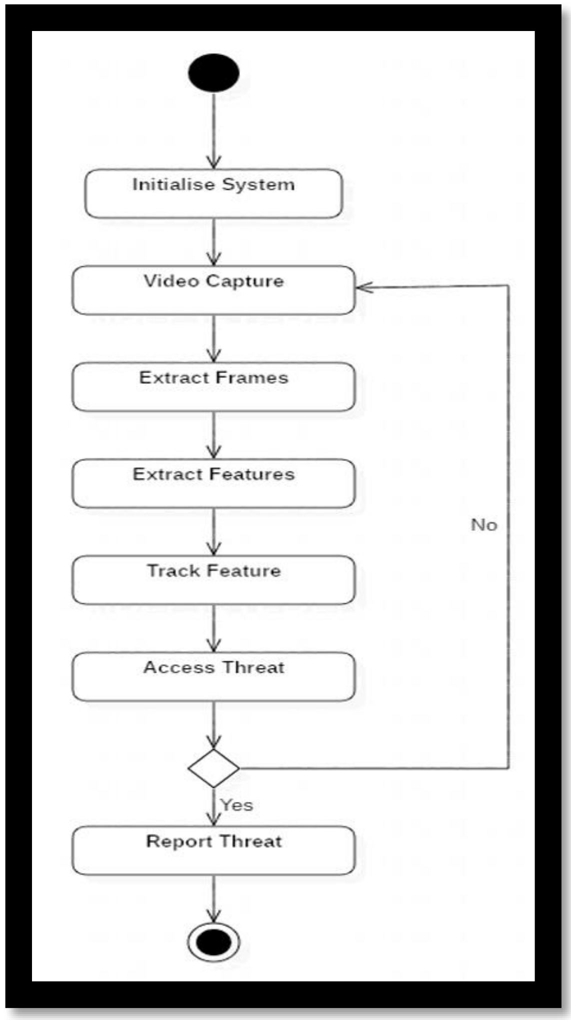

Figure 9 Activity Diagram

5.4 Design Constraints

- **Hardware:**
    - Smartphone with a camera and at least 2GB of RAM.
    - Battery consumption should remain under 15% per hour in operation.
- **Software:**
    - Python: Python 3.9
    - Libraries: Cv2, NumPy, Tensorflow,Pygame, Os, Pandas
    - Mobile platforms (Android and iOS) for mobile app deployment.

CHAPTER SIX: RESULTS & ANALSIS

We have conducted a group of tests to analyze the performance of each of the model, three desktop applications and the mobile application.

6.1 Model Performance Analysis

6.1.1 Confusion Matrix

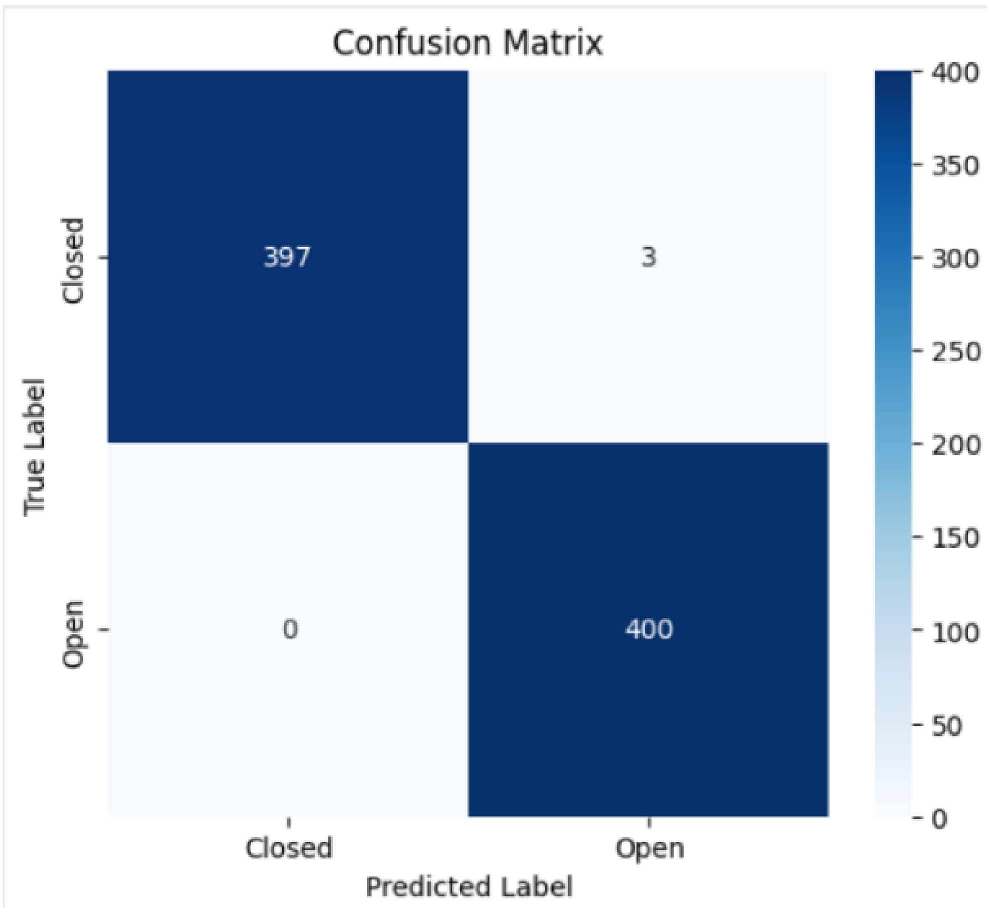

Figure 10 Confusion Matrix

- The model was able to predict correctly 400/400 open eyes.
- The model was able to predict correctly 397/400 closed eyes.

6.1.2 Classification Report

	precision	recall	f1-score	support
Closed	1.00	0.99	1.00	400
Open	0.99	1.00	1.00	400
accuracy			1.00	800
macro avg	1.00	1.00	1.00	800
weighted avg	1.00	1.00	1.00	800

Figure 11Classification report

6.1.3 Training and Validation Accuracy

Fine tuning training starts after epoch 27 which cause sudden change in the values due to the change in learning rates and parameters but it gradually corrects and reach its maximum accuracies which reach to 99.8%.

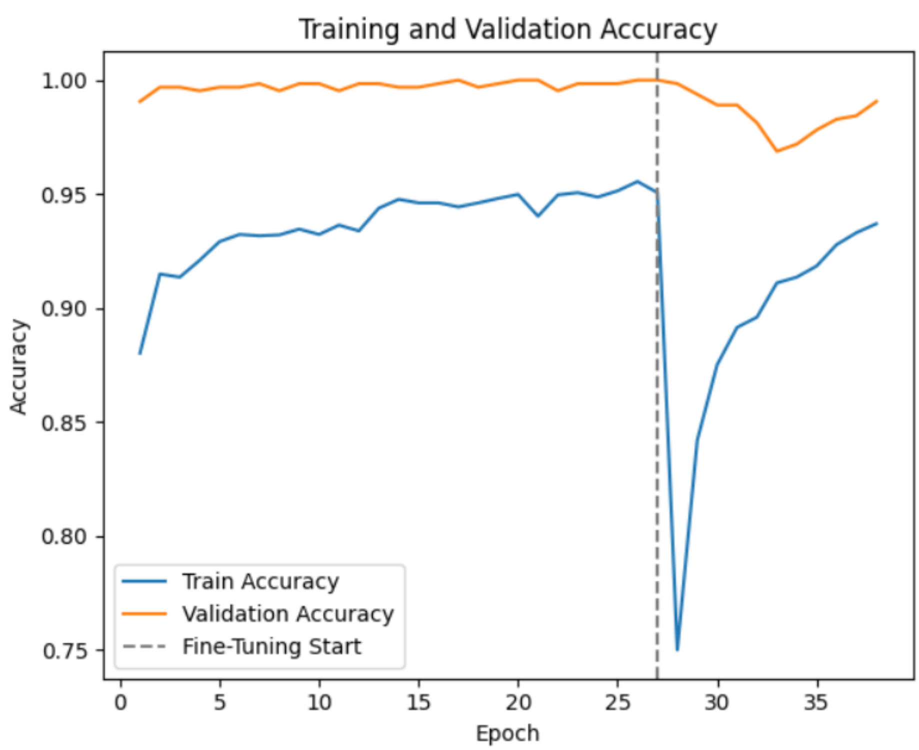

Figure 12 Training vs Validation Accuracy

6.1.4 Training and Validation loss

The same sudden change occurs in the loss but it gradually decrease again till it reaches 0.0062 loss.

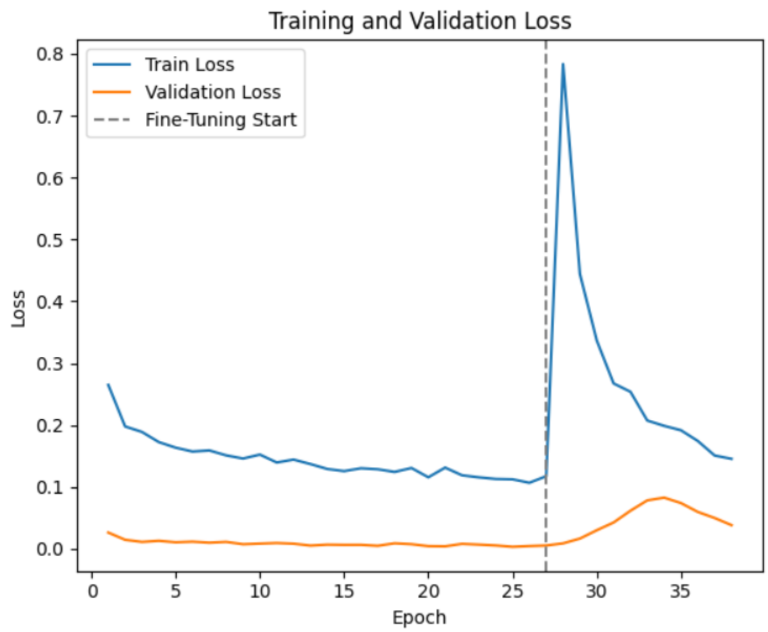

Figure 13 Training vs Validation Loss

6.1.5 Data Augmented Predictions Analysis

Some model prediction on augmented images which correctly predicts all of them

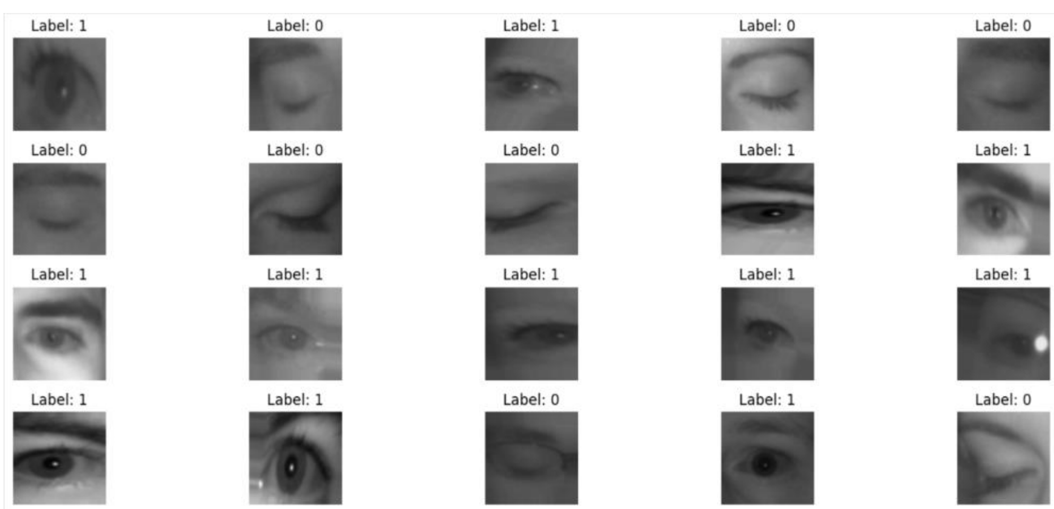

Figure 14 Predictions on Augmented Images

6.2 Desktop System Results

6.2.1 Haar Cascade System

This system uses OpenCV's Haar cascade classifiers to detect the face and eyes. While lightweight and fast, Haar cascades are sensitive to lighting and positioning.

- **Non_Drowsy detection:**

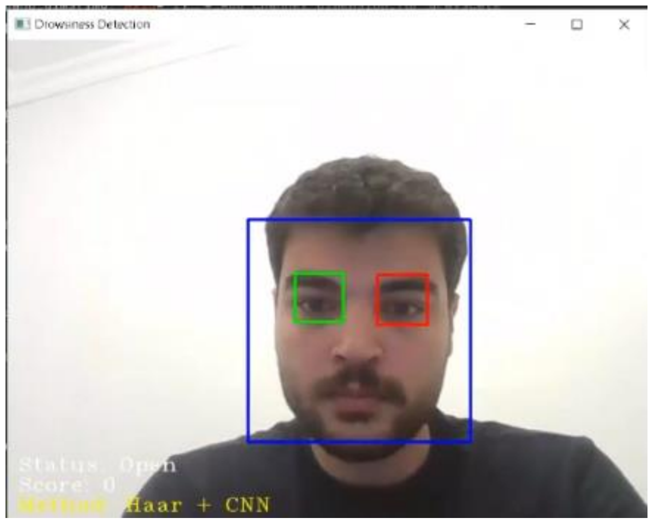

Figure 15 HAAR Non Drowsy

- **Drowsy detection:**

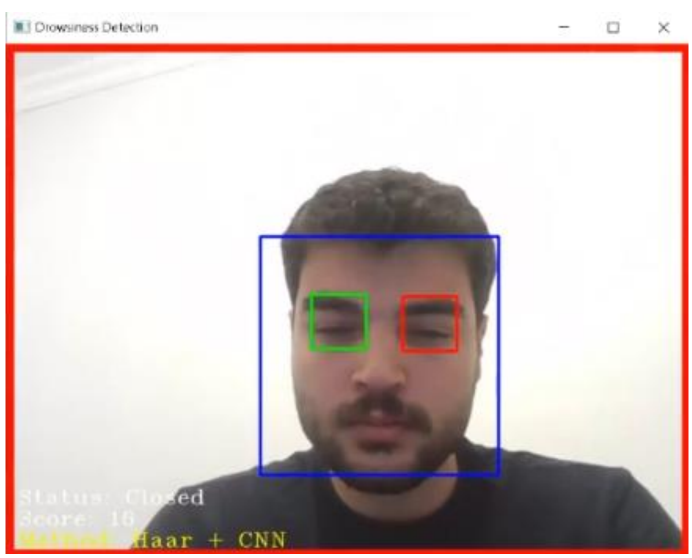

Figure 16 HAAR Drowsy

6.2.2 Dlib Landmark System

Dlib's 68-point facial landmark detector was used for precise and stable eye localization. This method performed better under variable angles and head tilts.

- **Non-Drowsy Detection:** score is 0 and status is open

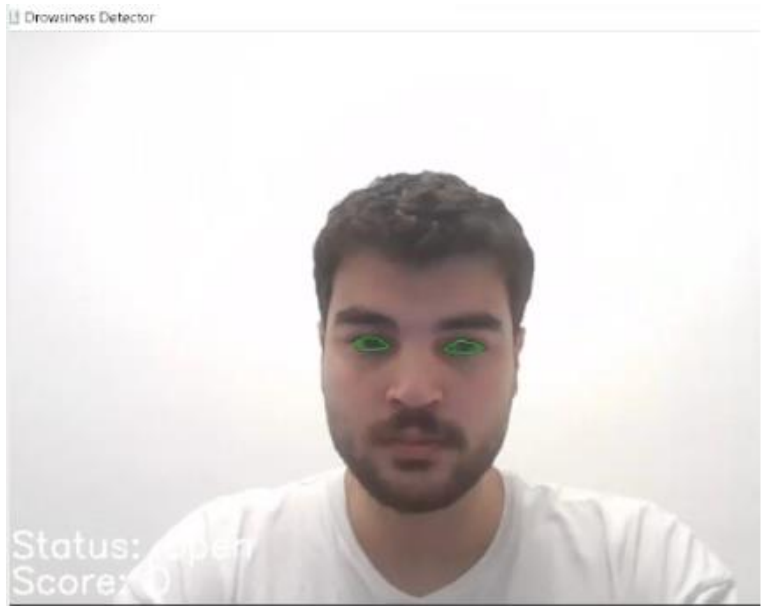

Figure 17 DLIB Non Drowsy

- **Drowsy Detection:** Alarm is triggered when the score exceeds threshold 20

Figure 18 DLIB Drowsy

- **Bad lighting detection:** model falsely predicts eye status using Dlib with bad lighting. It triggers the alarm while I'm not drowsy.

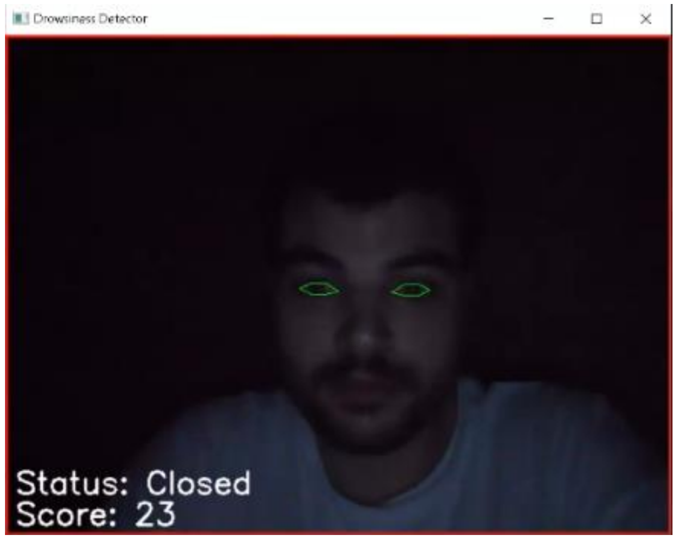

Figure 19 Bad Lighting Dlib

6.2.3 MTCNN System

MTCNN provided robust face and eye detection, even in partially occluded or low-light scenarios. It achieved the most consistent detection across all test cases.

- **Non-Drowsy Detection:** score is 0 and status is open

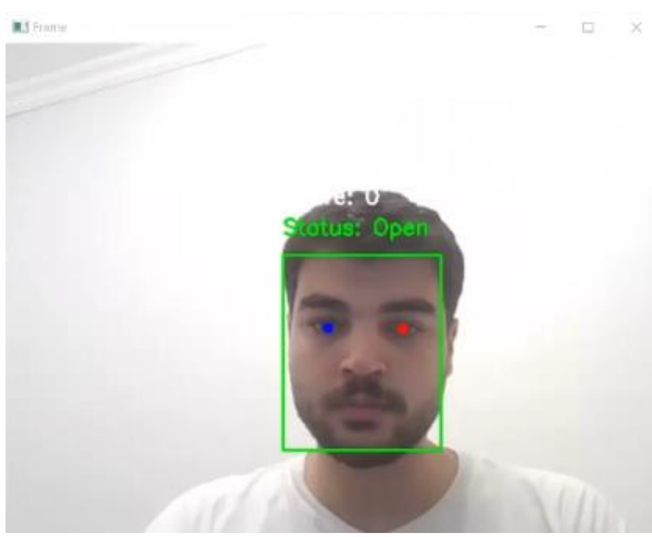

Figure 20 MTCNN Non Drowsy

- **Drowsy Detection:** Alarm is triggered when the score exceeds threshold 20

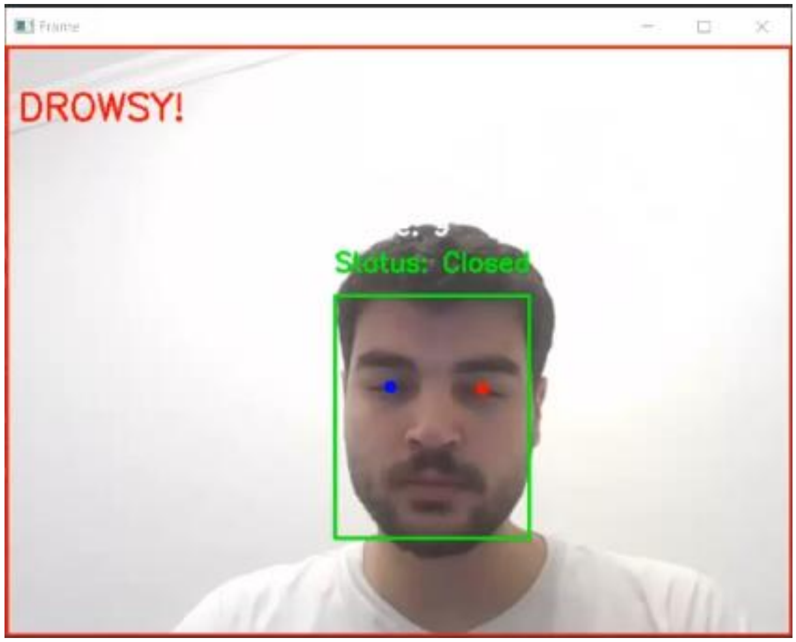

Figure 21MTCNN Drowsy

- **Bad lighting detection:** model correctly predicts eye status due to the high robustness of mtcnn under bad lighting conditions.

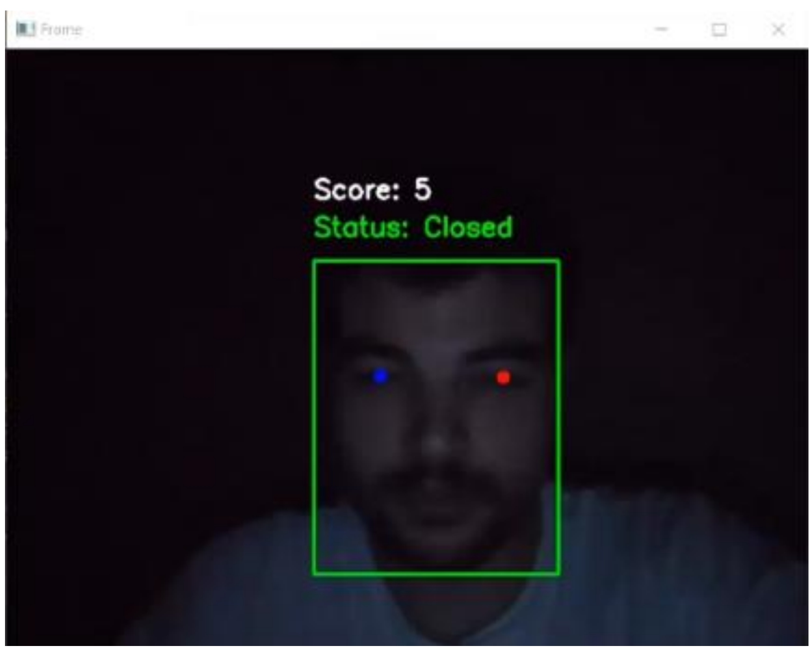

Figure 22 Bad Lighting MTCNN

6.3 Mobile Application Results

The final model was converted to TensorFlow Lite and integrated into a Flutter-based mobile application. The app accesses the device's front camera, captures live frames, classifies the driver's eye state using the lightweight CNN, and triggers an audible alarm when drowsiness is detected.

- **HomePage user Interface:**

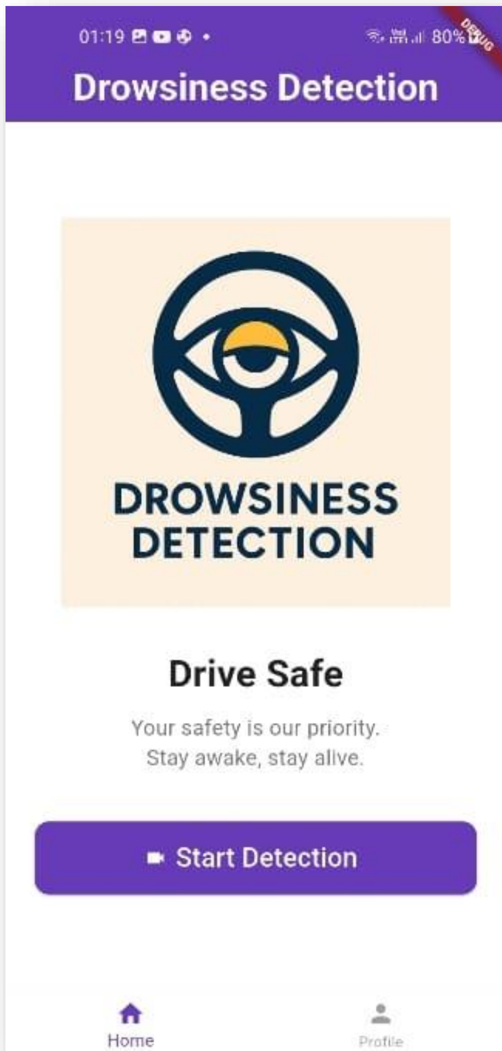

Figure 23 Homepage Screen

- **Live Detection Screen:**

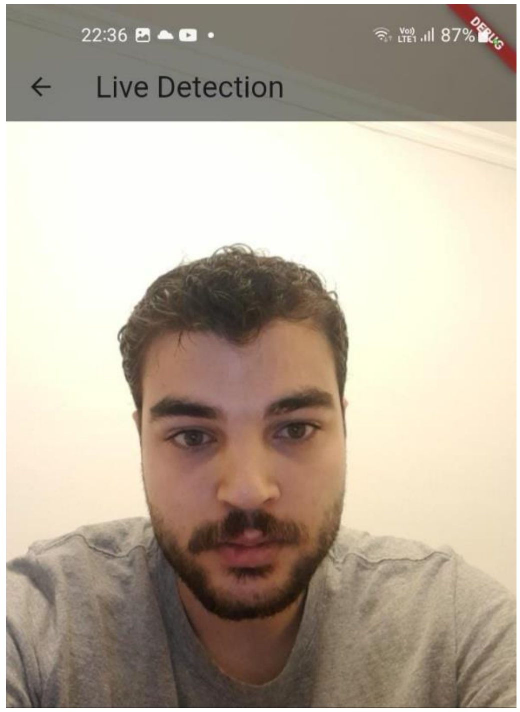

Figure 24 App Live Detection

The mobile app provides real-time feedback with minimal latency and operates entirely offline.

6.4 Comparative Analysis

	FPS	Alarm Responsiveness	Accuracy	CPU Usage
HAAR	5.5	1.756	Low	27%
Dlib	4.7	2.238	High	12%
MTCNN	0.9	9.566	Very High	14%

Figure 25 Desktop Applications Comparison

Table 3 Systems Comparison

Implementation	Detection Method	Eye Detection Accuracy	Alert Latency	Notes
Haar	Haar Cascade	Medium	Low	Fast but less accurate
Dlib	Landmark Points	High	Low	More stable detection
MTCNN	Deep Face Detection	Very High	Medium	Robust in all cases But heavy
Mobile App	MobileNetV2 + TFLite	High	Immediate	Lightweight and deployable

6.5 Summary

The evaluation across all four implementations demonstrates that the system reliably detects drowsiness through eye state classification. While the Haar and Dlib desktop systems showed good real-time performance, MTCNN proved most accurate and robust. The final mobile application achieved a balance between performance and portability, delivering real-time alerts with minimal delay. The scoring mechanism ensured that false positives were minimized, and alerts were triggered only during sustained drowsiness.

Chapter Seven: Future Trends (Department of Information Technology and Media Design, Nippon Institute of Technology, Miyashiro-Machi, Saitama 345-0826, Japan, 2021)

- Advancements in Wearable Devices: As wearable technology continues to evolve, there will be increased use of sensors embedded in items like smartwatches, headbands, or even clothing, to monitor physiological signs like heart rate, skin temperature, and muscle activity. This will help DDD systems capture more accurate real-time data to detect drowsiness.

- 5G and Real-Time Communication: The rollout of 5G networks will transform how DDD systems function by allowing faster, low-latency communication between vehicles, infrastructure, and cloud-based servers. This will enable real-time data sharing and analysis, improving the accuracy and speed of decision-making processes related to drowsiness detection.

- Artificial Intelligence and Machine Learning Evolution: AI and machine learning models will continue to advance, improving the accuracy of detecting subtle signs of drowsiness, such as changes in facial expressions, eye movements, and body posture. More sophisticated models will also be able to differentiate between fatigue, distraction, and other behavioral states to avoid false alarms.

- Vehicle-to-Everything (V2X) Communication: The future will see vehicles becoming part of a larger interconnected system (IoT), where DDD systems will be integrated into a vehicle-to-everything (V2X) network. This network will allow cars to communicate with each other, traffic signals, and infrastructure to improve road safety and provide real-time alerts about drowsy drivers.

- Autonomous Vehicles and DDD Integration: The rise of autonomous vehicles will lead to a new trend where DDD systems not only alert the driver but also take over the vehicle's control when drowsiness is detected. This trend is expected to enhance the safety of autonomous vehicles and provide a more seamless transition from human-driven to fully autonomous vehicles.

- Use of Big Data and Cloud Systems: As the amount of data generated by connected vehicles and mobile sensors continues to grow, DDD systems will increasingly rely on big data analytics and cloud computing. This will enable real-time analysis of large datasets to improve system accuracy, personalized alerts, and predictive capabilities.

- Integration with Public Safety Systems: DDD systems will likely be integrated with broader public safety networks, including emergency services and traffic management systems. In the future, when a drowsy driver is detected, the system could communicate with nearby emergency responders or send alerts to road safety authorities to prevent accidents.

- Global Adoption and Regulation: As the benefits of DDD systems become clearer, we can expect widespread adoption across the automotive industry. Governments may introduce regulations mandating the inclusion of drowsiness detection systems in vehicles, particularly in commercial fleets, to reduce road accidents caused by driver fatigue.

CHAPTER EIGHT: CONCLUSIONS

8.1 Conclusions

In this project, we successfully developed a deep learning solution for Driver Drowsiness Detection (DDD), leveraging state-of-the-art Convolutional Neural Networks (CNNs) to analyze visual cues such as eye and facial movements. This approach enables real-time monitoring of driver alertness, significantly enhancing road safety by providing early warnings of drowsiness and preventing potential accidents.

We also explored previous contributions in the field of DDD, reviewing various systems and methodologies that have been employed over the years. By analyzing existing solutions, we identified key strengths and limitations, allowing us to implement a more efficient and scalable approach for real-time drowsiness detection using mobile devices.

The system architecture we designed incorporates the use of mobile sensors, real-time video processing, and machine learning algorithms, offering an effective framework for detecting drowsiness signs. By utilizing the camera and sensors of a smartphone, our system captures critical visual and behavioral data that is processed to infer the driver's alertness status, thus providing a comprehensive solution for real-time monitoring.

Looking forward, we highlighted future trends and future enhancements in my project, The potential deployment of this system on mobile apps opens up the possibility of making drowsiness detection more accessible to a wider range of users, contributing to safer driving experiences on a global scale.

8.2 My Future Project Enhancements

To further improve the system's effectiveness and expand its usability, several enhancements are proposed for future development:

- **Facial Expression Detection**

  Perform hyperparameter optimization for your Convolutional Neural Network (CNN) model to improve performance. Fine-tune parameters like learning rate, batch size, and network depth to achieve better accuracy and reduce overfitting.

- **Extra Application Features**

  Focus on the deployment of the model in a mobile app, enabling real-time drowsiness detection on smartphones. Ensure the model is optimized for mobile platforms to provide seamless, efficient, and low-latency predictions using edge computing.

- **iOS Deployment Support**

  Integrate yawning detection as a key feature to enhance the accuracy of the drowsiness detection system. By detecting yawns, the system can make more reliable decisions about the driver's alertness, especially when paired with the eye-related metrics.

8.3 References

1. Bajaj, J. S., Kumar, N., Kaushal, R. K., Gururaj, H. L., Flammini, F., & Natarajan, R. 2023. System and Method for Driver Drowsiness Detection Using Behavioral and Sensor-Based Physiological Measures. *Sensors*.

2. Department of Information Technology and Media Design, Nippon Institute of Technology, Miyashiro-Machi, Saitama 345-0826, Japan. 2021. Trends and Future Prospects of the Drowsiness Detection and Estimation Technology.

3. Md. Ebrahim Shaik, Md. Milon Islam, Quazi Sazzad Hossain. 2021. A review on neural network techniques for the prediction of road traffic accident severity.

4. Safarov, F., Akhmedov, F., Abdusalomov, A. B., Nasimov, R., & Cho, Y. I. 2023. Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer-Vision and Eye-Blink Analyses for Enhanced Road Safety.

5. Shaik, Md. Ebrahim. 2023. *A systematic review on detection and prediction of driver drowsiness*.