



Drive Guard: AI-Powered Driver Monitoring System for Drowsiness Detection and Road Safety.

Prepared By

Name: Raghad AbdulAlem Hassan Muammar

ID No.: 243000043

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

Supervisor Name: Yasser Abdelhamid

Collage of Informatics, Midocean University,
Comoros

Acknowledgements

I would like to extend my special thanks to Dr. Yasser Abdelhamid for his exceptional guidance, valuable insights, and unwavering support throughout the course of this project.

His expert advice and continuous encouragement have played a vital role in the successful completion of this thesis.

I am truly grateful for the time, effort, and dedication he has invested in helping me achieve my goals.

Abstract

Driver drowsiness and syncope are the most relevant causes of traffic accidents. To address these safety issues, this project presents “Drive Guard”, an intelligent and AI-based driver monitoring system. With a high-definition camera, “Drive Guard” monitors facial expressions, eye movement and other critical behavioral parameters to identify signs of drowsiness and syncope. When drowsiness is detected, it alerts the driver acoustically, preventing accidents. And even if a fainting spell is detected, it can respond automatically: “Drive Guard” immediately sends an emergency alert to local services so security of the driver can then be guaranteed. “Drive Guard”, a qualitative leap in safeguards for drivers and other highway users, is the most complete solution for reducing accidents on the road due to fatigue and impairment.

List of Publications

Raghad AbdulAlem Hassan Muammar. Dr. Yasser Abdelhamid. “Drive Guard: AI-Powered Driver Monitoring System for Drowsiness Detection and Road Safety”. Manuscript in preparation for submission to *MidOcean Journal for Research and Studies*.

Drive Guard: AI-Powered Driver Monitoring System for Drowsiness	
Detection and Road Safety.....	ii
Acknowledgements	iii
Abstract	iv
List of Publications	v
List of Abbreviations	viii
List of Figures.....	ix
List of Tables	ix
Chapter 1: Introduction	2
1.1 Problem Definition	3
1.2 Thesis Objectives	4
1.3 Thesis Contribution.....	5
1.4 Thesis Organization	5
Chapter 2: Background	7
Chapter 3: Related Work.....	15
3.1 Related Work Summary.....	18
Chapter 4: Proposed Approach	19
4.1 Introduction	19
4.2 Dataset Description	19
4.5 System Workflow.....	21
Chapter 5: Experimental Results and Discussion	24
5.1 Experimental Setup	24
5.3 Results	27
5.4 Real-Time Testing.....	28
5.5 Discussions.....	29
5.6 Summary	30
Chapter 6 Conclusion and Future Work	31
6.1 Conclusion.....	31
6.2 Future Work	32

References.....	33
Appendix A: Source Code for DriveGuard System	35
A.1 Model Architecture	35
A.2 Training Procedure.....	35
A.3 Model Saving	36
A.4 After Training.....	36
A.5 Model Loading	36
A.6 Single Image Prediction	36
A.7	37
A.8 Live Camera Detection	37

List of Abbreviations

- **AI:** Artificial Intelligence
- **API:** Application Programming Interface
- **CNN:** Convolutional Neural Network
- **DDD:** Driver Drowsiness Detection
- **DL:** Deep Learning
- **DMS:** Driver Monitoring System
- **ECG:** Electrocardiogram
- **EEG:** Electroencephalography
- **EMG:** Electromyography
- **FPS:** Frames Per Second
- **GAN:** Generative Adversarial Network
- **GCN:** Graph Convolutional Network
- **GPS:** Global Positioning System
- **IR:** Infrared
- **JPG:** Joint Photographic Experts Group
- **LBP:** Local Binary Pattern
- **LSTM:** Long Short-Term Memory
- **ML:** Machine Learning
- **NHTSA:** National Highway Traffic Safety Administration
- **OPM:** Oculomotor Parameters Monitoring
- **RGB:** Red Green Blue
- **SOS:** Save Our Souls (universal distress signal)
- **SSD:** Single Shot Multibox Detector
- **SVM:** Support Vector Machine

List of Figures

Figure 1 Sample image showing the “open eyes” 1	21
Figure 2 Sample image showing the “open eyes” 2	21
Figure 3 Sample image showing the “open eyes” 3	21
Figure 4 Sample image showing the “close eyes” 4	21
Figure 5 Sample image showing the “close eyes” 5	21
Figure 6 Sample image showing the “close eyes” 6	21
Figure 7 System Workflow Diagram	23
Figure 8 CNN Model Architecture	25
Figure 9 Accuracy & Loss Function	26
Figure 10 Confusion matrix	27
Figure 11 Real-Time Output	28
Figure 12 WhatsApp emergency Message with driver location	29

List of Tables

Table 1 Related Work Summary	18
Table 2 Dataset	19

Chapter 1: Introduction

Artificial Intelligence is the progress of computer systems that are capable of doing things which until now required human intelligence. AI is capable of learning, reasoning, and self-correction that enables machines to perform tasks which typically require human intelligence (Russell, 2020). AI has been applied in many fields like medicine, robotics, financial market forecasting or voice/image recognition or gaming but finding use specially can be found in functions of autonomous vehicles.

AI has become a key part of the transportation sector, improving safety and efficiency. We have been largely dependent on the likes of AI in this particular space with one of the top applications being Driver Monitoring Systems (DMS). Drive AI also has its own set of transporters to use AI algorithms in order to monitor behavior behind those wheels for signs of potential impairment, alerting drivers and avoiding tragedies. (Babu, 2021)

There are nearly automatic feelings of weariness and fatigue, leading to a loss in reaction time and alertness among drivers who drive so far too much for too long. In addition, research has indicated that the cause of fatal-to-the-driver crashes is driver fatigue in 31% of the crashes. (Williamson, 2013) ((NHTSA), 2015)

Common approaches to drowsiness detection focus on vehicle behavior (like lane deviations and sudden breaking) (Desai, 2015). But such approaches cannot provide instantaneous assessments of the driver's state. These advancements in AI have enabled systems to use sensors and cameras to assess physiological expressions such as eye movements, facial expression, and head position to identify early signs of drowsiness (Zhao, 2020). These systems can, on detection, warn the driver via auditory or visual signals, thereby reducing the likelihood of fatigue-related accidents. (Abdar, 2021).

Next to drowsiness, distraction is the other top safety risk for drivers. Drowsiness could be detected in a millisecond when done correctly by an AI, unlike the DMS (Driver Monitoring System) which can only keep track of some basics indicators like the driver's face or position, while they are not driving next to our location (AP) (Sivaraman, 2013). If the system notices

driver distraction of any kind, it can provide feedback to the driver about maintaining attentiveness to the road. (Dasgupta, 2019)

DMS is not without challenges in AI implementation. This is an important concern that needs to be controlled considering the lack of homogeneity of the road Vs driver types in the real world (Awais, 2017). Also, privacy concerns regarding driving habits need to be tackled before it can be widely accepted. Nevertheless, we must be careful that the technology is not abused and that any personal and confidential information gathered is safely stored.

So, the AI in driver monitoring is an important technological door closer to be opened for assuring the safety of automotive. With the corresponding results of real-time analyzes of physiological changes in drivers (such as drowsiness and distraction) behind the wheel, this AI-powered DMS

1.1 Problem Definition

The risk of traffic accidents due to driver impairment, including drowsiness and fainting, remains a major problem in contemporary transportation due to inadequate road safety. Many research papers have proven the link between decreased alertness of drivers and road accidents, and stressed the importance of preventing such situations before they result in a disaster.

The Drive Guard project is an AI-empowered stage that illuminates these problems. Whereas conventional systems are capable of only warning about drowsiness, “Drive Guard” is also able to recognize initial signs of fatigue as well as even fainting.

The platform uses access to high-resolution cameras and advanced machine learning algorithms that are trained to detect subtle differences in the expressions on a person's face, their eye movement and other behavioral clues in order to give people real-time feedback and alerts. By curtailing fatigue-related accidents, further enhancing road safety and saving lives in the process, this method can truly make a difference.

Drive Guard integrates real-time AI-based monitoring systems with an automated emergency response system providing a novel approach to reduce accident risks caused by driver impairment through a proactive approach, preventing crashes and accelerating emergency response when necessary.

1.2 Thesis Objectives

This research seeks to create a driver surveillance system that enhances road safety through the identification and mitigation of drowsy and medical emergencies. The objectives are as follows:

Real-time AI model through which we can identify facial expressions, eye movements, and body language to detect changes in the driver like drowsiness and unconsciousness.

Distinguishing drowsiness from fainting by recognizing distinctive behavioural and physiological traits of each state.

We propose the implementation of a real-time alarm that delivers timely sound alerts to the driver when he or she is drowsy, thus preventing total loss of what the driver is really doing on the highway.

Developing an automated emergency response system that:

- Sends an SOS distress signal to emergency services.
- Transmits the driver's GPS location for more rapid rescue response.

System robustness and precision across different environments (day vs night, varying illuminations, different driving speeds).

Gaining insights for advancement in AI-based driver safety technologies by addressing a few of the current research gaps.

1.3 Thesis Contribution

This thesis presents several important contributions to the study of driver drowsiness detection with deep learning methods:

Eye State Detection Using CNN

A real-time eye status detection system using Convolutional Neural Network (CNN), which is an effective behavioral cue for drowsiness.

The use of dataset

The combination of training datasets are the real human face-eyes from subjects from a large spectrum of ages, races, and view angles, as well as synthetic images generated with state-of-the-art AI algorithms. It is for model robustness and generalization for different driver profiles to make the dataset more diverse and the density of the dataset higher.

Optimized CNN architecture

A customized CNN model for eye state classification in different illumination and environmental challenges, achieving a good level of precision and stability.

Real-Time Alert and Decision System

Based on the classification of the eye state, the proposed system sends timely alerts to identify early signs of drowsiness.

Simple and Practical Implementation

With the use of camera input and CNN-based analysis, it is possible to build scalable and feasible systems for adoption in cars without expensive hardware.

1.4 Thesis Organization

The thesis includes six chapters in which the main topics related to the development of an intelligent system for driver drowsiness and fainting detection based on computer vision and deep learning methods, has been addressed. It is aimed at enveloping the research in an understandable and rational form, from problem definition to assessment of solution.

The first chapter is a general summary of the issue of sleepiness and syncope while driving and its relevance for traffic safety. It also describes the research objectives and main contributions and presents the organization of the thesis.

In the second chapter provides the theoretical background on driver monitoring systems technologies with a focus on the importance of monitoring fatigue and drowsiness. It reviews the conventional methods and sensors applied and justifies why vision-based solutions are necessary.

In the third chapter, we present a literature review on studies on drowsiness detection. It reviews behavioral and physiological signal-based techniques, focusing on machine learning and computer vision applications, and highlights specific research areas for future work that this research may fill.

The fourth chapter describes the proposed CNN-based method. It presents data acquisition, image preprocessing, model architecture design, and a classification mechanism to recognize the driver's states.

The fifth chapter presents the results of the training and testing of the CNN model and evaluates its performance by comparing relevant metrics, followed by a discussion of the strengths and limitations of the study and a comparison with the existing literature.

The final chapter offers a summary of the main observations and contributions of the study for driver safety systems. It also summarizes possible directions in the future to improve the robustness and practicability of the model for different environments and situations.

Chapter 2: Background

Accidents caused by drowsy drivers or those who suffer from cardiac disease are some of the leading causes of injuries, and death in all over the world. Drowsiness slows reaction time and therefore, a motorist's ability to make good decisions, leading to a drastically higher chance of an accident. In extreme cases, this can result in loss of control of the vehicle, and therefore greatly increases the risk of fatal crashes. Current driver monitoring systems are mainly focused on detecting whether a driver has dozed off yet are unable to accurately recognize a fainting driver or manage state-level emergency response, putting driver and passenger safety at significant risk.

To tackle the dangers which lie from driver drowsiness and fainting, this project proposes "Drive Guard", an artificial intelligence-based driver monitoring system which aims to improve road safety by analyzing the driver's facial expressions and behaviors in real-time. When compared to the conventional "fatigue detection" systems that rely mainly on eye closure to determine drowsiness, "Drive Guard" employs cutting-edge computer vision and machine learning algorithms to analyze various drowsiness indicators, such as eye movement, facial expression, head tilt, and body posture.

It works in two main modes:

Drowsiness detection: The system sounds immediate auditory alarms if it detects drowsiness — such as slow eye movements, a long blinking time, or nodding to help refocus the driver.

Fainting detection & Emergency response: If the system detects that the eyes are closed for a long time, no facial expressions are detected, and there is no sign of moving, the system identifies it as a possible fainting event. In such cases, "Drive Guard":

- Sends an emergency notification to the medical services automatically
- Supports live GPS of the driver for quick response

"Drive Guard" integrates real-time AI-based monitoring systems with an automated emergency response system providing a novel approach to reduce

accident risks caused by driver impairment through a proactive approach, preventing crashes and accelerating emergency response when necessary.

The following key questions are addressed in this research:

- What are the most reliable physiological and behavioral correlates for drowsiness detection and fainting in drivers?
- What kind of performance does AI-based driver-monitoring systems provides compared to traditional methods of detecting drowsiness?
- How accurately can we differentiate between states of fainting/drowsiness using deep learning techniques?
- How effective is the integration of real-time alert mechanisms and emergency response features in driver safety systems?
- What are the main technical barriers to deploying AI-powered driver monitoring systems across a multitude of driving scenarios?

Driver drowsiness detection has attracted much attention in recent years and has become an important issue because of the serious safety problems arising from fatigued and drug-intoxicated drivers. At first, systems simply used relatively basic techniques to monitor what the driver was doing and how their physiological signs were behaving. Early methods tried to handcraft vehicle features, including steering and lane shifts and speed variations, to estimate fatigue. These behavioral cues might signal reduced attention or inhibition, and thus drowsiness. (Patel, 2011)

Similarly, physiological sensing techniques began to emerge whereby the state of a driver is directly assessed by collecting measures of cardiovascular activity, skin conductance, and electrical activity generated by the brain using EEG devices. While these metrics provided direct indications of fatigue, their applications were constrained by sensor intrusions and challenges in obtaining clean signals in on-the-move scenario. (Wang, 2020)

Simple camera systems also began to be used, which, by means of image processing with simple thresholds, could detect eye closure, yawning, or nodding the head. Early vision-based systems employed ad-hoc features (e.g., eye aspect ratio or blinking duration) to detect the presence of fatigue.

Yet they were generally sensitive to environmental conditions such as lighting, face pose, and occlusions, which made them not very robust. (Abtahi, 2014)

Computer vision and machine learning have improved systems' ability to automate complex visual feedback. The level of detail in faces was also well captured by high-resolution video recording cameras, and machine learning-

based classifiers (such as support vector machines and decision trees) trained on manually derived features to discriminate between alertness states. Although these methods could have better detection performance than just using simple thresholding, they still require for the features to be designed manually and do not scale well over different scenarios. (Picot, 2012).

The inflection points really came with deep learning, specifically Convolutional Neural Networks (CNNs), which were capable of automatically learning hierarchical features directly from raw images. Convolutional Neural network (CNNs) is a model inspired by the human visual system and have 2 or more convolutional layers; we can add fully connected layer to achieve classification.

(e.g. edges, textures, and complex facial structures). This facilitates to detect eye states (open, close, blink), facial landmarks and micro-expression for fatigue automatically without any hand-crafted features by CNNs. In addition, contemporary CNNs such as ResNet and MobileNet have been designed to provide real-time computation on embedded systems, which is suitable for in-vehicle installation. (Zhang Z. Z., 2021)

Acknowledging that driver fatigue occurs gradually, temporal models such as the Long Short-Term Memory (LSTM) networks were used. LSTMs are a type of RNN capable of learning long-term dependencies in sequential data, which makes them appealing for processing video sequence or physiological signals, in which temporal patterns such as long-lasting eye closure or repetitive nodding indicate drowsiness. Such hybrid architectures that use CNNs for extracting spatial features and LSTMs for temporal dynamics are effective in developing early and timely fatigue detection. (Jiao, 2023)

In order to resolve data paucity issues, GANs have recently been introduced. GANs consist of two neural networks, a generator and a discriminator, which play a zero-sum game to produce realistic images of drivers or physiological signals. It is important to add synthetic examples of rare events like fainting or drowsiness that would be rare in programmatic data so that the model learns them conservatively leading to generalization and temporal robustness. (Zhang Y. W., 2022)

The Graph Convolutional Network (GCN) uses graphs instead of grid data to generalize convolutions.

In the field of driver monitoring, facial landmarks can be represented as nodes in a graph and spatial and functional ties as edges. The learned features in

GCNs take into account such relational patterns, which lead to more accurate detection of subtle facial movements in head pose and occlusion variations. (Wang, 2020)

Contemporary driver monitoring systems commonly combine several sources of data like video, physiological signals and contextual information (e.g. environmental conditions, time of day and so on). Multimodal fusion methods fuse complementary inputs for increasing reliability and reducing false positives. For instance, fusing facial video analysis with heart rate variability affords a more complete view of driver state than does either modality in isolation. (Patel, 2011)

In addition, context-aware systems adapt sensitivity according to environment and driver (e.g., the alertness detection sensitivity is increased when driving in darkness or after longer driving times). Improvements in the user interface are made for reducing driver distraction due to false alarm and enhancing the acceptance by the user. (Picot, 2012)

Notwithstanding such remarkable progress, significant challenges for robustly deploying good drowsiness detection systems to drivers of different profiles are still posed within the context of the on-road environment. Differences in the facial characteristics of individuals, social practices, lighting, or the positions of the sensors also make things difficult for model generalization. However, there are also practical obstacles such as limited computational resources, concerns about data privacy and the need for real-time systems. (Wang, 2020)

Current research is also working on the interpretability of deep learning models to make it easier to understand for consumers and regulators. There are also efforts to adapt the system to modify detection thresholds based on individual drive profiles and driver habits. Vehicle control systems for preemptive safety interventions like adaptive cruise control or automatic braking, is a promising future direction. (Zhang Z. Z., 2021)

In general, the path of driver drowsiness detection has changed from resonance in simply behavior watch to intelligence system equipped with sophisticated Ai algorithm and multimodal data fusion technology. Such technologies could potentially save the lives from fatigue-related accidents on a large scale and could make the world's roads safer.

In recent years, the demand for road safety and advances in intelligent transportation systems have led to numerous developments in driver monitoring technologies. At first, most such efforts focused on explicit external driving behaviors like lane-keeping patterns or steering corrections, which could easily be collected from existing vehicle data but did not always capture more subtle internal driver states (e.g., fatigue or cognitive distraction). This void led to the move to in-cabin monitoring using cameras and biometric sensors, as this gives more immediate insights into a driver's physiological, or mental state.

These advances in hardware, including IR based night vision cameras and multi-function EEG headbands have significantly improved the sensors to monitor the vehicle. This was one of the main limitations of prior systems, whereas these modern technologies allow for the collection of data even in low-light or high-noise environments. In addition, the existence of more extensive annotations has enabled the training of better models for understanding both on-road rider and environmental behaviors across different groups of people, vehicles, and situations.

Similarly, incorporating multimodal data has also been cited as a potent strategy to enhance system resilience. For instance, we consider combinations of facial landmarks and vital signals such as heart rate variability (HRV) or respiration rate to improve the sensitivity for subtle fatigue-related changes that can potentially not be detected with facial data only. By fusing the signals from all these modalities, the system can dynamically adapt to different use cases where one of them is more reliable due to lighting conditions, head orientation or some driver specific properties. It significantly advances user acceptance by increasing the ability to adapt and decreasing occurs false positives.

Another active area of research is about the personalization in driver monitoring systems. Because individual responses to muscle fatigue are highly dynamic, conventionally measured static thresholds commonly do not capture intersubject variability to solve, adaptive systems are in development and will learn each individual's baseline over time and adjust the sensitivity appropriately. A driver who, say, naturally blinks more frequently may not be flagged under the same criteria as one who rarely blinks, unless an over-above-if-you-are-blinking-a-lot blink rate is surpassed. That individualized calibration makes it more accurate, and a system you may be able to trust.

At the same time, research is being done on processing AI directly in-vehicle with the help of edge computing and low-power embedded devices that allow for real-time inference. Edge computing allows instantaneous response times and improved privacy over cloud-based systems, where biometric data is transmitted to the cloud for analysis. Optimized deep learning models (for example, MobileNet, SqueezeNet and Tiny-YOLO) have introduced opportunities that make these embedded solutions more practical than ever before. Certainly, there has been a lot of attention paid to the driver alert and some attempts either proved timely, or pass away (unobtrusive but preparative) as they say.

Researchers also note that a warning could cause a false positive, which can lead to alert fatigue from drivers who start disregarding them. As such, an increasing number of intelligent alert systems are being rationalized that adjust the intensity and frequency of feedback according to the driver's reaction. A gentle tone might be used when the first signs of fatigue appear, then a blaring alarm or even vibrating seat padding for more dire circumstances.

In addition, regulatory requirements and ethical issues play an important role. With more of these systems being integrated into commercial vehicles and public transportation, some government and safety agencies are starting to mandate some minimum standards for driver monitoring. It is also a considerable matter of worry with where and how the data is being taken, saved and utilized. What is a need of the hour is that more secure data anonymization protocols must be put into place and clearer information should be given to users about what data is being collected, in order for the public to have trust in driver monitoring systems.

One of the main challenges in the design of good drowsiness detection system is to reach a balance between sensitivity and specificity. While a system with high sensitivity may catch every fatigue incident, it is also more likely to issue too many false positives which can bother the driver or make him/her deactivate the system. Conversely, a highly targeted system may not provide the false positives but do just as poorly in detecting early signs of drowsiness. This is where it gets tricky as finding the right trade-off will need extensive testing and labelled datasets which are realistic with proper handling of driving scenarios.

However, decoding a driver is not always simple to do. Mash-up of Cultural norms, age, gender and driving habits A younger driver may have more pronounced facial movements over an older driver or eye contact and posture might vary between cultures. Consequently, cross-cultural studies along with the data are important to guarantee that it is not a biased training sets and also ensure that models constructed therein will work fairly similarly for all people.

Furthermore, simulation-based training and testing platforms are gradually becoming an integral part of the process in the development and deployment of such systems.

Virtual driving simulators enable researchers to recreate dangerous situations in a controlled environment, where fatigue can be induced and monitored. This way, it not only improves the data collection but also keeps adding different algorithms at a place for continuous ping test in parallel conditions. Furthermore,

such simulators are increasingly used in driver education programs to raise awareness about the dangers of drowsy driving and to train users on how to respond to fatigue warnings effectively.

Drivers are also another element drawing concern with the use of emotional state analysis in driver monitoring. Stress, anger or anxiety can have a plague-like effect on our driving performance and how these emotions mix with fatigue is not only complex but dangerous too. Today, AI systems are being trained to recognize facial expressions for emotional responses or vocal intonation for understanding of the mood even physiology changes. Incorporating such a dimension within the drowsiness detection framework would provide additional context The understanding that might enable us to more reliably anticipate these high-risk states sooner.

Moreover, unsupervised and semi-supervised learning have also achieved much improvement in using unlabeled data reducing the cost of labeled data which always is labor-intensive and expensive to acquire.

These techniques enable the model to discover trends in these unlabeled driving behaviors and get better as they see more real-world information over time. This learning paradigm can be particularly relevant in personalized systems, where the model has to adapt itself to a specific driver yet it is infeasible that too lengthy calibration phases can be foreseen.

And as the development and use of autonomous and semi-autonomous vehicles increases in this decade, so too will the need for accurate driver state monitoring. Driver must ALWAYS be ready to take over (level 2 and level 3 automation) Thus, systems such as Drive Guard may act as this essential sentinel to make sure the human operator is always able to take control again should danger rear its ugly head. These will be vital to ensure that system failures and misjudgments do not creep in as we shift from human control to machine.

In sum, the increasing research and development in AI driver monitoring are attuned to a worldwide maturity about humans being in charge of road safety. The companies behind these systems are seeds in this wide growth of not only technical and technological solutions, but communities with smarter, more safe and responsible transportation ecosystems Significantly, the application of cutting-edge algorithms combined with real-time sensing and user-centered design is increasing the likelihood of “driver drowsiness detection systems” becoming a standard safety feature in cars around the globe.

The last decade has seen a revolution in computer vision with the advent of deep learning, especially Convolutional Neural Networks (CNNs), which enables systems to automatically extract spatial features from an image with minimal human effort. CNNs are especially handy for this problem because trying to classify driver drowsiness in video (e.g. the corners of eye, mouth, eyebrows that change to varying degrees under fatigue) is a difficult and required some nice features to do so. These networks are trained with large datasets to differentiate between the alert and drowsy facial expressions, which may be too subtle for a human observer.

In contrast to standard machine learning methods which need careful feature engineering, CNNs learn hierarchical features which abstract the visual information (lines and finally face structure). Due to this well-defined layering scheme, the proposed method provide solid detection capabilities even in presence of issues like partial occlusions (such as sunglasses) or different lighting and poses. More recent CNN architectures like ResNet or Inception and EfficientNet have achieved even higher accuracies with lower computational burdens — a sweet spot for deploying in real-time, embedded automotive systems.

Chapter 3: Related Work

Fatigued and unconscious drivers are considered to be a major contributor to fatal road accidents globally. A large number of researchers have proposed various countermeasures to identify and to prevent such incidents in the last decade based on different methodologies. This chapter reviews the past work and technologies for detecting driver fatigue and impairment.

It is points out that "driver drowsiness" is among the major causes of road casualties and is involved in almost 20% of all accidents, causing more than 1,200 deaths every year. It emphasizes the need of sound driver drowsiness detection systems to improve the safety of vehicles. It also qualifies detection methods as effectiveness vehicle-based, behavioral-based and physiological-based techniques that all measurable and different indexes of drowsiness. Vehicle-based measures are designed to monitor driving actions, and behavioral measures to measure physical signs such as yawning, blinking, and eye movements through camera. Physiological measures detect signals under ECG and EEG to indicate drowsiness through variations in heartbeats and periodic patterns of brain activity. This review also highlights several technological developments such as real-time monitoring platforms, smartphone applications and classification algorithms to assess driver alertness, etc. This document, therefore, serves as a compendium to the existing knowledge on fatigue detection methods, and recommendations for improvement. Overall, it serves as a comprehensive survey of the challenges and solutions surrounding driver drowsiness detection technologies. (Saini, 2014)

Presents a research study to develop a real-time driver drowsiness detection system using deep learning techniques such as Convolutional Neural Networks (CNN). The main motivation is to mitigate the serious problem of driver fatigue which is a major cause of motor vehicle accidents and death around the world. Proposed System: The proposed system with the Mobile Net architecture and Single Shot Multibox Detector is used for detecting if the eyes are opened or closed of a driver on the real-time video stream. To train the model, they created a dataset of about 4,500 images of drivers in different states of alertness. This approach to authenticity is similar to one they describe in which the balance of accuracy, computational efficiency, and

affordability allows such a system to be implemented with cheap hardware, such as a Raspberry Pi or generic cameras. This approach allows for a wider reach of drowsiness detection technology, specifically in cars that might not have sophisticated safety systems. It provides a high-level look at the methods used in the research, from data collection to algorithm design, and evaluation metrics, intending to contribute to the ongoing efforts in reducing. (Dipu, 2021)

Concentrates on the construction of a deep CNN–LSTM network for the identification of driver fatigue through EEG signals. The EEG data used by the model come from driving simulation with 11 participants, filtering and standardization of data were performed. You train the model for classification of tabular data through a combination of CNN and LSTM layers in a neural network. Experimental results show that the algorithm can accurately detect driver fatigue and has better robustness against the environmental noise. This method has potential application in driver fatigue detection systems. (Sheykhivand, 2022)

Propose a new solution to automatically detect driver fatigue via EEG signals that are based on deep learning methods, which use the Fusion of GAN and GCN. The study overcomes some drawbacks in previous literature that had largely only considered binary classifications for classifying fatigue and had a heavy reliance on self-reported data. (Ardabili, 2024)

Describes in detail real time Driver Face Monitoring System to identify the driver fatigue and distraction based of the machine vision techniques used. They propose a novel method for hypo vigilance detection based on features extending beyond facial region itself and eye region symptoms, including eye closure percentages and head rotation. Novel contributions include a simple head yaw rotation detection method with no reliance on explicit eye detection, model-agnostic symptom extraction that is adaptive to the current frame, and personalized symptom normalization through a brief training phase.

Fuzzy Expert System is used to determine driver's conditions based on the symptoms. Importantly, it reflects serious challenges in quantifying fatigue and concentration, crucial in improving public safety and minimizing road accidents associated with driving under the influence of sleepiness and distraction. The system reduces accident rates significantly by analyzing facial and eye symptoms and informing drivers of their hypo vigilance states.

The paper offers a substantial review of prior work on driver monitoring systems (DMS), and presents a delineation of methodology, as well as experimental results, that

suggest strongly positive indications for enhancing driver safety. (Sigari, 2013)

Focused on driver distraction and drowsiness based on data collected by two driver monitoring systems (DMS). The research investigates retrospective data of 330 drivers, in which reveals result of detailed analysis of driver profiles (short-distance vs long-distance drivers) and trip characteristics (driving time, speed, break frequency) shows on inattention. The study uses probit models to evaluate the risk factors of the alerts for distraction and drowsiness emitted by the systems. (Lobo, 2020)

3.1 Related Work Summary

Table 1 Related Work Summary

References	Datatype:	Training Model	Accuracy	Dataset
1	Image data (facial features, eye state) Physiological signals (EEG, ECG, EOG)	-Haar Cascade Classifiers -Fuzzy Wavelet-Packet-Based Feature -Extraction Hybrid approaches combining image processing and bio-signals	80% To 95%	Custom datasets created from simulation driving tests and real-time monitoring system
2	RGB images in JPEG or PNG format, annotated with bounding boxes for object detection.	TensorFlow Object Detection API, specifically the MobileNet_SSD_v1	over 90%	around 4500 training images and 700 testing images
3	EEG signals	A deep CNN-LSTM network	99.23%	A comprehensive dataset of EEG signals
4	-EEG signals (Electroencephalography) -ECG signals (Electrocardiography) -EMG signals (Electromyography) -Respiratory effort signals	-Generative Adversarial Networks (GAN) -Graph Convolutional Networks (GCN)	91.7% To 99.4%	20 participants in five different classes
5	Image data from video sequences of drivers' faces.	Fuzzy expert system combined with horizontal projection analysis for eye and face features.	90.8%	27 video sequences
6	2 Driver Monitoring Systems (DMS) -an embedded system installed in commercial vehicles and a -smartphone application available to the public.	Ordered probit models (OPM)	93%	1,658 sub-journeys and 11,034 distraction events and 6,949 drowsiness events

Chapter 4: Proposed Approach

4.1 Introduction

This chapter describes the methodology employed in the Drive Guard system design. It discusses dataset selection, system workflow, and real-time system design, which provides an alert and action system that will help to develop a robust and intelligent driver monitoring system that will be capable of accurately detecting drowsiness and fainting and responding immediately in order to avoid accidents.

4.2 Dataset Description

One of the publicly accessible datasets in Dataset table that has been chosen is the "Closed Eyes In The Wild (CEW)" it was selected, as it best met the requirements of this project, which include variation in facial expressions, ages, and lights. This selected dataset was merged with an AI-generated images dataset. The consolidated dataset was used for training and testing of the Drive Guard system.

Table 2 Dataset

Dataset Name	Link	Biomarker	Size	Limitations
Driver Drowsiness Dataset	https://www.kaggle.com/datasets/ismailnasri20/driver-drowsiness-dataset-ddd/data	Facial landmarks, eye movement, blink rate	2.76 GB 5000+ Records	limited to certain drowsiness conditions.
Drowsy Detection Dataset	https://www.kaggle.com/datasets/yasharjebraeily/drowsy-detection-dataset	Eye, mouth features, videos.	13.43 MB 8000+ records	focus on specific driver scenarios.
Drivers-drowsy	https://www.kaggle.com/datasets/khanh14ph/driver-drowsy	Eye movement, facial landmarks, video data.	3.02 GB 10,000 records	focused on specific age group, video processing required.

Drowsiness dataset	https://www.kaggle.com/datasets/dheerajperumandla/drowsiness-dataset	Eye blink, heart rate, facial landmark	175.84 MB 5000+ Records	Imbalanced data, some missing entries.
Frame Level Driver Drowsiness Detection (FL3D)	https://www.kaggle.com/datasets/matjazmuc/frame-level-driver-drowsiness-detection-fl3d	facial features, frame level video data	645.01 MB 5000+ Records	Requires high quality video data.
Drowsiness Detection Dataset	https://www.kaggle.com/datasets/prasadvpatil/mrl-dataset	Video, facial cues, physiological signals	22.85 MB 5000+ Records	Limited data on environmental factors.
Closed Eyes In The Wild (CEW)	https://parnec.nuaa.edu.cn/_upload/tpl/02/db/731/template731/pages/xtan/ClosedEyeDatabases.html	facial images, Closed Eyes, Open Eyes	20MB 2428 Image	Limited data size, Lack of diversity in lighting.

Sample Images from the Dataset (Awake)

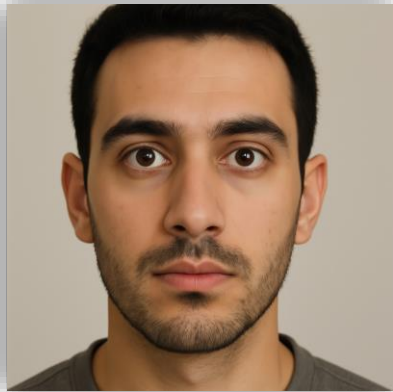


Figure 3 Sample image showing the "open eyes" 3

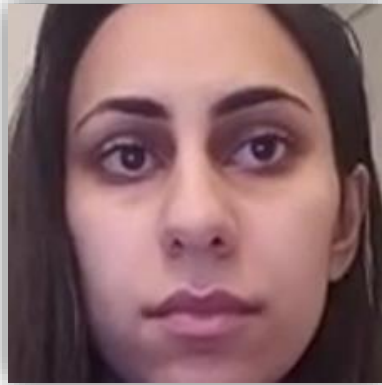


Figure 2 Sample image showing the "open eyes" 2



Figure 1 Sample image showing the "open eyes" 1

Sample Images from the Dataset (Drowsy)



Figure 6 Sample image showing the "close eyes" 6



Figure 5 Sample image showing the "close eyes" 5



Figure 4 Sample image showing the "close eyes" 4

4.5 System Workflow

- Camera Input → Captures live video frames of the face via webcam.
- Face and Eye Detection: Uses MediaPipe FaceMesh to extract facial landmarks and isolate eye regions from each frame.
- Preprocessing: The images are converted to grayscale, equalized, resized to 150×150 pixels, and normalized.

- Eye State Prediction (CNN Model): The preprocessed eye images are fed into a trained CNN to classify the eye state as Awake or Drowsy.

Drowsiness Monitoring Logic:

If Drowsy state persists for more than 2 seconds, an audio alarm is triggered. If drowsiness continues beyond 5 seconds, the system classifies the state as fainted.

Emergency Actions:

Upon fainting detection, the system automatically sends an emergency WhatsApp message including the driver's GPS location using geocoder and pywhatkit.

An emergency alarm sound plays, and the monitoring system terminates.

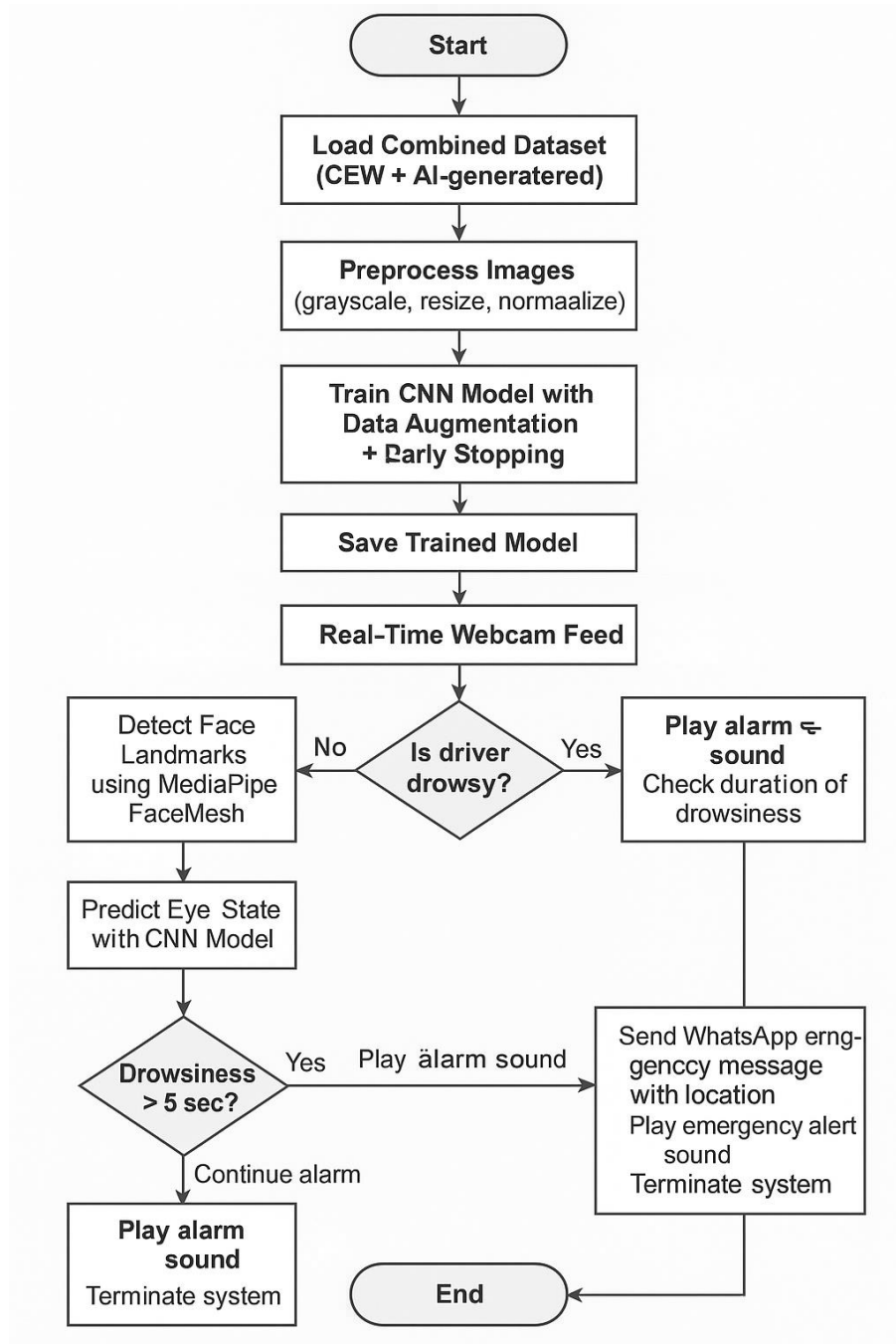


Figure 7 System Workflow Diagram

Chapter 5: Experimental Results and Discussion

5.1 Experimental Setup

In order to execute this project, -Jupyter Notebook- was used, and it was working on Windows OS. The Python libraries used were:

TensorFlow: to construct and train the model of CNN.

Scikit-learn: for splitting of data and evaluation.

OpenCV: for loading and preprocessing of the images.

Matplotlib: for visualization.

TQDM: for monitoring the task progress while training 2 From a generalizability context, our proposed architecture has several advantages compared to the state of the art.

Dataset:

The dataset is composed of “Awake/Drowsy”-labeled images.

Image Source Path:

“C:\Users\X1 YOGA\Downloads\All_data\Train_Data”

Image Size: 150 ppi (pixels per inch)

Classes:

Awake → Label 0

Drowsy → Label 1

Preprocessing Steps:

The sized of all the images were re-sized in to 150x150 pixel.

pixel values between [0, 1] were normalized

All class labels were expanded into one-hot vectors with `to_categorical()`.

Train/Test Split:

For training and testing, 80% and 20% of the data set was used, respectively.

Data was randomized before division.

A fixed `random_state = 42` was applied for reproducibility.

Model Architecture (Convolutional Neural Network)

Conv2D (32 filters, 3x3 kernel, ReLU) followed by MaxPooling2D
Conv2D (64 filters, ReLU) + MaxPooling2D
Flatten
Dense (128 units, ReLU)
Dropout (0.5)
Dense (2 units, Softmax activation).

Training Configuration:

Loss Function: Categorical Crossentropy.

Optimizer: Adam.

Metrics: Accuracy.

Epochs: 50.

Batch Size: 32.

Callback: EarlyStopping was used to monitor training loss and prevent overfitting.

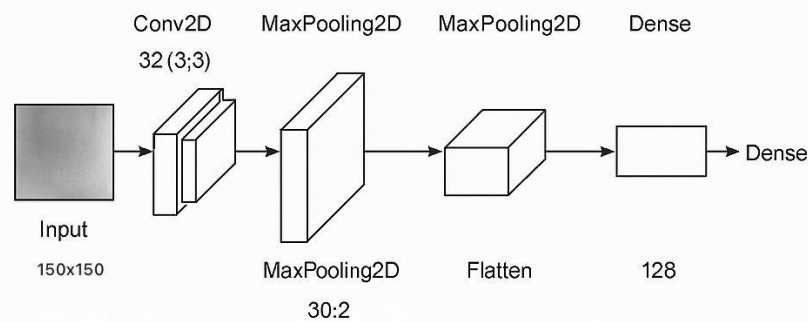


Figure 8 CNN Model Architecture

5.2 Evaluation Metrics

The model was evaluated using typical classification measures to evaluate its performance on the test set. The following metrics were used:

Accuracy: Overall valid accuracy of the model predictions

Precision: The proportion of true positive predictions in the total of predicted positives

Recall (Sensitivity): The ratio of true positive predictions to the total actual positives

F1-Score: A balance of precision and recall, being their harmonic mean

Confusion Matrix: a matrix that shows the counts of true positives, true negatives, false positives, and false negatives.

Evaluation: was performed using `classification_report()` and `confusion_matrix()` functions.

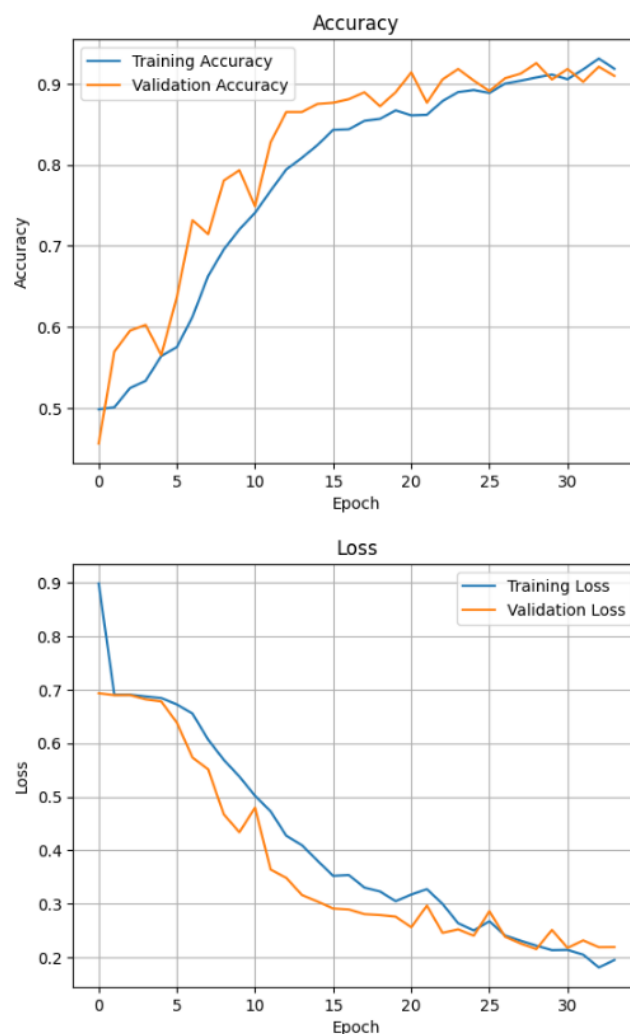


Figure 9 Accuracy & Loss Function

5.3 Results

The model was trained for 34 from 50 epochs. It was evaluated on the 20% unseen test set after training. The following results were observed:

The confusion matrix better delineated “Awake” and “Drowsy” mention states.

Class-wise Precision, Recall, F1-Score, and Support.

A high accuracy 93% of the model was obtained, which indicated good generalization.

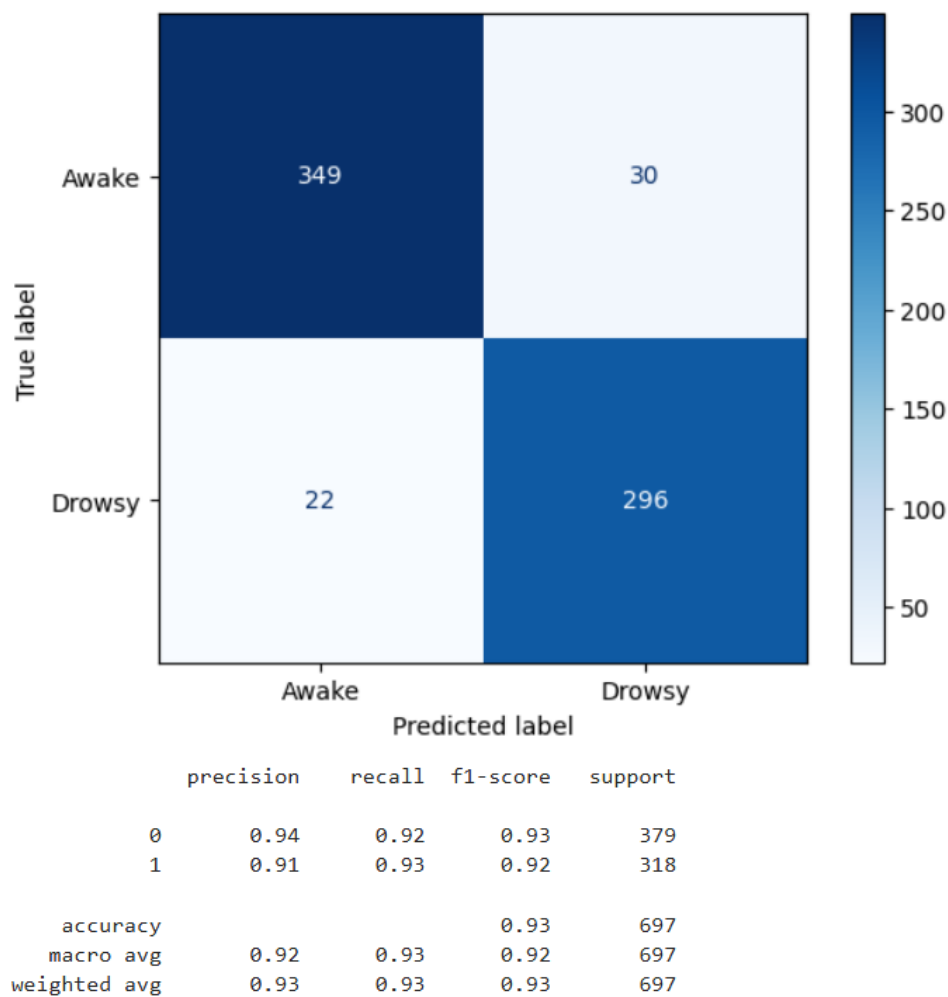


Figure 10 Confusion matrix

5.4 Real-Time Testing

In the final step, we implemented the trained CNN model into a real-time drowsiness detection system by using a webcam.

Key Components:

Eye detection was performed using the Mediapipe face mesh solution

Eyes were randomly cut from the frames of images in real-time and resized into 150×150 pixels.

The cropped eyes were fed through to the CNN model to make predictions on the eye state (Awake or Drowsy).

The model was characterized by the presence of:

Audio warning of drowsiness for over 2 seconds

Status tracking logic: If the user remained awake for more than 7 seconds after a sleep attempt (i.e., after a drowsy), the system categorized the user as “Fainted”.

Emergency response: If fainting was detected, the system fetched the current geolocation using geocoder and sent it via WhatsApp using pywhatkit, this demonstrated the practical utility of the model beyond offline testing and into real-world safety scenarios

```
Current state: Drowsy
Current state: Awake
Current state: Drowsy
Current state: Awake
Current state: Drowsy
Current state: Awake
Current state: Drowsy
Current state: Awake
Current state: Drowsy
Current state: Awake
Current state: Drowsy
Current state: Awake
Current state: Drowsy
Current state: Awake
Current state: Drowsy
Current state: Awake
Current state: Drowsy
FAINTED detected! Sending location and contacting emergency!
Current state: FAINTED
Location sent ✔
Emergency contacted ✔
```

Figure 11 Real-Time Output

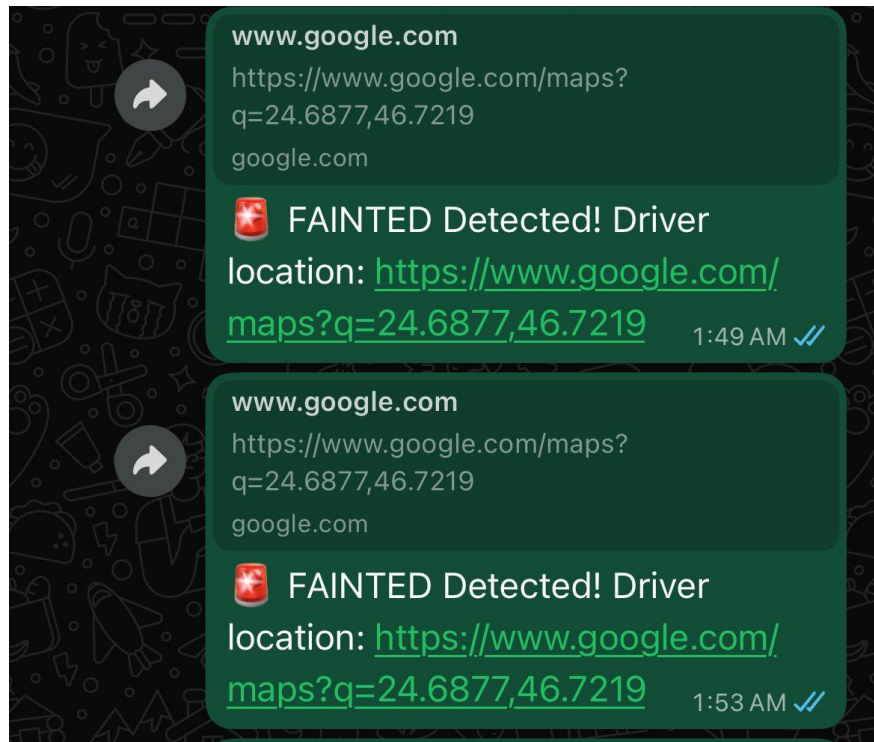


Figure 12 WhatsApp emergency Message with driver location

5.5 Discussions

Results show that the CNN model can successfully identify open vs. closed eyes and hence awake vs. drowsy states.

Strengths:

The model trained and tested well.

Real-time merging was achieved with low-latency phases.

Single frame accuracy was significantly higher for Mediapipe than traditional technique for eye detection.

Limitations:

Model accuracy is limited in low-light conditions or when the user wears glasses.

The model might need to be fine-tuned for other camera resolutions or angles.

Comparison with Other Studies:

In contrast to conventional Haar Cascade-based systems, CNN + Mediapipe is a much more robust and less error-prone eye detector.

5.6 Summary

In this chapter, we have provided the complete experimental analysis of our CNN-based drowsiness detection system. The method was trained and tested on a hybrid specific data set of face images and performed well on its classification task.

The real-time testing with a regular laptop webcam was successful, this proves the practical applicability of the model in controlled environments. The combination of audio warnings, positional trace and WhatsApp emergency notification stepped up the potential of being used in the system for future safe driving.

The results support the validity and effectiveness of our proposed approach, and indicate its potential to be deployed in safety-critical applications such as in driver monitoring systems and smart transportation, which in turn opens opportunities for real-world interaction.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

This research work introduces the process of an AI-based driver monitoring system to identify drowsiness and faint in real-time system based on a camera. Making use of Convolutional Neural Networks (CNN), the system was trained on a collection of real images of human faces from a diverse range of ages and AI-generated face images photographed from different angles. This variety of training data aided in improving system's robustness and generalization under real world environments.

The system monitors eye movements in real-time to identify signs of fatigue by characterizing eye states (open or closed). Also, one comprising an active reaction function for detecting irregular inactivity or fainting attacks. As soon as detected, the system activates a dual emergency protocol: sounding an immediate audible alert to the driver, and automatically alerting emergency services, which are provided the vehicle's GPS position to guarantee rapid intervention.

This proposed project is a practical application of deep learning and computer vision in the field of ITS. The coupling of real-time prediction, safety intervention and automatic emergency convenience reveals the applicability of AI in solving crucial public security issues. The results of this study provide a starting point for future improvements, such as integration with multi-modal sensors, field tests in real-world deployment, and extending to a larger scale of commercial vehicles. Eventually, this method endeavors to make significant contribution in decreasing the number of road traffic crash fatalities arising from fatigue-related and medical conditions & assist humanity as a whole which is an integral element of AI for social good.

6.2 Future Work

Although a strong potential of the Drive Guard system has been revealed for drowsiness and fainting detection employing visual analysis based on CNN, there are still multiple possibilities for future improvement. The addition of physiological sensors such as heart rate monitors or steering wheel grip sensors, which can enable a multi-modal technique that can enhance the accuracy of detection, especially when facial visibility is impacted adversely by lighting conditions or occlusion factors.

Furthermore, increasing and diversifying the data with additional real-world driving conditions, different illumination situations, and other age groups would improve the model's generalization and reliability. Also, it is interesting to study the real-time attention mechanisms or the transformer-based structure to achieve better responsiveness and false positive reduction for detecting fatigue and fainting in the future.

From a deployment point of view, optimizing the model for edge devices that possess constrained computational resources, such as in-car embedded systems, is essential for real-world use. Finally, field testing in simulated environments with actual drivers allows the system's performance to be verified under real dynamic conditions, as well as fine-tuning the emergency response process to meet local safety standards.

From these future directions, the system may become a complete and reliable solution that significantly contributes to improving road safety with the aid of artificial intelligence.

References

- (NHTSA), N. H. (2015). *Drowsy Driving Research*. www.nhtsa.gov.
- (2025, 5 27). Retrieved from drawio: <https://www.drawio.com/>
- (2025). Retrieved from leonardo.ai: <https://leonardo.ai/>
- (2025, 6 13). Retrieved from generated.photos: <https://generated.photos/datasets>
- Abdar, M. Z.-M. (2021). A Survey on Deep Learning-Based Driver Drowsiness Detection Systems. *Sensors*, 1991.
- Abtahi, S. O. (2014, 9 5). *YawDD: A yawning detection dataset*. Retrieved from <https://doi.org/10.1145/2557642.2563670>
- Ardabili, S. Z. (2024). A novel approach for automatic detection of driver fatigue using EEG signals based on graph convolutional networks. *Sensors*, 364.
- Awais, M. B. (2017). “A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve Road Safety”. *Sensors*, 1991.
- Babu, C. M. (2021). AI in Transportation: A Review of Safety and Monitoring Systems. *IEEE Transactions on Intelligent Transportation Systems*, 3124-3135.
- Closed Eyes In The Wild (CEW)*. (2025, 7 11). Retrieved from parnec.nuaa.edu.cn/_upload/tpl/02/db/731/template731/pages/xtan/ClosedEyeDatabases.html
- Dasgupta, A. G. (2019). “A Review on AI-Based Solutions for Driver Health Monitoring and Road Safety”. *IEEE Access*, 157237-157256.
- Desai, M. &. (2015). “Vehicle-Based Detection of Driver Fatigue Using Steering Behavior Analysis”. *Transportation Research Part C: Emerging Technologies*, 218-230.
- Dipu, M. T. (2021). Real-time Driver Drowsiness Detection using Deep Learning. *International Journal of Advanced Computer Science and Applications*, 8.
- Jiao, W. Y. (2023). A hybrid CNN-GRU model for real-time driver drowsiness detection. *Neural Computing and Applications*, 847–859.

- Lobo, A. a. (2020). Exploring Monitoring Systems Data for Driver Distraction and Drowsiness Research. *Sensors*, 3836.
- Patel, M. L. (2011). Applying neural network analysis on heart rate variability data to assess driver fatigue. *Expert Systems with Applications*, 7235–7242.
- Picot, A. C. (2012). On-line detection of drowsiness using brain and visual information. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 764–775.
- Russell, S. &. (2020). *Artificial Intelligence: A Modern Approach*. Pearson Education.
- Saini, V. a. (2014). Driver drowsiness detection system and techniques: a review. *International Journal of Computer Science and Information Technologies*, 4245--4249.
- Sheykhivand, S. a. (2022). Automatic detection of driver fatigue based on EEG signals using a developed deep neural network. *Electronics*, 2169.
- Sigari, M.-H. a. (2013). A driver face monitoring system for fatigue and distraction detection. *International journal of vehicular technology*, 263983.
- Sivaraman, S. &. (2013). Active Head Tracking and Pose Estimation in Intelligent Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 776-787.
- Wang, C. L. (2020). A survey of deep learning-based methods for drowsiness detection using physiological signals. *IEEE Access*, 107016–107029.
- Williamson, A. &. (2013). The Impact of Sleep Deprivation on Driving Performance. *Accident Analysis & Prevention*, 438-445.
- Zhang, Y. W. (2022). GAN-based data augmentation for driver drowsiness detection with limited datasets. *Computers in Biology and Medicine*.
- Zhang, Z. Z. (2021). Driver fatigue detection based on facial expression recognition and EEG signals. *Journal of Intelligent Transportation Systems*, 245-257. Retrieved from Journal of Safety Research: <https://doi.org/10.1016/j.jsr.2013.05.002>
- Zhao, C. W. (2020). “Facial Landmark-Based Detection of Driver Drowsiness Using Deep Learning”. *Expert Systems with Applications*, 113198.

Appendix A: Source Code for DriveGuard System

A.1 Model Architecture

```
import os
import numpy as np
import cv2
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import confusion_matrix, classification_report,
ConfusionMatrixDisplay

# Files Path
data_dir = r"C:\Users\X1 YOGA\Downloads\All_data\Train_Data"
img_size = 150

def load_images(folder_path):
    images = []
    labels = []
    classes = ['Awake', 'Drowsy']
    for idx, cls in enumerate(classes):
        class_folder = os.path.join(folder_path, cls)
        for file in os.listdir(class_folder):
            img_path = os.path.join(class_folder, file)
            img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
            if img is not None:
                img = cv2.resize(img, (img_size, img_size))
                images.append(img)
                labels.append(idx)
    return np.array(images), np.array(labels)

#Data Load
X, y = load_images(data_dir)
X = X / 255.0
X = np.expand_dims(X, axis=-1)
y = to_categorical(y)

# Data Spilt
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, shuffle=True,
random_state=42)
```

A.2 Model training

```
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(img_size, img_size, 1)),
    MaxPooling2D(2, 2),
    Dropout(0.25),
```

```

        Conv2D(64, (3,3), activation='relu'),
        MaxPooling2D(2, 2),
        Dropout(0.25),

        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(2, activation='softmax')
    ])

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

# Stop early
early_stop = EarlyStopping(monitor='val_loss', patience=5,
restore_best_weights=True)

# Augmentation at trining
datagen = ImageDataGenerator(
    rotation_range=5,
    zoom_range=0.05,
    width_shift_range=0.05,
    height_shift_range=0.05
)
datagen.fit(X_train)

```

A.3 Model Saving

```

history = model.fit(
    datagen.flow(X_train, y_train, batch_size=32),
    validation_data=(X_val, y_val),
    epochs=50,
    callbacks=[early_stop]
)

```

A.4 After Training

```

#Save model after the Train
import pickle
model.save('eye_model.h5')
with open('history.pkl', 'wb') as f:
    pickle.dump(history.history, f)

```

A.5 Model Loading

```

#Load Trained model
from keras.models import load_model
model = load_model('eye_model.h5')

```

A.6 Single Image Prediction

```

# Predictions
y_pred = model.predict(X_val)
y_pred_classes = np.argmax(y_pred, axis=1)

```

```

y_true = np.argmax(y_val, axis=1)

# Confusion matrix
cm = confusion_matrix(y_true, y_pred_classes)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Awake",
"Drowsy"])
disp.plot(cmap="Blues")
plt.show()

# Performance report
print(classification_report(y_true, y_pred_classes))

```

A.7

```

import cv2
import numpy as np
from keras.models import load_model

# Load the trained model
model = load_model('eye_model.h5')

# Load Image and Covert it to grayscale
image_path = r"C:\Users\X1 YOGA\Downloads\test9.jpeg"
img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

# Change Image Size
resized = cv2.resize(img, (150, 150))
resized = resized / 255.0
resized = np.expand_dims(resized, axis=-1)
resized = np.expand_dims(resized, axis=0)

# Pridect
prediction = model.predict(resized)
pred_class = int(prediction[0][0] > 0.5)

#Print Result
label = 'Drowsy ' if pred_class == 0 else 'Awake '
print(f'Prediction: {label}')

```

A.8 Live Camera Detection

```

import cv2
import mediapipe as mp
import numpy as np
from tensorflow.keras.models import load_model
import time
import pygame
import geocoder
import pywhatkit as kit

# Initialize sound system
pygame.mixer.init()
alarm_sound = r"C:\Users\X1 YOGA\Downloads\alert.mp3.wav"

# Load trained CNN model

```

```

model = load_model('eye_model.h5')
IMG_SIZE = 150 # Input image size for the model

# Initialize Mediapipe Face Mesh
mp_face_mesh = mp.solutions.face_mesh
face_mesh = mp_face_mesh.FaceMesh(static_image_mode=False,
                                   max_num_faces=1,
                                   refine_landmarks=True,
                                   min_detection_confidence=0.5,
                                   min_tracking_confidence=0.5)

# Start webcam
cap = cv2.VideoCapture(0)

# Preprocess eye image to feed into CNN model
def preprocess_eye(eye_img):
    gray = cv2.cvtColor(eye_img, cv2.COLOR_BGR2GRAY)
    gray = cv2.equalizeHist(gray)
    resized = cv2.resize(gray, (IMG_SIZE, IMG_SIZE))
    normalized = resized / 255.0
    return np.expand_dims(normalized, axis=(0, -1))

# Send location via WhatsApp in case of fainting
def send_emergency_message():
    try:
        g = geocoder.ip('me')
        location = g.latlng
        if location:
            message = f" FAINTED Detected! Driver location:
https://www.google.com/maps?q={location[0]},{location[1]}"
        else:
            message = " FAINTED! Driver location not available."

        kit.sendwhatmsg_instantly("+9660000000000", message, wait_time=10,
tab_close=True)
        print("Location sent ✓")
        print("Emergency contacted ✓")
    except Exception as e:
        print("WhatsApp/send error:", e)

# Timing control
drowsy_start = None
alarm_playing = False
fainted_triggered = False
last_state = None

# Main loop
while True:
    ret, frame = cap.read()
    if not ret:
        break

    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = face_mesh.process(rgb)

    label = "No face"
    if results.multi_face_landmarks:

```

```

for face_landmarks in results.multi_face_landmarks:
    h, w, _ = frame.shape
    left_eye_idx = [362, 385, 387, 263, 373, 380]
    right_eye_idx = [33, 160, 158, 133, 153, 144]

    # Extract eye region from landmarks
    def extract_eye(eye_points):
        x_coords = [int(face_landmarks.landmark[i].x * w) for i in
eye_points]
        y_coords = [int(face_landmarks.landmark[i].y * h) for i in
eye_points]

        x_min, x_max = max(min(x_coords) - 5, 0), min(max(x_coords) + 5, w)
        y_min, y_max = max(min(y_coords) - 5, 0), min(max(y_coords) + 5, h)
        return frame[y_min:y_max, x_min:x_max]

    # Get both eyes
    left_eye_img = extract_eye(left_eye_idx)
    right_eye_img = extract_eye(right_eye_idx)

    if left_eye_img.size != 0 and right_eye_img.size != 0:
        # Resize right eye to match left eye
        right_eye_img = cv2.resize(right_eye_img, (left_eye_img.shape[1],
left_eye_img.shape[0]))
        both_eyes = cv2.hconcat([left_eye_img, right_eye_img])
        eye_input = preprocess_eye(both_eyes)

        # Predict state using model
        prediction = model.predict(eye_input, verbose=0)
        label = "Awake" if np.argmax(prediction) == 0 else "Drowsy"
        color = (0, 255, 0) if label == "Awake" else (0, 0, 255)

        # Print state only if changed
        if label != last_state:
            print("Current state:", label)
            last_state = label

        now = time.time()

        if label == "Drowsy":
            if drowsy_start is None:
                drowsy_start = now
            elif (now - drowsy_start) > 2 and not alarm_playing:
                # Start alarm if drowsy > 2 sec
                pygame.mixer.music.load(alarm_sound)
                pygame.mixer.music.play(-1)
                alarm_playing = True

        # Consider as FAINTED if drowsy > 5 sec
        if (now - drowsy_start) > 5 and not fainted_triggered:
            print("FAINTED detected! Sending location and contacting
emergency!")

            print("Current state: FAINTED")
            send_emergency_message()
            fainted_triggered = True
            pygame.mixer.music.load(alarm_sound)
            pygame.mixer.music.play()
            time.sleep(15) # Play alert for 15 seconds

```

```

        pygame.mixer.music.stop()
        cap.release()
        cv2.destroyAllWindows()
        break

    elif label == "Awake":
        drowsy_start = None
        if alarm_playing:
            # Stop alarm when awake
            pygame.mixer.music.stop()
            alarm_playing = False

        # Display state on screen
        cv2.putText(frame, f"Status: {label}", (30, 40),
                    cv2.FONT_HERSHEY_SIMPLEX, 1.2, color, 3)

        # Draw rectangle around both eyes
        x_min = min([int(face_landmarks.landmark[i].x * w) for i in
left_eye_idx + right_eye_idx]) - 5
        x_max = max([int(face_landmarks.landmark[i].x * w) for i in
left_eye_idx + right_eye_idx]) + 5
        y_min = min([int(face_landmarks.landmark[i].y * h) for i in
left_eye_idx + right_eye_idx]) - 5
        y_max = max([int(face_landmarks.landmark[i].y * h) for i in
left_eye_idx + right_eye_idx]) + 5

        x_min = max(x_min, 0)
        y_min = max(y_min, 0)
        x_max = min(x_max, w)
        y_max = min(y_max, h)

        cv2.rectangle(frame, (x_min, y_min), (x_max, y_max), color, 2)

    # Show video frame
    cv2.imshow("Driver Monitor", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release resources
cap.release()
cv2.destroyAllWindows()

```