



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.....	17
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	17
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 tiredness and fatigue to which lead to any extent, a loss of reaction time and alertness in those drivers who drive 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 braking) (Desai, 2015). These approaches, however, cannot offer immediate contextual assessments of a 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 surely, the AI of driver monitoring is an important technological door closer that has to be opened for the purpose of safety of automotive. With the commensurate results of real-time analyses of physiological changes in a driver (like drowsiness and distraction) behind the wheel, this DMS is AI-enabled.

1.1 Problem Definition

Accidents caused by impaired drivers caused by drowsiness and fainting is a big issue today due to insufficient road safety. Numerous studies have established that less alertness of drivers is causatively linked to road accidents, and are highly concerned regarding preventing these situations from ever reaching the point of crisis.

The Drive Guard project is an AI-powered platform, which addresses these concerns. Traditional systems are only capable of providing warnings about drowsiness, whereas "Drive Guard" can detect the earliest signs of fatigue, as well as even fainting.

The platform takes advantage of high-resolution cameras and sophisticated machine learning algorithms, calibrated to recognize the most subtle variances in a person body language appropriateness, facial expressions, eye movement and other behavior cues to provide people with real time feedback and alerts about their monitored behavior. This idea, which can significantly

help curb accidents stemming from fatigue, can make the roads safer for us all and potentially save lives.

Drive Guard is a unique way to enhance driver monitoring and road safety by integrating real-time AI-based monitoring systems with an automated emergency handling response action system, allowing us to explore a systematic improvement of accident risk from driver impairment, utilizing proactive measures to prevent crash induced accidents in the first place and also speed up response time when a response to an incident is required.

1.2 Thesis Objectives

This study aims to develop a driver monitoring system that improves road safety through the detection and mitigation of drowsy and medical emergencies. The goals are to:

Create a real time AI model whereby we can identify facial gestures, eye movements, and body language to detect deviations in the driver state such as drowsiness and unconsciousness.

Be able to distinguish between drowsiness and fainting by identifying unique behavioral and physiological profiles which distinguishes the two states.

We can suggest the build of a real-time alarm to sound the driver to their mapping capabilities when he/she becomes drowsy, thereby stopping a total loss of what the driver is truly doing while on the highway.

Create an automated emergency response which:

- Will send an SOS Distress Signal to emergency services
- Will send the drivers GPS location which enables a faster response from rescuers

The system robustness and accuracy across conditions (day vs night, different illuminations, different driving speeds)

Gain insights into creating solutions for AI driver safety technologies by addressing some the current gaps in research.

1.3 Thesis Contribution

This thesis highlights some significant contributions to the area of driver drowsiness detection using deep learning methods:

Eye State Detection Using CNN

The project is a real-time eye state detection system using Convolutional Neural Network (CNN), which is an efficient behavioral cue for drowsiness.

The use of dataset

Training the dataset includes real human face-eyes from subjects with a wide range of ages, races, and view angles, and also synthetic images made with cutting edge AI algorithms. This is to obtain model robustness and generalization for different driver profiles to increase diversity in the dataset and maintained density of the dataset.

Optimized CNN architecture

A custom CNN model for eye state classification in different lighting and environmental challenges, achieving a pleasant accurate and stable output

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. Napping drunk drivers stop the car with the closing of the eyes alone; "Drive Guard"

employs advanced computer vision and machine learning algorithms to process a range of drowsiness signals that include eye movement patterns, the facial expression of the driver, head tilt angles and body purses.

There are two primary modes of action:

Drowsiness detection - If the system detects drowsiness, it issues immediate aural alarms to the driver when they detect symptoms such as slow eye movements, prolonged blink duration, and nodding or head buckling to help refocus the driver.

Fainting detection & Emergency reaction: The system detects symptoms of drowsiness and dismisses it after prolonged closing ($> 15s$) and subsequently an absence of valid facial expressions, and there was no input of moving. The system will label it as a probable faint 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.

Facial landmarks may be represented as nodes in a graph in driver monitoring and spatial and functional relationships as edges. The learned representation in GCNs utilizes such relational characteristics meaning that it can accurately recognize facial movements or subtle variations in head pose and occlusion. (Wang, 2020)

Commonly modern driver monitoring systems utilize a suite of data sources, including video, physiological signals, and contextual information (e.g. environmental conditions, time of day, etc.). The usage of multimodal fusion techniques combines complementary inputs for greater reliability and to minimize false positives. For example, each facial video analysis integrated with heart rate variability gives a better understanding of driver state than either modality on its own. (Patel, 2011)

There are also advantages to context-centric systems that adapt sensitivity of the system based on context and driver (e.g., alertness detection sensitivity is increased when the driver is driving in darkness or after sustained driving). There are enhancements to the user interface to reduce driver distraction due to false alarms and is more acceptable to the user. (Picot, 2012)

Nevertheless, in the on-road context of further deployment the robust development of good drowsiness detection systems to various driver profiles has encountered substantial challenges. Different facial characteristics, social practices, lighting, and position and placement of the sensors all hinder model generalizability. Practical issues also exist with limited computational resources, concerns about data privacy and the need for real-time systems. (Wang, 2020)

Current research is also exploring the interpretability of deep learning models to enhance and simplify understanding for consumers and regulators. There are also looking at adapting the system to adjust the detection thresholds based on individual driving profiles and driver habits. Development of vehicle control systems with pre-emptive measures to enhance safety, e.g., adaptive cruise control or automated braking is also an exciting future direction. (Zhang Z. Z., 2021)

Overall, the history of drowsy driver detection systems has transformed from passive developments in simply behavior monitoring, to intelligent, proactive systems with advanced Ai algorithm and multimodal data fusion technologies,

which promise to save lives Vetter than serious injury from fatigue and fatigue-related crashes and can make the world roads safer.

In recent years, the urgent public safety concerns and advancements in intelligent transportation systems, have resulted - so far - in driver monitoring system from conception to a merger of great technological advances. Early on - Driver monitoring had a focus on explicitly observable external behaviors of driving like general lane-keeping, patterns and correcting steering to remain in lane, because these are very easy to collect from existing data the car gives, but does not always capture the more subtle but equally damaging - internal driver states (e.g., tiredness, cognitive distraction). The next iteration of driver-monitoring systems went to in-cabin monitoring with cameras and bio-sensors to directly provide insights into the driver's perceived physiological or mental well-being.

These technologies and improvements of hardware, such as IR based night vision cameras and multi-function EEG headbands, have advanced sensors for monitoring the user in the vehicle. This was one of the limiting factors with historic systems while today's technologies allow for data collection, even in low-lighting and noisy environments. Further, the availability of richer annotations has allowed better models to be developed for modeling both on-road rider and environmental behavior allow for modeling across population groups, vehicles, and scenarios.

In a similar fashion, multimodal data has been recognized as another powerful mechanism to improve the resiliency of a system. For example, we can look at combinations of facial landmarks with vital signals like heart rate variability (HRV) or respiration rate and increase sensitivity for subtle fatigue changes that may not be captured through facial data alone. By fusing signals across all these modalities, the system is responsive to different use cases where one of the modalities may be more reliable based on lighting, head orientation, or some other driver specific features. This absolutely improves user acceptability as it increases the potential for adaptation and makes erroneous false positives less likely.

Another interesting area of research is about the personalization, or individualization, of driver monitoring systems. Because each person's behavior in response to muscle fatigue is highly dynamic, typical static thresholds for monitoring behavior usually don't capture intersubject

variability effectively. To address this, adaptive systems are being designed and will learn an individual's baseline over time and adapt sensitivity accordingly. If a driver has a higher natural blinking frequency for example, their individual characteristics may not fall outside of typical parameters unless they blink above a critical unique threshold. Having that individualized calibration makes the system much more trustworthy, and probably accurate.

At the same time, there is research into processing AI in-vehicle running from edge computing and low-power embedded devices to enable real-time inference. Edge computing allows for instantaneous response times and improved privacy versus cloud systems where biometric data must necessarily be sent to the cloud for processing. Optimized deep learning models too (for example, Mobile Net, Squeeze Net and Tiny-YOLO) are reducing barriers for viable solutions. Surely much of the attention has focused on driver drowsy and inattentive monitoring and as a response there have been some attempts that either became timely or fall into the category of either (unobtrusive but preparative).

Researchers note that a warning could become a false positive and drivers soon learn to ignore them or develop alert fatigue. For this reason, a growing number of intelligent alert systems are being designed to calculate implications of the driver's reaction, and proportion the level and frequency of feedback. A first sign of fatigue could generate a gentle chime; at other severe levels, a loud alarm or even vibrating padding in the seat.

One major challenge for good drowsiness detection system design is to find the balance between sensitivity and specificity. An extremely sensitive system, while able to identify every incident of fatigue, is likely to produce too many false positives which may disturb the driver and/or cause him/her to disable the system. An extremely specific system, on the other hand, may not deliver false positives but will perform poorly in identifying the first signs of drowsiness. This is the arduous part, and you need a lot of testing for feasible labeled datasets that employs proper handling of driving scenarios.

However, producing a decoder accurately is not always simple. Individuals incorporate cultural norms, age, gender and driving habits. A younger driver may display more noticeable facial features in comparison to an older driver or eye contact and posture vary between cultures. Therefore, the combination of cross-cultural studies and the data can be a useful tool to make sure it is not

bias training sets and to ensure that the models developed in those sets will act reasonably similarly for all individuals.

Additionally, simulation-based training and testing platforms are increasingly becoming main components in the process of developing and deploying these types of systems.

Virtual driving simulators give researchers the opportunity to recreate hazardous scenarios where they can induce fatigue and record the event. Researchers will acquire not only better data but can also keep adding additional algorithms in a place for testing in parallel conditions. Also, these virtual simulators are being increasingly leveraged in driver educational programs to educate the occurrence and risks of drowsy driving and to teach drivers knowledge of how best to respond to fatigue warnings as well.

There is also driver aspects that are of concern when using emotional state analysis in driver monitoring. Stress, anger, or anxiety can plague and compromise our driving performance on their own, and how these dramaturgic states relate to one's fatigue state is not only complicated, it could also be incredibly dangerous too. We see every day, efforts from AI systems attempting to be trained to identify emotional responses via facial expressions, vocal intonation for mood or even physiology changes. Integrating this context within an episodic drowsiness detection framework will add useful context that may lend to earlier identification of these high risk's states.

Not to mention unsupervised and semi-supervised learning have shown promise in making use of unlabeled data and in reducing the cost of labeled data, while always laborious and a costly acquisition. These strategies allow the model to pick out patterns in these unlabeled forms of driving behaviors and improve as it sees more real-world data over time. This kind of learning strategy, may be especially relevant in the case of personalized systems depending on the demands of the driver, since its an unrealistic assumption too much calibration phase would be warranted.

As the development and adoption of autonomous and semi-autonomous vehicles rises this decade, so will the requirement for better driver state monitoring. Many vehicles must ALWAYS expect the operator to take over (Level 2 and Level 3 Automation). Therefore, platforms such as Drive Guard could act as a sentinel for the operator, ensuring they are ready to take control should danger arise. This will be critical to ensure that failures and

misjudgments associated with system-induced ambiguities are eliminated, as we transition to human control to machine.

In sum, the growing body of research and development in artificial intelligence driver monitoring fits very well into a global maturation of thinking from humans, and organizations, being the responsible party for road safety. The organizations sponsoring these systems are early-stage components in this broad maturation of not only technical and technological solutions, but communities with a more capable, safer, and responsible transportation ecosystems. Importantly, the combination of advanced algorithms with real-time sensing and user-centered place, is increasing the likelihood of “Driver Drowsiness Detection Systems” becoming a standardized safety feature in the cars of the future.

The last decade has also experienced a paradigm shift in computer vision, with the recent rise of deep learning, particularly with Convolutional Neural Networks (CNNs), which allows systems to efficiently automatically extract significant spatial features from images or frames with low human intervention. CNNs are very useful for this problem, because while trying to classify driver drowsiness from video (the containment of changes to corners of the eye, mouth and eyebrows that vary in turn with fatigue) is complex, it does require some good features to accomplish this. CNNs are trained with large datasets that classify the alert and drowsiness facial expressions that can sometimes be too subtle for human recognition.

In contrast to standard machine learning methods that require thorough feature engineering, CNNs learn a hierarchy of features that allow abstraction from the raw visual information (lines that lead to facial line structures). Due to the hierarchical approach provided in CNNs the proposed data driven methods would be robust for even partial occlusions (sunglasses) or changes in illumination and pose. Furthermore, even more recent CNN architectures such as ResNet, Inception and Efficient Net have provided better accuracy under reduced computation costs—a sweet point for purpose designed real-time embedded automotive systems.

Chapter 3: Related Work

Drowsy and unconscious drivers are regarded as a significant factor in fatal accidents on a global scale. Many researchers have developed different countermeasures to help detect and ultimately prevent these occurrences over the past decade and these approaches can be better understood. This chapter provided an overview of various previous works and technologies for detecting impairments and fatigue.

It highlights that "driver drowsiness" is one of the primary causes of road casualties and is involved in approximately 20% of all accidents, resulting in over 1,200 fatalities every year. It stresses the importance of sound drowsiness detecting technologies to improve vehicle safety.

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 developed a dataset of approximately 4,500 images of drivers in various states of alertness. The work documented in this approach

to authenticity is also similar to one they describe in which the accuracy, computational efficiency, and cost allow for such a solution to be applied with inexpensive hardware (e.g. a Raspberry Pi or generic cameras) in a much broader application of drowsiness detection technology, particularly in automobiles without sophisticated safety systems. It provides an overview of the methods used in the research from obtaining the data, to algorithm design, and a review of evaluation metrics with the goal of learning to contribute to ongoing efforts in reduction (Dipu, 2021)

Focuses on the development of deep CNN–LSTM network for the identification of driver fatigue through EEG signals. The EEG data applied by the model is based on driving simulation with 11 participants with data filtering and standardization. The model trains the classification of tabular data using a neural network combination of CNN and LSTM layers.

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. It has serious implications for measuring fatigue and attention, both important in enhancing public safety, and reducing road accidents associated with driving while sleepy or distracted. The system can reduce accidents by analyzing facial and eye symptoms, and notifying drivers of their hypo vigilance state. The paper provides a detailed review of previous work

on driver monitoring systems (DMS), and a description of methodology and experimental results, which strongly suggest likely positive implications for 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)	-Generative Adversarial Networks (GAN)	91.7% To 99.4%	20 participants in five different classes

	-EMG signals (Electromyography) -Respiratory effort signals	-Graph Convolutional Networks (GCN)		
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)

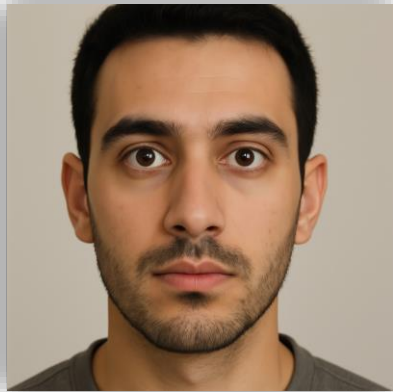


Figure 3 Sample image showing the “open eyes” 3

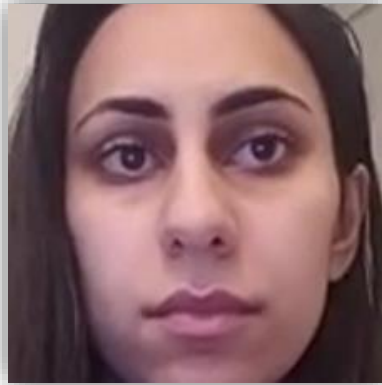


Figure 2 Sample image showing the “open eyes” 2



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.

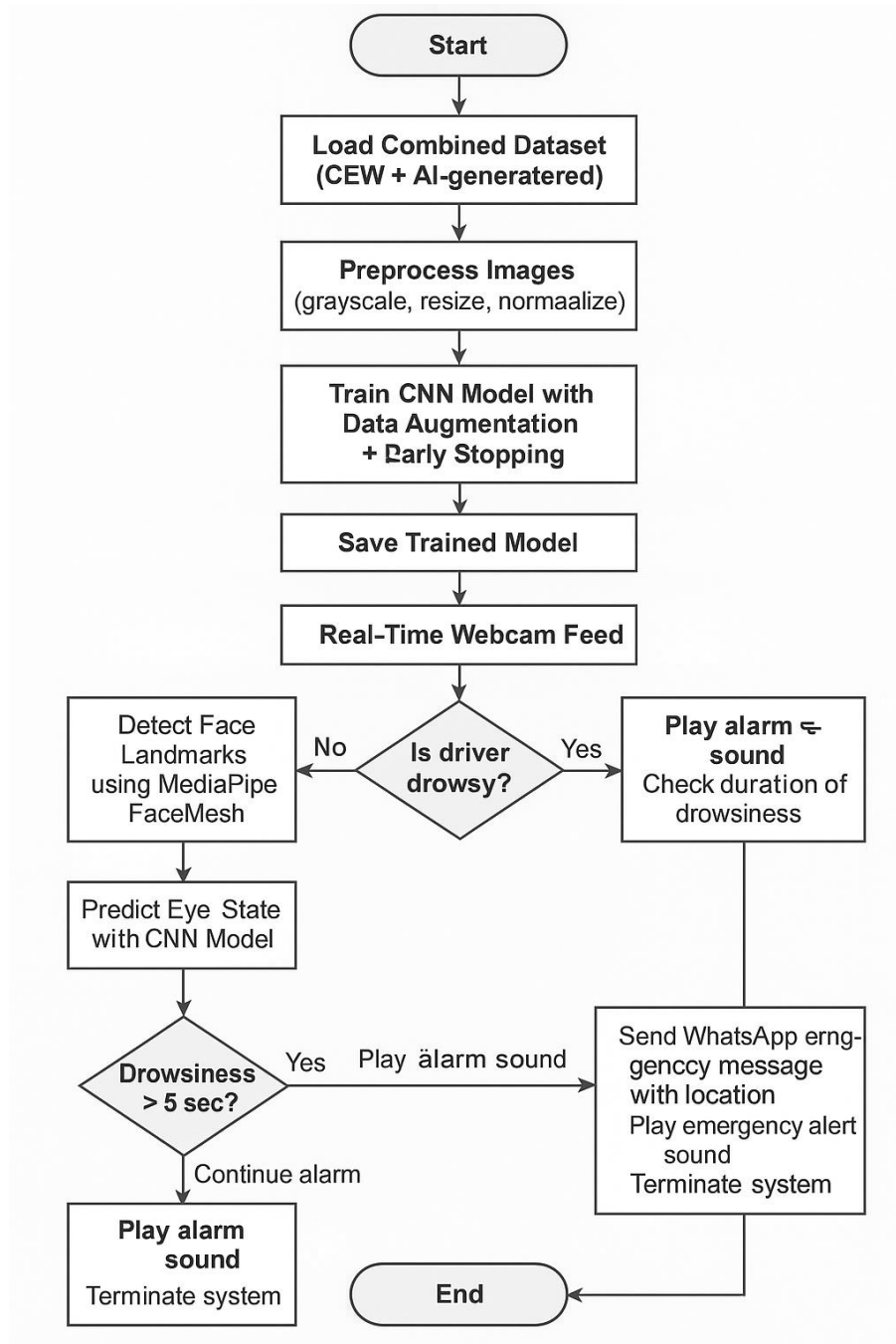


Figure 7 System Workflow Diagram

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.

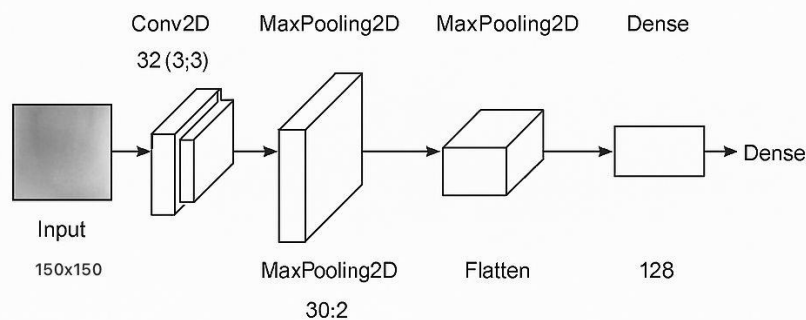


Figure 8 CNN Model Architecture

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.

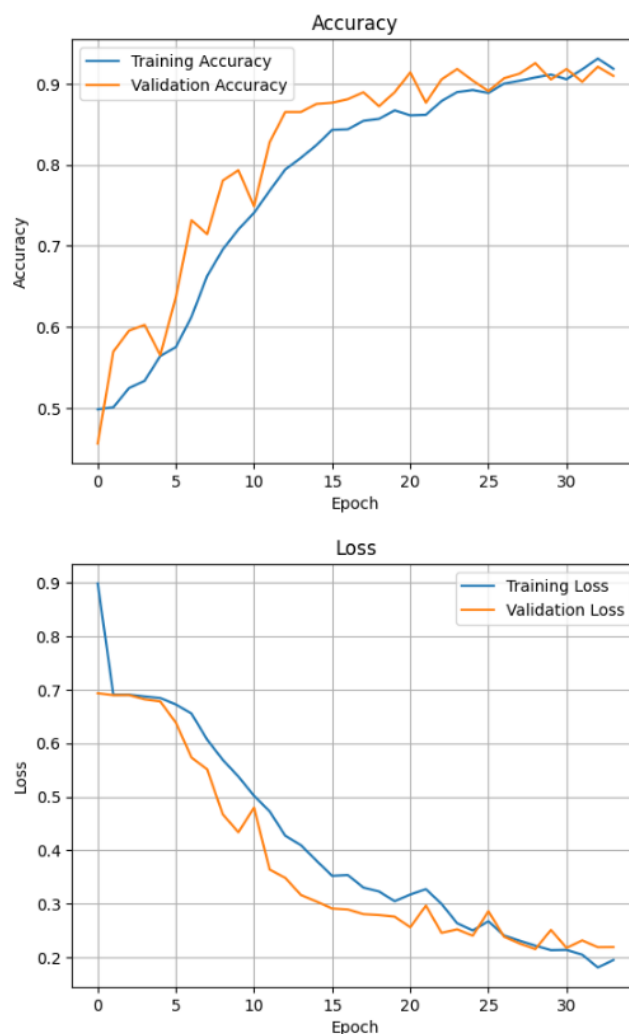


Figure 9 Accuracy & Loss Function

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.

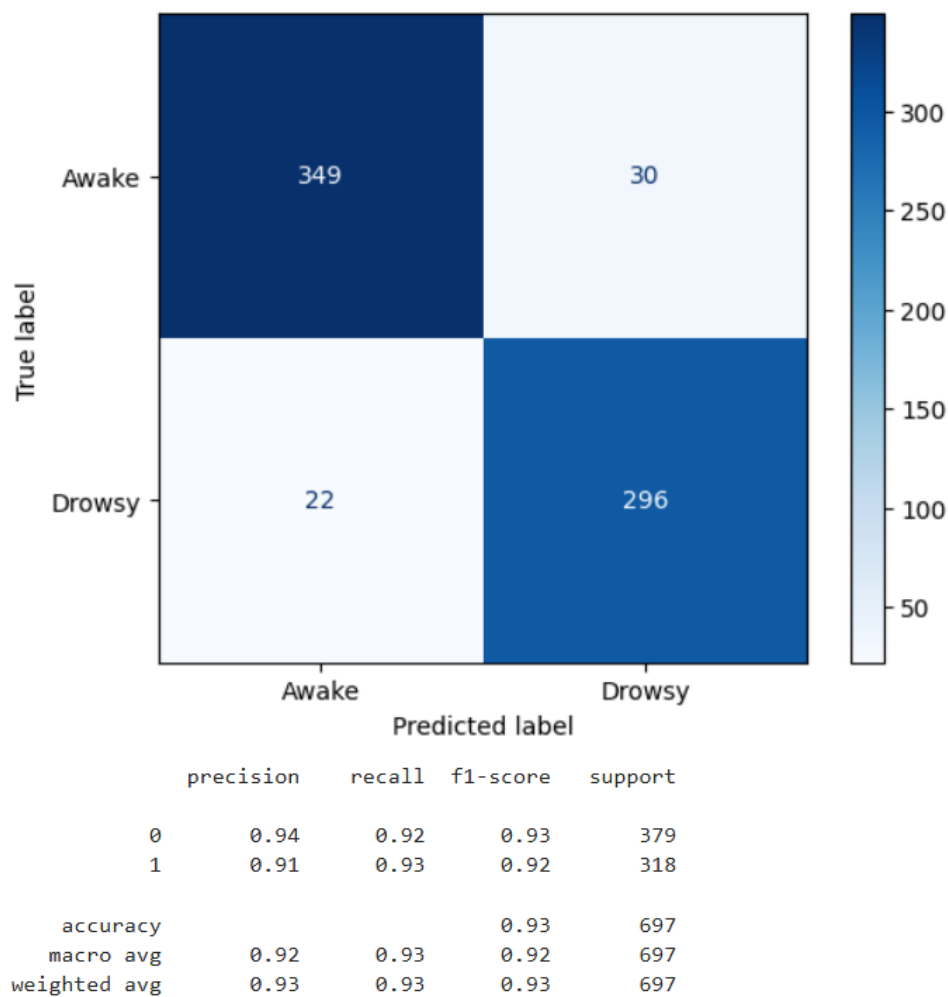


Figure 10 Confusion matrix

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

Figure 11 Real-Time Output

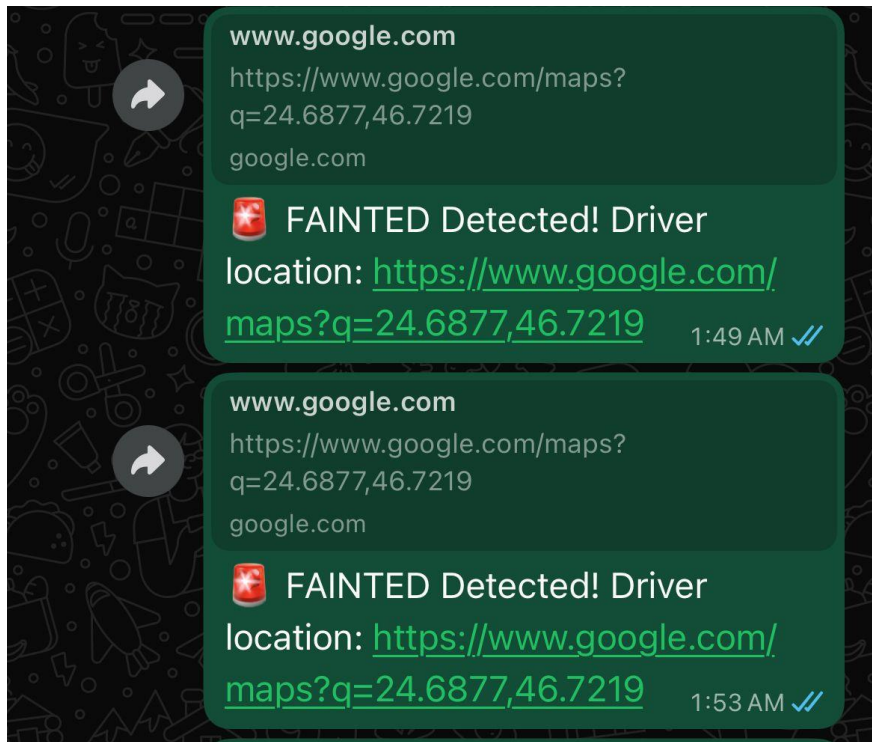


Figure 12 WhatsApp emergency Message with driver location

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.

Real-time trials with a standard laptop webcam were conducted successfully and indicated the real-world feasibility of the proposed model in controlled settings. The implementation of auditory warnings, positional trace, and WhatsApp emergency warnings contributed to the feasibility of being integrated in the system for future safe driving.

The results reinforce the validity and effectiveness of our proposed method and demonstrate the potential to be considered in safety-critical contexts such that is the application in driver monitoring systems, smart transport, which opens avenues for real-world interaction.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

This research study includes the implementation process of an AI-Driver Monitoring System to detect drowsiness and fainting in real-time system using a camera. By utilizing Convolutional Neural Networks (CNN), the system trained with a range of real human face images of varying ages and AI-generated facial images from differing angles. Having a variety of training data for the system increased the system's robustness and ability to generalize in real world environments.

The system monitors the movement of the eyes with the goal of detecting signs of fatigue, characterizing an eye state (open and closed). Also, one which has an active reaction function of detected unusual long inactivity of inactivity, or fainting episodes. On detection, the two-emergency system notification occurs: a real-time audible emergency sounds for the driver, and an alert to emergency services who will receive the vehicle's GPS position to allow quick deployment. This proposed project is a real-world application of deep learning and computer vision in the field of Intelligent Transportation System (ITS). The combination of predicting in real-time, safety notification and automatic emergency benefit shows the possibility of using AI as a benefit to solving crucial public safety issue.

The outcome of this study presents opportunities for progression, like integrating multi-modal sensor, field based of real-world deployment and scaling in large commercial vehicles. Ultimately, this process aims to satisfy ably contribute to reducing the number of road traffic crash deaths caused by fatigue-related and medical incidents and to assist humanity as a whole which is a core part of AI for Social good.

6.2 Future Work

While the Drive Guard system functionalities have been shown for drowsiness and fainting detection based on visual analysis using convolutional neural networks (CNN), much room for improvement in the future exists. Incorporating device-based physiological sensors (e.g., heart rate sensors, steering wheel grip) would add a multimodal approach which could improve detection accuracy when facial-based visibility is adversely affected due to lighting conditions and/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()

```