Driver Drowsiness Detection Using Deep Learning

Thesis

Submitted in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

SAURABH KUMAR SINGH

(222CS029)

Under the Guidance of

Prof. Annappa B

Department of Computer Science and Engineering,

NITK, Surathkal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,

SURATHKAL, MANGALORE - 575025

June, 2024

DECLARATION

I hereby declare that the P.G **Major Project Work Thesis** entitled **Driver Drowsiness Detection Using Deep Learning** which is being submitted to the National Institute of Technology Karnataka Surathkal, in partial fulfilment of the requirements for the award of the Degree of **Master of Technology in Computer Science and Engineering** in the department of **Computer Science and Engineering**, is a bonafide report of the work carried out by me. The material contained in this Report has not been submitted to any University or Institution for the award of any degree.

Saurabh Kumar Singh
222CS029
Department of Computer Science and Engineering
NITK, Surathkal

Place: NITK, Surathkal.
Date: 06.06.2024

CERTIFICATE

This is to certify that the P.G. **Major Project Work Thesis** entitled **Driver Drowsiness Detection Using Deep Learning** submitted by **Saurabh Kumar Singh 222CS029** as the record of the work carried out by him, is accepted as the P.G. Project Work Thesis submission in partial fulfilment of the requirements for the award of degree **Master of Technology in Computer Science and Engineering** in the Department of **Computer Science and Engineering**.

Prof. Annappa B

Project Guide &

Professor,

Computer Science & Engineering,

National Institute of Technology,

Karnataka – 575025, India.

Dr. Manu Basavaraju

DPGC Chairman &

Associate Professor,

Computer Science & Engineering,

National Institute of Technology,

Karnataka – 575025, India.

ACKNOWLEDGEMENT

I would like to express my heartfelt appreciation to my research supervisor, **Prof. Annappa B**, *Professor in the Department of Computer Science and Engineering at NIT Surathkal*. His unwavering guidance, support, and encouragement throughout my research work have been invaluable. I am grateful for his systematic and well-planned approach, which motivated me to stay on track and complete my research within the given timeframe. His insightful suggestions and timely feedback played a significant role in the smooth execution of my research. His constant motivation, appreciation, and flexibility allowed me to overcome challenges and achieve success in my work.

I would also like to extend my thanks to the Head of the Department, **Dr. Manu Basavaraju**, as well as the faculty members and staff of the *Department of Computer Science and Engineering at NIT Surathkal* for their assistance and support throughout my research journey.

Additionally, I am grateful to my research mentor, **Saraswati Koppad**, a research scholar in the *Department of Computer Science and Engineering at NIT Surathkal*, for her kind guidance and support during the course of my research.

Saurabh Kumar Singh (222CS029)

NIT Karnataka

Abstract

A National Sleep Foundation poll found that a startling 20% of drivers admitted to driving while sleep deprived, highlighting alarming statistics suggesting a worrying correlation between sleepy driving and accidents. This corresponds to a markedly elevated risk of collision, underscoring the pressing requirement for inventive approaches to reinforce traffic safety. This study investigates a driver drowsiness detection system using multiple deep learning architectures for real-time analysis of eye states and facial expressions. The effectiveness of InceptionNetV3, EfficientNetB2, MobileNetV2, and a stacked ensemble model in discriminating between alert and drowsy drivers is evaluated using the Media Research Lab (MRL) Eye dataset. The objective is to train models that can discriminate between drivers who are alert and those who are sleepy.

This research not only emphasizes the critical road safety risks posed by driver drowsiness, but it also demonstrates the transformative potential of deep learning in driver state monitoring. This research advances the possibility of driver drowsiness detection systems that deliver early alerts, thereby reducing accidents, saving lives, and improving everyone's driving safety.

Keywords: Deep Learning, Driver Drowsiness Detection,InceptionV3, EfficientNetB2, MobileNetV2, Stack Ensemble, Tensorflow

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

DDD	Driver Drowsiness Detection
CNN	Convolutional Neural Network
MRL	Media Research Lab
NHTSA	National Highway Traffic Safety Administration
EEG	Electroencephalogram
EOG	Electrooculography
EMG	Electromyography
ADAS	Advanced Driver-Assistance Systems
pBCI	passive brain-computer interface
CNN	Convolutional Neural Network

Chapter 1

Introduction

Drowsy driving is a serious safety risk that puts lives on the road in danger. The National Highway Traffic Safety Administration (NHTSA) estimates that drowsy driving causes 100,000 police-reported crashes each year, with 50,000 injuries and about 800 fatalities as a result [11]. Drowsy driving is still a serious hazard on our roadways. To improve road safety, this worrying trend demands creative solutions. This study explores a potentially useful method for detecting driver fatigue that makes use of deep learning.

Through the examination of live data obtained from a camera positioned strategically inside the car, we are able to derive important information about how attentive a driver is. This information consists of eye activity (yawning and blinking), minor head movements, and facial expressions. The selected methodology presents multiple benefits: it is non-invasive, economical, and necessitates less extra apparatus in contrast to conventional techniques that depend on physiological sensors.

In this study, we explore three pre-trained deep learning architectures, InceptionV3, EfficientNetB2, and MobileNetV2, with different activation functions in their final layers. We investigate both single models and stacked ensemble model, which combine the outputs of multiple models to find the best combination for accurate driver drowsiness detection. Through analysis of experiment results, this study seeks to improve deep learning-based driver drowsiness detection systems In the not too distant future, imagine a world in which in-car technologies are able to identify fatigue and promptly notify drivers, averting collisions, sparing lives, and making driving safer for everybody.

Chapter 2

Literature Survey

The field of enhanced driving and driver drowsiness detection is examined by Elena Magan et al. [9]. assistance systems (ADAS), reveals a complex field of study. Previous research has employed a variety of approaches to identify driver fatigue, including physiological signals, metrics derived from the vehicle, and vision-based methods. Author employ a non-intrusive technique in their paper to warn sleepy drivers without causing unnecessary concern. With the use of two primary approaches—a neural network model and a combination of fuzzy logic and deep learning-based feature extraction—the study demonstrates similar accuracy rates of roughly 60% on test data and 65% on training data. Significantly, the system based on fuzzy logic demonstrated a 93% specificity, which drastically reduced the number of false alarms. The study used deep learning techniques in conjunction with recurrent and convolutional neural networks to analyze facial image sequences that were taken at 60-second intervals and extract features. Even while the results show promise, the study admits to moderate overall accuracy rates, suggesting room for improvement. As a result, the suggested approaches encourage continued development to increase detection precision while also providing a strong foundation for future research.

M Suriya et al. [16] conduct a thorough investigation of the detection of driver drowsiness, highlighting its critical significance in traffic safety and arguing for the urgent need for efficient detection technologies. Their thorough review of the literature demonstrates the wide range of approaches used in earlier research, from CNN-based models to EEG(electroencephalogram) sensors, all with the goal of tracking physiological and psychological states to reduce accidents. Their research presents an innovative strategy by combining a CNN-based hybrid model with EEG sensors in an effort to precisely detect driver fatigue levels and deliver real-time notifications. The model is able to distinguish between stages of weariness and drowsiness by carefully observing EEG signals, which provide direct insights into the driver's brain activity, and by utilizing CNN algorithms' potent pattern recognition capabilities. It shows how CNN algorithms and EEG signals may work together seamlessly by sending out timely notifications to stop mishaps. By identifying driver tiredness with high accuracy and efficiency, this integration eventually improves road safety. Furthermore, the model notifies drivers of their tired or sleepy conditions and may prevent accidents by sending signals to the driver's mobile device and turning on alerts in the vehicle system. Even with these encouraging outcomes, real-world application still faces difficulties due to things like individual EEG pattern variability, outside interferences, and the requirement for discreet sensor location. It is imperative to do additional validation and testing on a range of driving conditions and demographics. This underscores the necessity of continuous refining and optimization endeavors to guarantee the model's dependability and efficiency in real-world situations. By tackling these issues, the research hopes to make a substantial contribution to the advancement of increasingly complex and dependable driver assistance systems that have the potential to save lives and lower the frequency of traffic accidents.

Praveen Tumuluru et al. [19] emphasize how vital it is for road safety and how good detection systems in cars are required to stop accidents brought on by sleepy drivers. This study presents a stacked ensemble model that makes use of lightweight models, such as SqueezeNet, ShuffleNet, and MobileNet-V2, with the goal of achieving accurate and efficient drowsiness detection without placing an undue computing strain on vehicle systems. Using a stacked ensemble technique, the study maintains low computing complexity while improving overall accuracy by pooling the predictions of several lightweight models. This approach is especially useful for real-time sleepiness detection systems, when speed and accuracy are critical factors. By balancing these two aspects, the stacked ensemble model aims to provide dependable detection without sacrificing the functionality of the vehicle's onboard equipment. The NTHU-DDD dataset, a popular dataset for studies on driver drowsiness detection, was used for the experiments. The results of these experiments show how successful the suggested paradigm is. In comparison to more conventional, heavyweight models, the chosen lightweight models, MobileNet-V2, SqueezeNet, and ShuffleNet, are renowned for having shorter inference times and lower processing complexities. These models work best when coupled in a stacked ensemble, which enhances detection performance. The study's findings show that the stacked ensemble model has a promising performance in identifying driver fatigue. It strikes a compromise between reduced computational demands and excellent accuracy, which is necessary for realistic deployment in real-world situations. The model guarantees that it may be deployed on regular vehicle hardware without requiring specialist, high-performance computing resources by minimizing the computational load. The report does, however, admit several shortcomings. The lack of a thorough examination of implementation issues in the real world is one major drawback. Although the suggested model performs well in a controlled experimental environment, there are more difficulties when using it in a variety of real-world driving scenarios. Variations in illumination, dissimilar driving styles, and background noise can all affect how well the tiredness detection system performs.

Driver sleepiness is addressed with a unique strategy by Saad Arif et al. [2]. Their work focuses on detecting driver drowsiness through the analysis of electroencephalogram (EEG) signals utilizing a passive brain-computer interface (pBCI). EEG signals provide an unobstructed view of brain activity and can identify sleepiness by altering certain frequency bands, such as δ, θ, α, and β. Simulated driving activities with EEG monitoring are used in the study. The EEG data is processed by feature selection techniques to retrieve pertinent information that is then fed into machine learning classifiers. By using optimal ensemble models, the researchers were able to classify drowsiness with an astounding 85.6% accuracy. Strong performance measures (precision, recall, F1-score, specificity, and AUC) are also displayed by the model. This research's spatial localization of sleepiness detection is an important component. The study finds that the F8 site in the right frontal brain is particularly useful for detecting drowsiness-related alterations by carefully positioning EEG electrodes there. This enables driver alerts and targeted actions. There is hope for practical uses for the EEG-based pBCI system. It provides an unobtrusive means of tracking tiredness in real time and sending out early alerts to stop mishaps. Adding such devices to cars might greatly improve traffic safety. But there are restrictions. Although it works well in controlled environments, more testing is required in a variety of driving situations. For practical implementation, real-world issues like individual variances in EEG and environmental variability must also be addressed. Finally, author offer a novel method for detecting sleepy driving using pBCI that is based on EEG. Their study demonstrates how early detection and neurophysiological cues might lead to increased road safety. This work provides a viable answer to the pressing problem of sleepy driving, opening the door for safer roads and fewer accidents, by utilizing powerful machine learning and spatially localized data.

A comprehensive literature review exploring the complex field of real-time driver drowsiness detection systems is provided by Yeresime Suresh et al [15]. Their thorough analysis not only highlights how crucial it is to recognize drowsiness as soon as possible in order to minimize auto accidents, but it also charts the development of detection methods throughout time. The survey starts with conventional approaches that rely on behavioral cues and moves on to show the rise of increasingly complex automated systems, especially those that use deep learning algorithms. By looking at a number of research, the authors clarified the usefulness of putting sleepiness detection models into practice. They also presented low-weight designs that work well for real-time monitoring, including Raspberry Pi implementations. Additionally, the survey delves into the nuances of various detection strategies, covering tactics such as using facial landmarks and adaptive threshold techniques for sleepiness detection. The authors offer insightful information on areas that are ready for more study and development by addressing the drawbacks of current approaches, such as the requirement for flexibility and the dependence on particular traits like face landmarks. The survey also highlights the increasing trend of deep learning algorithms being used in drowsiness detection systems, realizing that these algorithms can improve road safety by enabling timely alerts to drivers who are experiencing sleepiness. The author provide a thorough analysis of the literature, which advances knowledge in the field and paves the way for future studies aimed at creating more reliable and efficient detection systems that can lessen the risks related to driving while intoxicated. This comprehensive review highlights the need for ongoing innovation and improvement to handle the difficulties of sleepiness detection in a variety of driving scenarios and environments, in addition to improving our understanding of present approaches.

The crucial problem of driver drowsiness in traffic accidents is explored by P. S. Nandhini et al. [10] They offer a real-time drowsiness detection model utilizing Deep Learning techniques, with a focus on early symptom identification via facial landmark analysis. The work uses Deep Learning algorithms to examine important facial traits related with drowsiness, leveraging the abundant information found in facial expressions. The goal is to identify minor symptoms of sleepiness before they become problematic. Furthermore, an adaptive threshold technique is employed to dynamically modify the detecting system's sensitivity, guaranteeing excellent performance in a variety of driving scenarios and surroundings. In order to set the stage for the creation of a lightweight, real-time sleepiness detection model based on Deep Learning for early symptom recognition, the introduction emphasizes the frequency of accidents due to driver tiredness and the shortcomings of conventional detection approaches. The authors offer insights into current approaches to sleepiness detection by a thorough review of the literature, emphasizing the shortcomings of present techniques and the need for more sophisticated detection systems. The study's findings and conclusions highlight the suggested model's potential to increase the accuracy of drowsiness detection and reduce accidents brought on by fatigued drivers, but they also acknowledge some potential drawbacks, such as the model's accuracy and applicability in a variety of driving scenarios. To sum up, the research conducted by P. S. Nandhini and colleagues has made a noteworthy contribution to the field of sleepiness detection technology. It presents a viable approach to improving road safety and lays the groundwork for further research in this crucial area. To effectively manage the risks associated with drowsy driving and ensure driver safety on the roads, drowsiness detection system innovation and improvement are essential.

In a thorough review of the literature, Ajinkya Rajkar et al.[13] address the pervasive problem of driver fatigue in traffic accidents, emphasizing the critical need for sophisticated detection technologies to reduce related hazards. The survey highlights the frightening numbers and the substantial impact of sleepy driving on road safety, as well as the frequency of incidents that are linked to driver fatigue. The investigation of increasingly complex and automated detection systems has been prompted by the inability of conventional detection techniques, which depend on behavioral characteristics, to reliably identify sleepiness at an early stage, despite heightened awareness. The literature provides insights that highlight the shortcomings of current methods and emphasize the need for cutting-edge technologies, especially those that use Deep Learning algorithms for real-time monitoring. By allowing the study of complex facial data, such as facial landmarks and expressions, deep learning presents a viable path for tiredness detection by identifying subtle markers of fatigue. This makes it possible to identify symptoms early and take proactive measures to stop accidents before they happen. The survey underscores the significance of accommodating heterogeneous driving conditions and surroundings, underscoring the necessity for resilient and adaptable detection systems that can operate dependably in a range of situations. The study's introduction lays the groundwork for the creation of a lightweight, real-time Deep Learning-based sleepiness detection model that aims to address the shortcomings of current techniques by emphasizing early symptom recognition and promptly alerting sleepy drivers. With the use of facial landmark analysis and Deep Learning algorithms, the suggested model presents a viable way to improve road safety and lessen the dangers of sleepy driving. The survey does, however, recognize some possible drawbacks with the suggested model, such as its accuracy and suitability for a variety of driving situations. This emphasizes the necessity for additional study and improvement to guarantee the efficacy and dependability of the detection system in practical situations. To sum up, the literature review conducted by Ajinkya Rajkar et al. offers insightful information about the state of sleepiness detection technology today and establishes the foundation for further research projects targeted at creating more reliable and efficient detection systems to solve this serious safety issue.

Parth Patel et al.[12] address the crucial problem of driver fatigue, which is a major cause of traffic accidents. They suggest a real-time system for detecting fatigue in drivers that makes use of Deep Learning algorithms. The study underlines the drawbacks of conventional detection techniques that rely on behavioral characteristics as well as the significance of early symptom identification in preventing accidents. The goal of the project is to use convolutional neural networks (CNNs), one type of Deep Learning algorithm, to evaluate facial expressions and identify sleepiness in real time. Furthermore, an adaptive threshold technique is used to dynamically identify driver fatigue and generate alarms as necessary. The literature review's conclusions highlight the shortcomings of current strategies and the need for cutting-edge detection technologies like deep learning to provide timely alerts that reduce the risk of driver fatigue-related accidents. The goal of the research is to improve the accuracy of sleepiness detection and lower the number of accidents by promptly alerting sleepy drivers. The suggested approach is noteworthy for achieving an astounding 96% accuracy in real-time video input for detection. Nonetheless, some drawbacks like the model's generalizability and performance under various driving circumstances are recognized, highlighting the necessity for more investigation and improvement to guarantee the efficacy and dependability of the detection system in practical situations. However, several limitations such as the model's generalizability and performance under different driving conditions are acknowledged, emphasizing the need for further research and development to ensure the effectiveness and reliability of the detection system in real-world scenarios. However, several limitations such as the model's generalizability and performance under different driving conditions are acknowledged, emphasizing the need for further research and development to ensure the effectiveness and reliability of the detection system in real-world scenarios.

In a thorough review of the literature, Farshad Farahnakian et al.[5] painstakingly examine the complex issues surrounding the identification of driver fatigue, a significant contributing factor to traffic accidents. Advanced technologies like Deep Learning are desperately needed to enable accurate and timely identification of drowsiness, as traditional methods based on observable behavioral indicators may not be able to identify it in its early phases. The likelihood of traffic accidents is greatly reduced by timely alarms from detection devices, underscoring the critical relevance of early detection in reducing accidents caused by driver sleepiness. Furthermore, convolutional neural networks (CNN), in particular, show great potential for interpreting complex face cues and making real-time predictions of tiredness, which will improve the precision and effectiveness of drowsiness detection systems. Real-time detection systems play a crucial role in promoting road safety by swiftly notifying drivers when they observe signs of exhaustion. This adds an extra layer of protection to assist prevent accidents caused by drivers' distractions. The literature review does, however, also recognize the necessity of more studies targeted at improving Deep Learning models for sleepiness detection. Potential drawbacks would be addressed, and the adaptability of such systems under various driving scenarios and among various demographic groups will be investigated. In summary, the literature review conducted by Farshad Farahnakian et al. not only provides insight into the current state of drowsiness detection technology, but it also establishes the foundation for future developments that will integrate increasingly complex detection systems to further improve road safety.

Md. Tanvir Ahammed Dipu et al. [4] concentrate on the most recent developments in driver drowsiness detection technology, with a special emphasis on the use of Convolutional Neural Networks (CNN) and MobileNet with Single Shot Multibox Detector (SSD) for object detection. The study highlights how important real-time detection technologies are becoming in reducing the dangers of sleepy driving, which is one of the main causes of traffic accidents. The study's conclusions point to a calculated method for using CNN-based MobileNet with SSD architecture for real-time object detection, with a focus on yawning and closed/open eyes in video streams. In comparison to conventional approaches, this novel methodology offers better accuracy and computing efficiency, marking a paradigm shift in the field of sleepiness detection. By combining MobileNet and SSD, it is possible to quickly and accurately detect objects, which improves road safety by allowing notifications to be sent to sleepy drivers. The study emphasizes how crucial it is to conduct ongoing research and development to improve drowsiness detection systems in order to overcome current shortcomings and improve overall performance. With its innovative approach to real-time driver alertness monitoring, the suggested methodology represents a substantial leap in the identification of driver drowsiness. Through the utilization of SSD architecture in conjunction with CNN-based MobileNet, the research makes impressive progress in object identification, especially in recognizing crucial signs of fatigue as yawning and open/closed eyes. By providing alerts to sleepy drivers in real-time, real-time video stream analysis lowers the likelihood of accidents brought on by driver drowsiness. The combination of SSD and MobileNet improves detection accuracy while maintaining computing efficiency, allowing video data to be processed quickly and with low latency. This is essential for real-world applications, as prompt notifications can significantly reduce the risk of mishaps.

Harisudha Kuresan et al.[3] conduct a thorough literature review on the sophisticated application of deep learning techniques for driver drowsiness detection systems. They specifically highlight the combination of Convolutional Neural Networks (CNN) and Transfer Learning. The study emphasizes how important it is for these technologies to detect driver drowsiness in order to improve road safety. The study's findings show that the suggested model works noticeably better than conventional approaches, registering an astounding accuracy rate of more than 87.4% in sleepiness detection. The sophisticated use of CNN for face feature detection and the integration of Transfer Learning and OpenCV for implementation, which collectively guarantee reliable and effective real-time driver monitoring, are credited with this high degree of accuracy. The technique used includes a thorough examination of facial expressions, which are important markers of sleepiness. CNN carefully identifies facial characteristics, and Transfer Learning improves the model's capacity to generalize from previously trained models. OpenCV makes practical implementation easier by monitoring factors like eye closure, yawning, and head motions, enabling real-time processing and alarm systems. Notwithstanding these encouraging findings, the survey also identifies important drawbacks, such as the difficulty presented by human dishonesty, in which drivers may purposefully or inadvertently conceal indicators of fatigue, increasing the danger of an accident. The problem is exacerbated by the fact that current awareness campaigns have little success in reducing crashes brought on by tired drivers. The synopsis of the introduction highlights the pressing need for a deep learning-based system to identify driver drowsiness and goes over the different factors that are taken into account, such as speed, RPM, and facial expressions. The abstract outlines the primary goal of creating a model that successfully identifies tiredness using CNN and Transfer Learning and emphasizes the vital relevance of addressing driver drowsiness, a major cause of traffic accidents worldwide. Overall, the survey offers a comprehensive analysis of recent developments in driver drowsiness detection technologies, emphasizing the potential of deep learning methods to boost detection efficacy and accuracy while recognizing current constraints and the need for continued research to overcome them in order to eventually promote safer driving conditions.

Santosh Kumar Satapathy et al. [8] demonstrate the critical role that machine learning and specifically deep learning algorithms plays in improving road safety. Their research highlights the efficiency of Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNN) in identifying sleepiness; the MLP model achieves an astounding 86% accuracy rate, indicating its potential for real-world use. The suggested method makes use of cutting-edge computer vision algorithms to extract and evaluate face traits that are important indicators of tiredness, like eye closure and yawning. The system reduces the risk of fatigue-related accidents by processing and interpreting real-time video data and sending timely alarms to drivers through the use of deep learning algorithms. The ineffectiveness of early sleepiness detection by traditional methods based on behavioral cues emphasizes the need for cutting-edge technology like deep learning. Since timely notifications from detection systems can greatly lower the probability of traffic accidents, early detection is essential in minimizing accidents caused by driver drowsiness. The study's conclusions show that deep learning approaches have promise for treating driver fatigue, but they also highlight the necessity of ongoing testing and improvement to guarantee practical application. Subsequent research endeavors aim to incorporate supplementary physiological and behavioral indications in order to augment the system's resilience and suitability for diverse driving contexts and demographics. Increasing the system's performance in different lighting scenarios and expanding the variety of facial expressions in the training set are essential steps in developing a solution that is more widely applicable. The suggested methodology demonstrates the promise of deep learning in creating more precise and dependable driver sleepiness detection systems. It combines CNN for face feature identification and Transfer Learning in conjunction with OpenCV for implementation. Setting a standard for future developments in the sector, the work strikes a balance between high precision and computing economy. The survey is an important tool for academics and practitioners who want to create driver assistance systems that are more dependable and efficient. It also advances the larger objective of decreasing traffic accidents and improving driver safety via technological innovation.

Hovannes Kulhandjian et al. [7] combine radar and visual sensors to achieve impressive accuracy rates of over 95% in detecting drowsy drivers. The study emphasizes the crucial need to identify driver fatigue to improve road safety, utilizing the complementary advantages of radar detection and optical surveillance technology. The proposed approach employs a micro-Doppler radar sensor to capture physiological movements and changes in facial characteristics indicative of drowsiness, alongside a camera for visual monitoring. By integrating various sensory inputs with data fusion techniques and deep learning algorithms, the system offers a reliable foundation for real-time detection and alerts. The visual surveillance system examines facial expressions and eye movements, which are important markers of exhaustion, while the radar sensor picks up minute physiological changes that coincide with sleepiness. This dual-sensor approach addresses the limitations of conventional single-sensor methods, which often fail to capture the full spectrum of sleepiness indicators, significantly improving detection accuracy and reliability. The system's deep learning models are trained to decipher intricate patterns from radar and visual data, ensuring a thorough and accurate detection process. The research indicates that integrating information from multiple sensors can overcome the constraints of individual sensor systems, offering a more comprehensive evaluation of the driver's condition. The end result is a drowsy driver detection system that can detect symptoms of fatigue with high accuracy in real-time, enabling prompt remedial actions like automated interventions or driver notifications. The study underscores the technology's practical applications and its potential to significantly reduce the number of accidents caused by tired drivers. The system's high accuracy rate suggests it could be a dependable tool for both personal and commercial vehicles, supporting broader road safety initiatives. The robustness of the detection process is enhanced by the incorporation of radar sensors, which add a layer of biometric analysis immune to visual occlusions and lighting conditions that can affect camera-based systems. The results demonstrate robust performance under various conditions, indicating a high degree of adaptability and resilience.

Chapter 3

Problem Statement

To develop CNN models for driver drowsiness detection that enhance accuracy and efficiency, reduce computational complexity, and improve road safety by reliably identifying signs of driver fatigue.

3.0.1 Problem Description

Detecting driver drowsiness using traditional methods like Electroencephalography (EEG) and Electrocardiography (ECG) poses several challenges, including complexity and limited accuracy, which hinder their widespread adoption. Additionally, these traditional approaches entail significant costs associated with equipment procurement and maintenance. As an alternative, Convolutional Neural Network (CNN) models are proposed for drowsiness detection, offering superior performance and reliability compared to traditional approaches . The challenge is to design and implement CNN models capable of accurately and efficiently detecting drowsiness, while overcoming the complexity and accuracy limitations of traditional EEG and ECG techniques.The objective of this project is to increase accuracy while cutting down on calculation time and complexity. By addressing these challenges and harnessing the power of deep learning, this system will offer improved accuracy and reliability in detecting driver drowsiness, thereby enhancing road safety and reducing the risk of accidents caused by drowsy driving.

3.1 Motivation

Driver drowsiness is a significant factor contributing to road accidents, posing a severe threat to public safety. Traditional methods for detecting drowsiness have proven to be complex, less accurate, and often costly, limiting their widespread adoption. With advancements in machine learning and deep learning, specifically Convolutional Neural Networks (CNNs), there is an opportunity to develop more reliable, efficient, and cost-effective solutions. This project is motivated by the need to enhance driver safety by leveraging CNNs to create an accurate and real-time drowsiness detection system, ultimately reducing the risk of accidents caused by drowsy driving.

3.2 Contributions

- Implementation of various CNN architectures tailored for drowsiness detection, including InceptionV3, EfficientNetB2, MobileNetV2, and a stacked ensemble model.
- Exploration of ensemble techniques to combine the strengths of different models, enhancing overall performance.
- Comprehensive evaluation and comparison of these models, demonstrating the highest accuracy achieved by the stacked ensemble model at 0.9621

3.3 Organization of Thesis

Chapter 1 provides a brief introduction to the problem of driver drowsiness detection, its significance, and the objectives of the project. Chapter 2 presents a comprehensive literature survey, reviewing existing methods and technologies used in drowsiness detection and identifying their limitations. Chapter 3 outlines the problem statement, with Section 3.1 detailing the motivation behind the project, and Section 3.2 highlighting the contributions made by this research. Chapter 4 discusses the proposed methodology, including an in-depth examination of the different CNN architectures considered for the project. Chapter 5 covers the implementation details, including a description of the dataset used, preprocessing steps, and the modifications applied to the models. Chapter 6 contains the results and analysis, presenting the performance outcomes of the implemented models. Finally, Chapter 7 concludes the thesis with a summary of the findings, followed by a section on future work, suggesting directions for further research and improvements.

Chapter 4

Proposed Methodology

4.1 Proposed Workflow

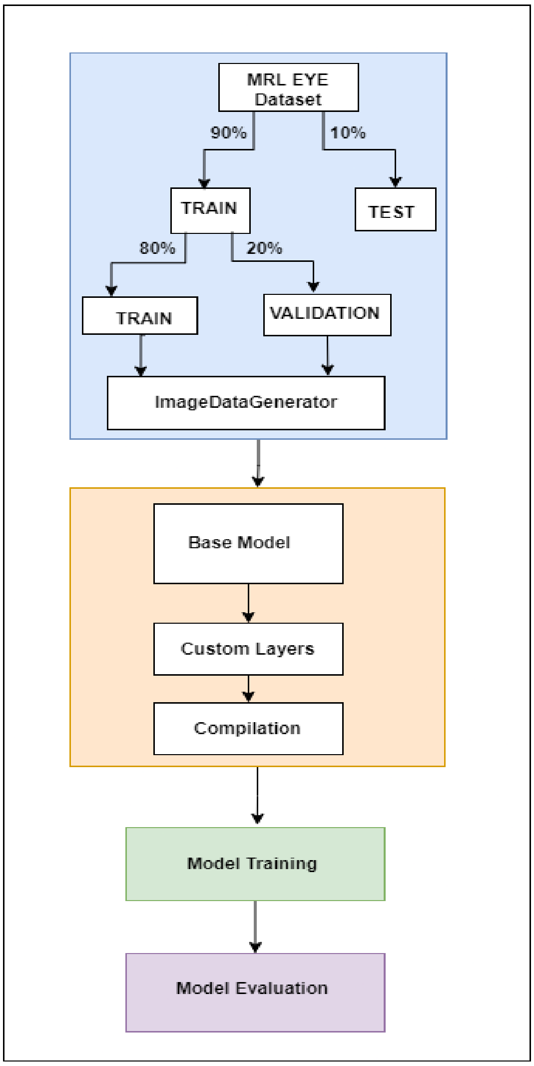

Figure 4.1: Proposed Methodology for Driver Drowsiness Detection

Figure 4.1 shows the methodology which is used in driver drowsiness detection. The MRL Eye Dataset [6] is used in this research in order to create and assess the driver drowsiness detection system. To aid with training and testing of the model, the dataset—which comprises an extensive array of photos capturing eye states—is separated into various subsets. Ten percent of the data is initially set aside for testing the model's performance, while the remaining ninety percent is used for training. The 90% training data is further divided into 80% for real training and 20% for validation in order to provide robust training. This method keeps an eye on the model's performance throughout training and makes necessary adjustments to ensure generalization and avoid overfitting. Several augmentation approaches are used to improve the dataset before feeding it into the Convolutional Neural Network (CNN) model. The classification task then makes use of the pretrained models from the Tensorflow library. These pre-trained models on the ImageNet dataset offer a strong basis for transfer learning, enabling optimization for the particular sleepiness detection job. After being trained to categorize the input images, the CNN model produces a binary response, with 0 denoting alert and 1 denoting sleepy.

4.2 Convolution Neural Network Architectures

4.2.1 InceptionV3 [17]

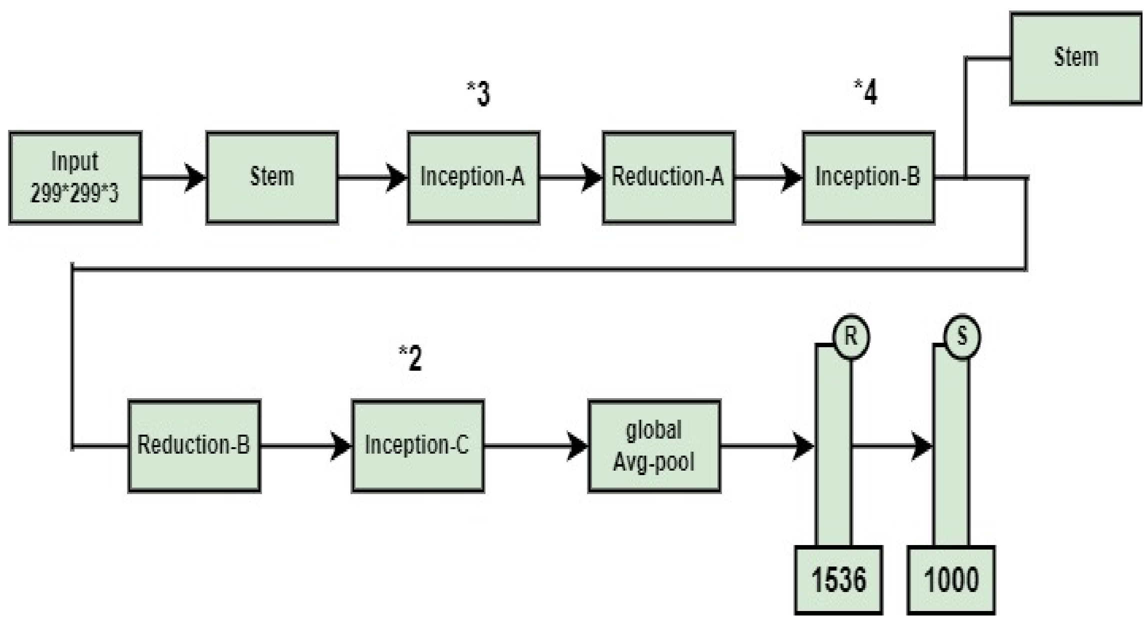

Figure 4.2: InceptionV3 Architecture

Google created the InceptionV3 architecture which is shown in figure 4.2, a convolutional neural network (CNN) with exceptional efficiency for image categorization that strikes a compromise between computational efficiency, depth, and width. The 48-layer architecture makes use of inception modules, which enable the network to gather visual data at various scales and resolutions by performing convolutions of various sizes (1x1, 3x3, and 5x5) within a single module. Factored convolutions, which divide bigger convolutions into smaller ones—for example, dividing a 3x3 convolution into two 1x3 and three 1x1 convolutions—are used by InceptionV3 to improve computational efficiency. Auxiliary classifiers, which are smaller networks added at intermediate layers in the architecture, are also included to provide additional gradient signals during training to speed up and stabilize convergence. In order to control computational complexity and memory needs, grid size reduction techniques, which involve strided convolutions and pooling layers, are deliberately utilized to downsample the spatial dimensions of the input. Each layer's input is normalized by batch normalization, which lowers internal covariate shift and increases training stability and speed.

Inception modules can also use average pooling in addition to convolutions to capture spatial information. In addition, the model's use of label smoothing during training enhances generalization and guards against overfitting.

The InceptionV3 model is pre-trained on the ImageNet dataset, which consists of more than a million images in a thousand classifications. This enables it to pick up a wealth of features that can be adjusted for a variety of applications. A succession of inception and reduction blocks that analyze input at several scales and downsample feature maps comprise the architecture's first stem, which consists of simple convolutions and max-pooling. Fully connected layers for the final classification come last in this arrangement. With the use of auxiliary classifiers, factorized convolutions, multi-scale processing, batch normalization, and other techniques, InceptionV3 may achieve great performance and robustness, which makes it an effective tool for image classification applications.

4.2.2 EfficientNetB2 [18]

Comprising convolutional neural networks, EfficientNetB2 is renowned for achieving the ideal balance between accuracy and efficiency through compound scaling. This architecture, which consists of 237 layers, uses a methodical technique based on particular scaling coefficients to equally scale up the network's width, depth, and resolution. The basic stem of convolutions and max-pooling layers forms the foundation of the architecture. The next set of mobile inverted bottleneck MBConv blocks improves feature extraction while preserving computing efficiency. They include squeeze-and-excitation optimization, depthwise separable convolutions, and residual connections. During training, the model makes use of swish activation functions to enhance gradient flow and convergence. EfficientNetB2 starts with a 3x3 convolution layer and then uses 16 MBConv blocks arranged in seven stages, each with less spatial dimensions and more channels. These phases lead to the final fully linked layer for classification, which is followed by a global average pooling layer. EfficientNetB2 may use learnt features for a variety of applications because it has been pretrained on the ImageNet dataset. EfficientNetB2 can be deployed in resource-constrained environments and still achieve state-of-the-art results in image classification tasks thanks to the design principles of the architecture, which include compound scaling and the use of efficient MBConv blocks.

4.2.3 MobileNetV2 [14]

Building upon the original MobileNet architecture with numerous significant enhancements that improve speed and performance, MobileNetV2 is a highly optimized convolutional neural network architecture specifically made for mobile and resource-constrained environments. Figure 4.3 shows MobileNetV2 architecture. Its fundamental building blocks are inverted residual blocks, which flip the conventional layout of residual blocks by making the shortcut connection the primary path and subjecting the processing path to a number of light operations. A depthwise separable convolution, which factorizes a standard convolution into two separate steps (depthwise convolution for effective spatial feature extraction and pointwise convolution to com-

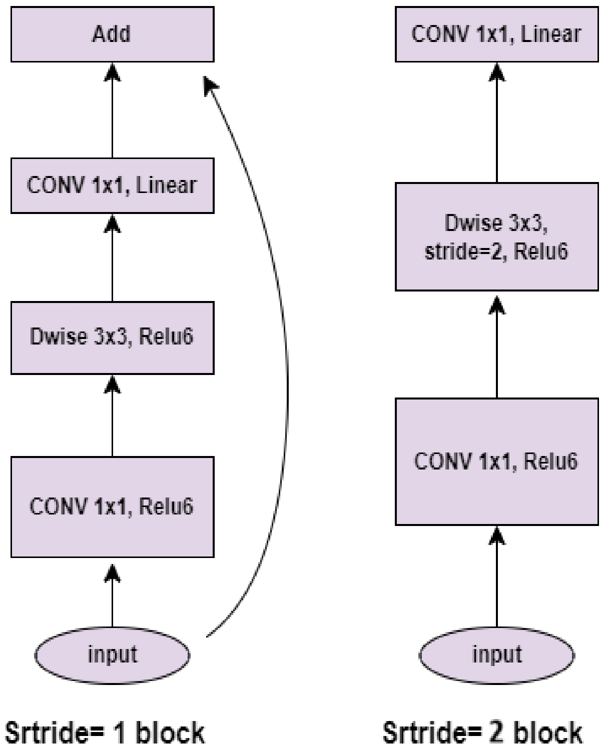

Figure 4.3: MobileNetV2 Architecture

bine the filtered channels), a 1x1 convolution to expand the input channels, and an additional 1x1 convolution to project the output channels back to a lower-dimensional space are some examples of these operations. This architecture preserves important information while enabling effective feature extraction. MobileNetV2 employs two kinds of inverted residual blocks: stride 1 blocks, which downsample the feature maps by a factor of two to reduce computational complexity and enable multi-scale feature representation, and stride 2 blocks, which preserve the spatial dimensions of the feature maps for deeper feature extraction.

By using linear bottlenecks and bottleneck design, the architecture further optimizes. ReLU6 activations are employed in the 1x1 projection layers of the bottleneck design to increase feature representation and add non-linearity. Linear bottlenecks are used in the thin layers (depthwise convolution layers) to eliminate non-linearities and maintain feature integrity while avoiding superfluous complexity. Through "width multipliers," which let users change the number of channels in each layer to scale the model's architecture and govern its size and complexity without compromising the basic design, MobileNetV2 also provides flexibility. Furthermore, the ImageNet dataset is used to pre-train MobileNetV2, offering a solid basis for transfer learning across a range of computer vision applications. MobileNetV2 is a strong yet effective model that can be used in mobile and embedded apps thanks to this pre-training, which helps the model learn a wide range of features that can be tailored to a variety of applications with little fine-tuning.

4.2.4 Stacking Ensemble [1]

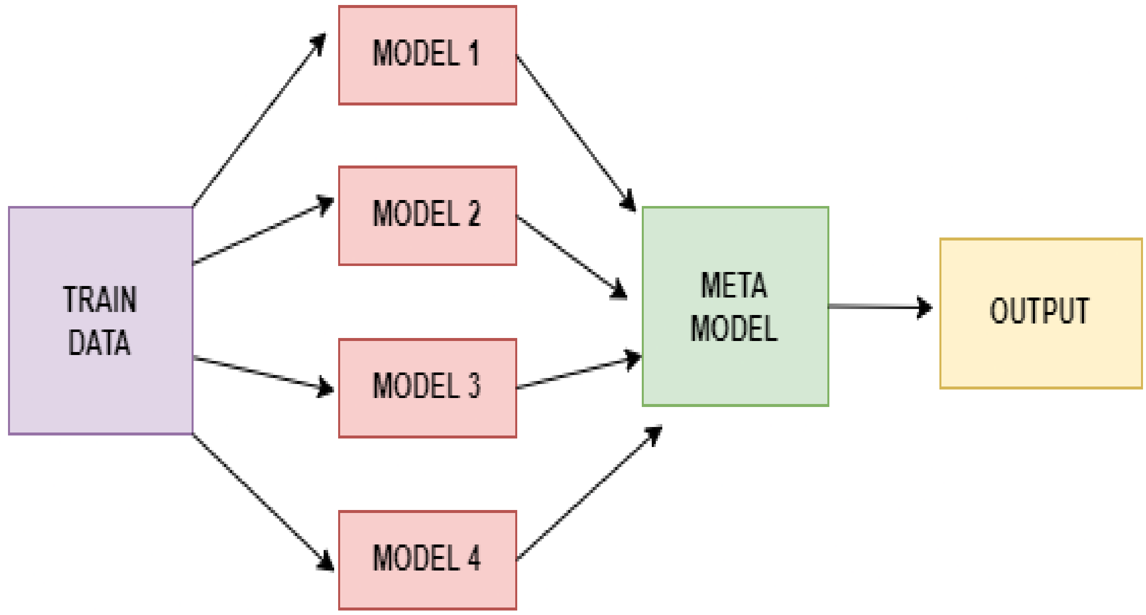

Figure 4.4: Stacking Ensemble Classifier

Stacking ensemble is an advanced machine learning method that combines several base models, or learners, to take use of each model's advantages and improve predicted performance as shown in Figure 4.4. Stacking ensemble is a heterogeneous ensemble approach since it uses a variety of models, in contrast to classic ensemble methods like bagging and boosting, which usually use homogeneous models. The first step in the process is to choose and train a variety of base models, which can be anything from simple algorithms like k-nearest neighbors, decision trees, and support vector machines to more sophisticated ones like neural networks and gradient boosting machines. Following training, these basic models' predictions are gathered and fed into a second-level model called the meta-learner, also referred to as the stacking model.

This meta-learner, which is often a straightforward model such as logistic regression or linear regression, learns how to optimally integrate the predictions of the base models to get an enhanced output in the end, however more sophisticated models may also be employed if needed. In order to ensure that the meta-learner is trained on independent data and prevents memory of base models' predictions, the training data is separated into two parts for the stacking process: one part is used to train the base models, while the other is used to generate predictions that are input for the meta-learner. By repeatedly splitting the data and averaging the base models' predictions over several folds, cross-validation can be used to improve resilience.

By mixing models with various advantages and disadvantages, stacking ensemble enhances prediction performance and increases its ability to capture both linear and non-linear interactions. It does, however, present several difficulties, including the requirement for rigorous validation and hyperparameter tuning, computational intensity, and the crucial selection of base models and meta-learners. Stacking ensembles are useful for complicated machine learning problems because they combine several models through a meta-learner, leveraging the different capabilities of individual models to provide higher accuracy and generalization compared to individual models.

Chapter 5

Implementation Details

5.1 Dataset Used

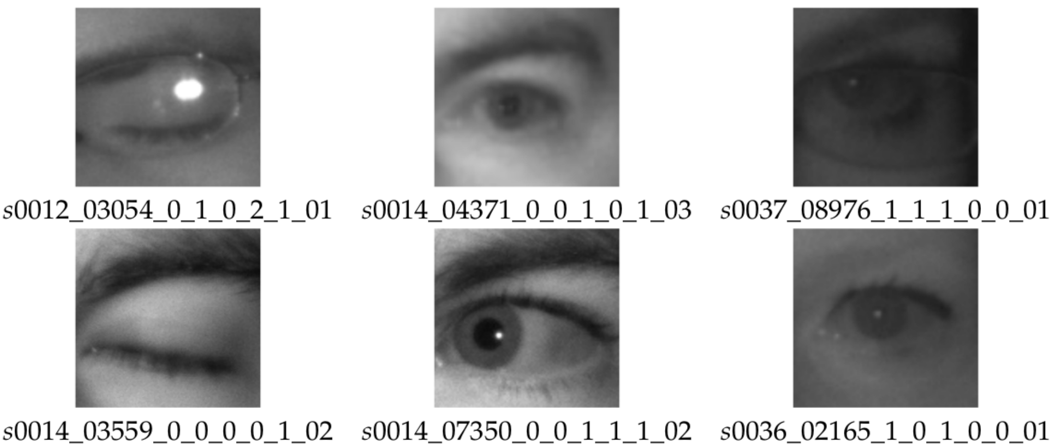

Figure 5.1: Images from MRL Eye Dataset

5.1.1 Dataset Description

Identifying eyes and their components, estimating gaze, and determining the frequency of eye blinking are crucial tasks in computer vision, especially in the domain of analyzing driving behavior. To solve these responsibilities, a significant amount of real-world testing data has been collected in the last few years. To help with this research, a large-scale dataset of images of human eyes called the MRL Eye Dataset [6] was created. This collection includes both low-resolution and high-resolution infrared photos that were taken using various devices and lighting conditions.The MRL Eye Dataset's diversity and size allow it to be used for testing a variety of features and trainable classifiers. Few images from dataset is shown in figure 5.1. It also streamlines algorithm comparison by classifying images into separate groups for testing and training.

5.1.2 Dataset Properties

To guarantee complete data for analysis and training, the dataset has thorough annotations for every image that include a range of properties:

- **Subject ID:**The dataset ensures a broad sample of human subjects with data from 37 individuals, including 4 women and 33 men.

- **Image ID:**The dataset contains 84,898 photos in total, offering a sizable quantity of information for testing and training models.

- **Gender:**Gender-specific analysis is made possible by the annotation of the subject's gender (0 for a man, 1 for a woman) on every image.

- **Glasses:**To comprehend how glasses affect eye detection algorithms, it is essential to know whether the person is wearing spectacles (0 for no, 1 for yes). This information is provided by the dataset.

- **Eye State:**This characteristic, which is crucial for activities like blink detection and sleep monitoring, determines whether the eye is open or closed (0 for closed, 1 for open).

- **Reflections:**Depending on how large they are, reflections in the eyes can be classified as none (0), small (1), or big (2). This annotation facilitates the assessment of algorithm performance in various reflection settings.

- **Lighting Conditions:**Every picture has its lighting settings categorized as either bad (0) or good (1), which serves as a foundation for evaluating how well algorithms hold up under different lighting situations.

- **Sensor ID:**Three distinct sensors, each with a different resolution, were used to take the images:

  - **RealSense (ID 01)** with a resolution of 640 x 480.

– **IDS (ID 02)** with a resolution of 1280 x 1024.

– **Aptina (ID 03)** with a resolution of 752 x 480.

5.2 Data Preprocessing

5.2.1 Data Splitting

The dataset was strategically divided based on the eye state annotation to maximize the efficiency and robustness of the model's training and evaluation processes:

- **Initial Split:** The entire dataset was divided into two main sets: 90% for training and 10% for testing.
- **Training and Validation Split:** The 90% of the data allocated for training was further divided into two subsets: 80% for training and 20% for validation.

By using this technique, the model is tested and validated on different, unknown data after being trained on a significant amount of the data, giving a realistic evaluation of its performance.

5.2.2 Rescaling

All of the photos were rescaled by a factor of 1/255 to normalize the pixel values to the range [0, 1], which facilitated faster convergence throughout the model training process. One typical preprocessing technique for picture data is normalization, which helps to speed up and stabilize the training process.

5.2.3 Data Augmentation

Various data augmentation techniques were employed to enhance the model's generalization ability and robustness. By artificially increasing the diversity of the training data, these augmentations help the model to perform better on new, unseen data. The following augmentations were applied:

- **Rotation Range:** Images were randomly rotated within a range of 20 degrees to make the model invariant to the orientation of the eye images.

- **Shear Range:** Shearing transformations with an intensity of 0.2 were applied, introducing slight distortions to simulate variations in perspective.

- **Zoom Range:** Images were randomly zoomed in and out by a factor of 0.2 to allow the model to handle variations in the size of the eye region. Width Shift Range: Horizontal shifts of 20% of the image width were applied to account for horizontal movements of the eyes.

- **Height Shift Range:** Vertical shifts of 20% of the image height were applied to simulate vertical movements.

These augmentations collectively improve the model's ability to generalize across different scenarios and lighting conditions.

5.2.4 Input Size

Every image was adjusted to have the same 80x80 size. Consistency in input dimensions is guaranteed by this scaling, which is essential for the convolutional neural network (CNN) model to process images efficiently. When training a model, uniform picture dimensions aid in preserving both computational efficiency and structural integrity.

5.2.5 Batch Generation

In order to efficiently allocate computational resources for the training, validation, and testing stages, batches of 8 images were created. By updating model weights more frequently, batch processing improves training efficiency in addition to assisting with data fit into memory.

5.3 Model Architectures and Modifications

5.3.1 Standalone Pre-trained Models

We used three standalone pre-trained models: InceptionV3, EfficientNetB2, and MobileNetV2. For each model, the following modifications and configurations were applied:

- **Input Layer:** Modified to accept input images of size 80x80x3.
- **Top Layers:** Removed the fully connected top layers to use the model as a feature extractor.
- **Additional Layers:** Flatten layer to convert the 2D feature maps into 1D feature vectors.
- Dense layer with 64 units and activation function.
- Dropout layer with a dropout rate of 0.2 to prevent overfitting.
- Final Dense layer with 1 units and sigmoid activation for binary classification.
- **Training:** All layers of the pre-trained model were frozen to retain their pre-trained weights.

5.3.2 Stacked Ensemble Model

In addition to the standalone models, we implemented a stacked ensemble model combining InceptionV3, EfficientNetB2, and MobileNetV2:

- **Base Models:** InceptionV3, EfficientNetB2, and MobileNetV2 models were used as base models.
- Each base model was configured with its pre-trained weights and without the top fully connected layers.
- **GlobalAveragePooling2D** was applied to the output of each base model to obtain feature vectors.

- **Concatenation:** The feature vectors from all three base models were concatenated to form a combined feature representation.
- BatchNormalization layer to normalize the concatenated features.
- Dense layer with 256 units and ReLU activation.
- Dropout layer with a dropout rate of 0.2.
- BatchNormalization layer.
- Dense layer with 128 units and activation function.
- Final Dense layer with 1 unit and sigmoid activation for binary classification.

5.4 Training:

To maintain uniformity and enable a comprehensive comparison study, I used the same set of parameters and hyperparameters for all models, including both standalone and ensemble techniques. Because of its effectiveness in improving convergence and overall model performance, the Adam optimizer was chosen. Regarding loss functions, binary cross-entropy was employed, reflecting its binary classification nature. The main statistic used to assess the effectiveness of the model was accuracy, which expresses the percentage of properly classified samples relative to the total number of samples and is a direct assessment of the efficacy of the model. All model layers were subjected to a uniform dropout rate of 0.2 in order to improve generalization and reduce overfitting. The model's resilience was increased by ensuring that they did not rely unduly on any one feature thanks to this dropout rate. Ten training epochs were used for each model, which was a good compromise between allowing for sufficient learning and avoiding overfitting. A batch size of eight was used during the training procedure to guarantee that the models could process data well and maintain good performance.

Chapter 6

Results And Analysis

The main metric utilized to assess our models' performance is accuracy. In classification problems, accuracy is a commonly used and basic indicator that indicates how well the model's predictions match the true labels. It is defined as the ratio of correctly predicted samples to the total number of samples, expressed mathematically as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

where:

- **True Positives (TP):** The number of samples that were correctly predicted as positive.
- **True Negatives (TN):** The number of samples that were correctly predicted as negative.
- **False Positives (FP):** The number of samples that were incorrectly predicted as positive (also known as Type I errors).
- **False Negatives (FN):** The number of samples that were incorrectly predicted as negative (also known as Type II errors).

6.1 Different Activation Function Used

6.1.1 ReLU

Rectified Linear Unit, or ReLU for short, is a well-liked activation function seen in deep learning models. The definition of the ReLU function is f(x)=max(0,x),

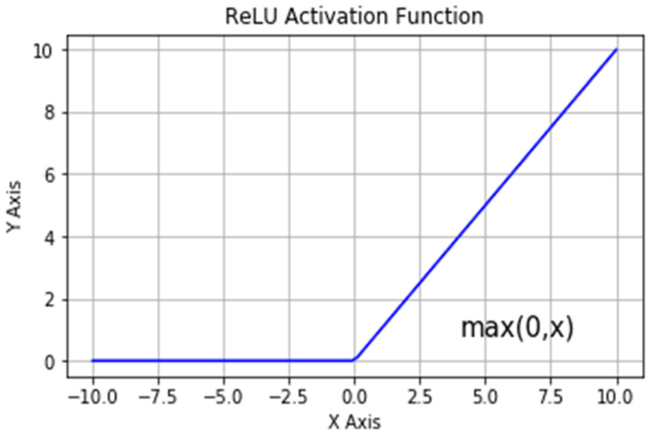

Figure 6.1: ReLU

which indicates that if the input is positive, it outputs the value directly; if not, it outputs zero. The simplicity and efficacy of this non-linear function have led to its widespread adoption. By addressing the vanishing gradient issue that may arise with sigmoid or tanh activation functions, ReLU makes deep neural network training quicker and more effective. ReLU contributes to the maintenance of the positive gradient by allowing only positive values to pass through, which makes sure that the network's weights are updated more successfully during backpropagation.ReLU also adds sparsity to the network by deactivating neurons that don't add anything to the prediction. This can improve generalization and reduce overfitting, both of which can improve the performance of the model. ReLU has drawbacks despite its benefits, such as the "dying ReLU" problem, which occurs when neurons cease to function and cease learning if they regularly produce zero. To address these problems, a number of ReLU modifications, including Leaky ReLU and Parametric ReLU, have been put forth.

6.1.2 Leaky ReLU

An activation function called Leaky ReLU (Leaky Rectified Linear Unit) is employed in neural networks to overcome the shortcomings of the regular ReLU (Rectified Linear Unit) function, namely the "dying ReLU" issue. As a result of the typical ReLU's conversion of any negative input to zero, neurons may go dormant during training if they repeatedly produce zero. When the input is negative, leaky ReLU alters the ReLU function by permitting a tiny, non-zero gradient.

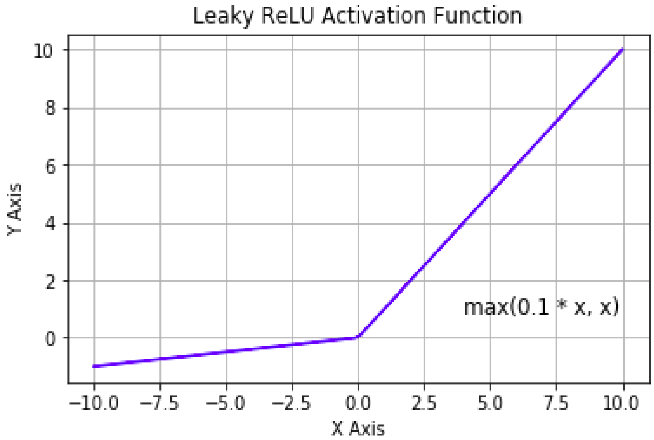

Figure 6.2: Leaky ReLU

Mathematically, Leaky ReLU is defined as:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$$

where α is a small positive constant, typically set to a value such as 0.01. This modification ensures that the gradient is not zero for negative inputs, enabling backpropagation to continue updating the weights of the network even for neurons that receive negative inputs.

Like the normal ReLU, the leaky ReLU maintains its computing efficiency and its capacity to alleviate the vanishing gradient problem. It does, however, outperform ReLU in that it guarantees some degree of neuronal activity, which can result in enhanced training performance and quicker convergence. This is especially helpful for deep neural networks, where it's critical to maintain an efficient gradient flow. Furthermore, by making the parameter α learnable, variations such as Parametric ReLU (PReLU) expand on this idea by enabling the model to adaptively identify the ideal slope for negative inputs during training.

6.1.3 SELU

An activation function called SELU (Scaled Exponential Linear Unit) seeks to promote self-normalization while addressing several frequent problems in deep neural networks, such as vanishing and exploding gradients. The SELU function is defined as:

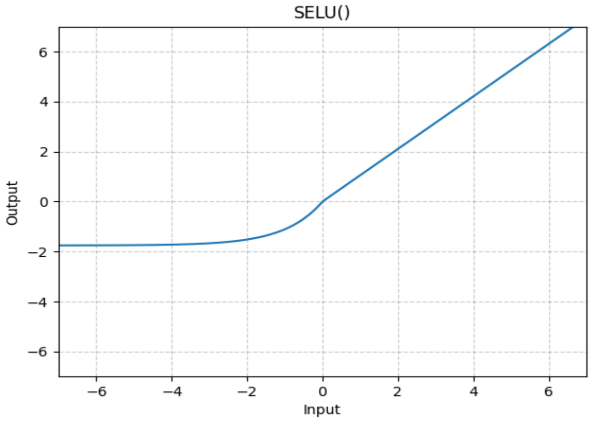

Figure 6.3: SELU

$$f(x) = \lambda \times \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leq 0 \end{cases}$$

where λ and α are constants, with typical values of $\lambda = 1.0507$ and $\alpha = 1.6733$.

Because of a feature of SELU called "self-normalization," each layer's output will maintain a mean of 0 and a standard deviation of 1, provided that the inputs are likewise normalized. This characteristic can aid in reducing the vanishing and exploding gradient issues that deep networks frequently face, enabling more reliable and efficient training.

Furthermore, on a variety of deep learning tasks, SELU has been demonstrated to perform better than other activation functions, including ReLU and its variations, especially in architectures with several layers. It's crucial to remember that SELU could not always perform better and that it must to be empirically verified on particular tasks.

6.1.4 ReLU6

ReLU6 (Rectified Linear Unit 6) is a variant of the ReLU activation function designed to cap the activation at a maximum value of 6. This can help prevent issues with large activations and improve the stability of the network. The ReLU6 function is defined as:

$$f(x) = \min(\max(0, x), 6)$$

or equivalently:

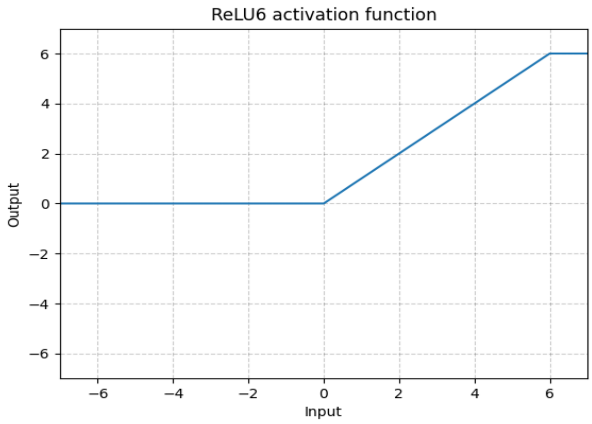

Figure 6.4: ReLU

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } 0 < x < 6 \\ 6 & \text{if } x \geq 6 \end{cases}$$

ReLU6 has been particularly popular in mobile and low-power applications due to its bounded nature, which can make the implementation more efficient and help prevent the model from producing extremely large activations that could cause numerical instability. It was introduced in the context of the MobileNets architecture, which aims to create efficient neural networks for mobile and embedded vision applications.

The key advantages of ReLU6 include:

Bounded Activation: Prevents the activation values from becoming too large, which can help with numerical stability.

Simplicity: Maintains the simplicity and non-linearity characteristics of the standard ReLU function.

Efficiency: Particularly useful in scenarios with limited computational resources, such as mobile and embedded devices.

6.1.5 SiLU

The SiLU (Sigmoid Linear Unit) activation function, also known as the Swish function, is defined as:

$$f(x) = x \cdot \sigma(x)$$

where $\sigma(x)$ is the sigmoid function:

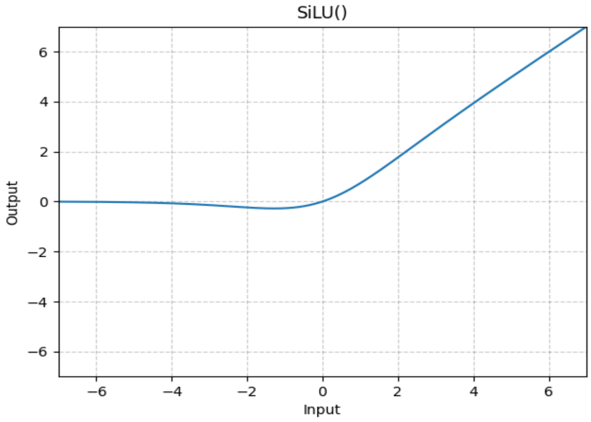

Figure 6.5: SiLU

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The SiLU function yields a smooth, non-monotonic curve by combining the characteristics of the sigmoid and linear functions. By permitting modest negative values during training and avoiding the zero gradient issue that is frequently linked to ReLU, this can help neural networks function better.

The key advantages of SiLU include:

Smoothness: SiLU is smooth and differentiable everywhere, which can aid gradient-based optimization, in contrast to ReLU, which is piecewise linear.

Non-monotonicity: SiLU's non-monotonicity can promote richer representations and improved information flow.

Empirical Performance: On a variety of tasks, SiLU has been demonstrated to perform well in practice, frequently outperforming ReLU and other activation functions.

6.1.6 GeLU

The GeLU (Gaussian Error Linear Unit) activation function is defined as:

$$\text{GELU}(x) = x \cdot \Phi(x)$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution:

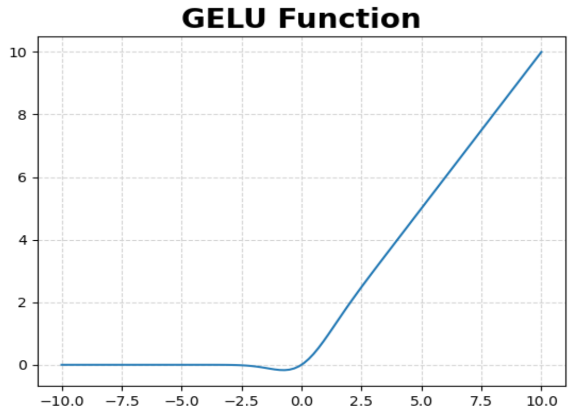

Figure 6.6: GeLU

$$\Phi(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

An approximation often used in practice is:

$$\text{GELU}(x) \approx 0.5 \cdot x \left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715 \cdot x^3\right)\right)\right)$$

Because the GeLU activation function uses a probabilistic gating function to scale the input x according to its value, it can be thought of as a smoother variant of the ReLU activation function. As opposed to ReLU's strict thresholding, this enables values close to zero to be scaled down gently.

The key advantages of GeLU include:

Smoothness: GeLU is smooth and differentiable, which helps in gradient-based optimization.

Probabilistic Interpretation: The probabilistic nature of the function allows it to retain small negative values, which can help in learning richer representations.

Empirical Performance: GeLU has been shown to perform well in various deep learning tasks and is used in state-of-the-art models such as BERT.

6.2 Results Of Different Models

6.2.1 InceptionV3

Below are the training(table 6.1), validation(table 6.2) and test(table 6.3) results.

Activation Function	**Training Accuracy**	**Training Loss**
RELU	0.9568	0.1161
Leaky_ReLU	0.9495	0.1320
selu	0.9567	0.1147
Relu6	0.9382	0.1564
silu	0.9411	0.1515
gelu	0.9410	0.1525

Table 6.1: InceptionV3's Training Accuracy and Training Loss

Activation Function	**Val Accuracy**	**Val Loss**
RELU	0.9100	0.2350
Leaky_ReLU	0.9117	0.2233
selu	0.9051	0.2527
Relu6	0.9034	0.2343
silu	0.9012	0.2456
gelu	0.9020	0.2208

Table 6.2: InceptionV3's Validation Accuracy and Validation Loss

Activation Function	**Test Accuracy**	**Test Loss**
RELU	0.9488	0.1327
Leaky_ReLU	0.9542	0.1278
selu	0.9577	0.1211
Relu6	0.9488	0.1327
silu	0.9456	0.1452
gelu	0.9467	0.1403

Table 6.3: InceptionV3's Test Accuracy and Test Loss

6.2.2 EfficientNetB2

Below are the training(table 6.4), validation(table 6.5) and test(table 6.6) results.

Activation Function	Training Accuracy	Training Loss
RELU	0.6434	0.4287
Leaky_ReLU	0.6523	0.4312
selu	0.6513	0.4211
Relu6	0.6612	0.3825
silu	0.6264	0.4633
gelu	0.6234	0.4473

Table 6.4: EfficientNetB2's Training Accuracy and Training Loss

Activation Function	Val Accuracy	Val Loss
RELU	0.6123	0.4322
Leaky_ReLU	0.6212	0.4421
selu	0.6624	0.4623
Relu6	0.6223	0.3712
silu	0.6032	0.3823
gelu	0.6112	0.4208

Table 6.5: EfficientNetB2's Validation Accuracy and Validation Loss

Activation Function	Test Accuracy	Test Loss
RELU	0.6231	0.4623
Leaky_ReLU	0.6543	0.4234
selu	0.6312	0.4211
Relu6	0.6253	0.3921
silu	0.6242	0.4532
gelu	0.6246	0.4112

Table 6.6: EfficientNetB2's Test Accuracy and Test Loss

6.2.3 MobileNetV2

Below are the training(table 6.7), validation(table 6.8) and test(table 6.9) results.

Activation Function	Training Accuracy	Training Loss
RELU	0.9425	0.1431
Leaky_ReLU	0.9421	0.1340
selu	0.9423	0.1237
Relu6	0.9354	0.1544
silu	0.9421	0.1235
gelu	0.9354	0.1465

Table 6.7: MobileNetV2's Training Accuracy and Training Loss

Activation Function	Val Accuracy	Val Loss
RELU	0.9160	0.2270
Leaky_ReLU	0.9187	0.2433
selu	0.9081	0.2647
Relu6	0.9054	0.2753
silu	0.9022	0.2566
gelu	0.9060	0.2278

Table 6.8: MobileNetV2's Validation Accuracy and Validation Loss

Activation Function	Test Accuracy	Test Loss
RELU	0.9458	0.1427
Leaky_ReLU	0.9642	0.1268
selu	0.9527	0.1241
Relu6	0.9418	0.1537
silu	0.9386	0.1322
gelu	0.9427	0.1543

Table 6.9: MobileNetV2's Test Accuracy and Test Loss

6.2.4 Stacked Ensemble Model

Below are the training(table 6.10), validation(table 6.11) and test(table 6.12) results.

Activation Function	Training Accuracy	Training Loss
RELU	0.9513	0.1676
Leaky_ReLU	0.9558	0.1218
selu	0.9405	0.1556
ReLU6	0.9408	0.1609
silu	0.9323	0.1439
gelu	0.9421	0.1643

Table 6.10: Ensemble Model's Training Accuracy and Training Loss

Activation Function	Val Accuracy	Val Loss
RELU	0.9289	0.2206
Leaky_ReLU	0.9154	0.2853
selu	0.9340	0.2544
ReLU6	0.9241	0.2578
silu	0.9112	0.2524
gelu	0.9220	0.2743

Table 6.11: Ensemble Model's Validation Accuracy and Validation Loss

Activation Function	Test Accuracy	Test Loss
RELU	0.9557	0.1992
Leaky_ReLU	0.9510	0.1494
selu	0.9621	0.1395
ReLU6	0.9505	0.1585
silu	0.9494	0.1432
gelu	0.9581	0.1893

Table 6.12: Ensemble Model's Test Accuracy and Test Loss

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The project's completion signifies the successful integration of state-of-the-art methodologies for driver drowsiness detection. Through the utilization of advanced machine learning techniques, particularly Convolutional Neural Networks (CNNs), the project has demonstrated the transformative potential of both individual and ensemble models in improving performance. The highest accuracy obtained using InceptionV3 was 0.9577%, 0.6543% using EfficientNetB2, and 0.9527% using MobileNetV2. These results underscore the effectiveness of CNNs in accurately identifying signs of drowsiness, proving their utility in this critical application. The exploration of ensemble learning approaches further validated the effectiveness of combining diverse models, culminating in a remarkable accuracy of 0.9621% with the stacked ensemble model.

Ensemble techniques have emerged as a powerful tool for addressing the limitations of standalone models, providing increased precision, dependability, and resilience in detecting subtle signs of driver fatigue. By merging the strengths of various models, the project has achieved a level of robustness and reliability that is crucial for real-world applications. The extensive research conducted on these ensemble approaches has yielded promising results, highlighting their potential to significantly enhance road safety by mitigating the risks associated with driver drowsiness. These advancements not only demonstrate the project's success but also pave the way for future innovations in driver assistance systems, showcasing the substantial benefits of leveraging advanced machine learning and data fusion techniques.

7.2 Future Work

With the project's present phase coming to an end, it creates opportunities for future research and development focused at improving the drowsiness detection system's capabilities. Considering the rapidly changing field of artificial intelligence and driver safety technologies, a number of areas jump out as viable directions for more investigation and improvement. The following outlines these key areas of focus for future work:

- **Advanced Model Architectures** Investigating more complex CNN designs, including recurrent neural networks (RNNs) or attention mechanisms, can provide better temporal modeling and context-aware sleepiness detection capabilities.

- **Adaptive Learning Systems** The adaptability and responsiveness of the model can be improved by using adaptive learning systems that dynamically modify model parameters and thresholds in response to real-time feedback and driver behavior patterns.

- **Edge Computing Integration** Low-latency inference and real-time decision-making can be made possible by integrating the sleepiness detection model with edge computing platforms or onboard vehicle technologies. This improves the system's usefulness in realistic driving circumstances.

- **User-Centric Design** To guarantee that the sleepiness detection system is user-friendly, non-intrusive, and efficiently notifies the driver of alerts and cautions in a timely manner, user-centric design concepts and usability testing are incorporated.

- **Longitudinal Studies** are on longitudinal research to evaluate the sleepiness detection system's long-term efficacy and effects on driving behavior, preventing accidents, and general road safety.

Bibliography

[1] Stamatios-Aggelos Alexandropoulos, Christos Aridas, Sotiris Kotsiantis, and Michael Vrahatis. *Stacking Strong Ensembles of Classifiers*, pages 545–556. 05 2019.

[2] Saad Arif, Saba Munawar, and Hashim Ali. Driving drowsiness detection using spectral signatures of eeg-based neurophysiology. 14:1153268, 03 2023.

[3] Sasi Preetham Ch, Shanmanth Guduru, Vinay Ronagala, Harisudha Kuresan, and Samiappan Dhanalakshmi. Automatic system for driver drowsiness detection system using deep learning. In *2023 International Conference for Advancement in Technology (ICONAT)*, pages 1–4, 2023.

[4] Md.Tanvir Dipu, Syeda Hossain, Yeasir Arafat, and Fatama Rafiq. Real-time driver drowsiness detection using deep learning. *International Journal of Advanced Computer Science and Applications*, 12:844–850, 08 2021.

[5] Farshad Farahnakian, Janika Leoste, and Fahimeh Farahnakian. Driver drowsiness detection using deep convolutional neural network. In *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–6, 2021.

[6] R. Fusek. Pupil localization using geodesic distance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11241 LNCS:433–444, 2018.

[7] Hovannes Kulhandjian, Nicolas Martinez, and Michel Kulhandjian. Drowsy driver detection using deep learning and multi-sensor data fusion. In *2022 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–6, 2022.

[8] Santosh Kumar Satapathy, Hetarthi Mori, Khushi Naik, and Nitin Singh Rajput. Automated drowsiness detection for driver safety: A deep learning-based approach. In *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, pages 774–779, 2023.

[9] Elena Magán, M. Paz Sesmero, Juan M. Alonso-Weber, and Araceli Sanchis. Driver drowsiness detection by applying deep learning techniques to sequences of images. *Applied Sciences*, 2022.

[10] P.S. Nandhini, S. Kuppuswami, S. Malliga, P. Srinath, and P. Veeramanikandan. Driver drowsiness detection using deep learning. In *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1031–1036, 2022.

[11] National Safety Council. Drivers are falling asleep behind the wheel, 2020. Accessed: 2024-05-20.

[12] Parth P. Patel, Chirag L. Pavesha, Santoshi S. Sabat, and Shraddha S. More. Deep learning based driver drowsiness detection. In *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 380–386, 2022.

[13] Ajinkya Rajkar, Nilima Kulkarni, and Aniket Raut. *Driver Drowsiness Detection Using Deep Learning*, pages 73–82. 07 2021.

[14] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[15] Yeresime Suresh, Rashi Khandelwal, Matam Nikitha, Mohammed Fayaz, and Vinaya Soudhri. Driver drowsiness detection using deep learning. In *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, pages 1526–1531, 2021.

[16] M Suriya, K Preethie, M Amitha, M Sumithra, I Sumaiya Fathima, and M Vishnu. An efficient driver drowsiness detection using deep learning. In *2023*

9th International Conference on Advanced Computing and Communication Systems (ICACCS), volume 1, pages 2036–2041, 2023.

[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[18] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. 05 2019.

[19] Praveen Tumuluru, S.Sai Kumar, N. Sunanda, Jaswanth Sai Koduri, Teegela Ayyappa, and Kilaru Balasankar. Sddd: Stacked ensemble model for driver drowsiness detection. In *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 1374–1380, 2023.