Strathmore UNIVERSITY

SU+ @ Strathmore University Library

Electronic Theses and Dissertations

2024

Drowsiness detection system during driving using IoT and Machine Learning.

Somo, Abubakar Mohamed
School of Computing and Engineering Sciences
Strathmore University

Recommended Citation

Somo, A. M. (2024). *Drowsiness detection system during driving using IoT and Machine Learning* [Strathmore University]. http://hdl.handle.net/11071/15649

Follow this and additional works at: http://hdl.handle.net/11071/15649

This work is availed for free and open access by Strathmore University Library.
It has been accepted for digital distribution by an authorized administrator of SU+ @Strathmore University.
For more information, please contact library@strathmore.edu

Drowsiness Detection System During Driving Using IoT and Machine Learning

Abubakar Mohamed Somo

111335

A Research Proposal submitted to the School of Computing Engineering, Strathmore University, for the Award of a master's degree in information technology

STRATHMORE UNIVERSITY

NAIROBI, KENYA

August 2023

DECLARATION

I declare that this work has not been submitted or previously submitted and approved in whole or in partial for the award of a degree by this university or any other university. To my knowledge and belief, this dissertation contains no material previously published or written by another person except where due reference is made in the thesis.

Student

Abubakar Mohamed Somo

111335

Signature ……………………………. Date ………………………………………

Supervisor

Dr. Dickson Odhiambo

Lecturer School of Computing and Engineering Sciences

Faculty of Information Technology

Signature ……………………………. Date 04 April, 2023 ………………………………………

ACKNOWLEDGEMENTS

I wish to thank God for giving me the strength and guidance to write this thesis proposal.

I would like to express my deepest gratitude to my supervisor, Dr. Dickson Owour, for their invaluable guidance, encouragement, and unwavering support throughout the journey of completing this thesis. Their expertise, patience, and constructive feedback have been instrumental in shaping the direction and quality of this work.

I am also indebted to Solomon Itotia, whose insightful suggestions and scholarly input have enriched the content of this thesis.

My sincere appreciation goes to my family for their unconditional love, understanding, and encouragement. Their unwavering belief in my abilities has been a constant source of motivation.

I extend my heartfelt thanks to my friends and colleagues for their inspiration, camaraderie, and intellectual exchange, which have enriched my academic experience.

Finally, I am grateful to all the participants who generously contributed their time and insights to this research.

This work would not have been possible without the support and encouragement of all those mentioned above. Thank you.

ABSTRACT

The interest in implementing drowsiness detection systems through the integration of IoT and Machine Learning, especially in the automotive and transportation sector is growing significantly. By utilizing this technology, it becomes possible to monitor and identify instances of driver drowsiness, addressing safety concerns related to fatigue related accidents. However, the widespread adoption and application of these drowsiness detection systems encounters some challenges such as poor telecommunication for network connectivity for IoT devices and ensuring efficient resource utilization within the constraints of Machine Learning. These are the main challenges faced by drowsiness detection systems during driving. This study designs and implements an efficient drowsiness detection system that utilizes Machine Learning and IoT technologies. The approach will involve the deployment of an IoT connected sensor, which is a camera within the vehicle's environment. This sensor will collect real-time data on the driver's eye movements. This raw data is then preprocessed to extract the relevant features and then processed information will be fed into the Machine Learning model. This model, which is optimized for low-resource environments will be able to perform real time drowsiness classification. Our model will employ CV2, KNN and Dlib algorithms independently. The purpose of implementing these distinct machine learning algorithms is to conduct a comprehensive assessment and comparison of their performance. By doing so, we will be able to determine which algorithm yields the best results in terms of accuracy, thus enabling us to make an informed decision. The implemented solution will aim to enhance transparency and consistency in the acquisition of drowsiness related data. This initiative will make things easier for drivers and demonstrate how we can use IoT and Machine Learning technologies to solve problems around detecting drowsiness. By using both hardware and software, the system will show how we can use IoT concepts to solve common problems in drowsiness detection. The hardware we're using includes a computer camera as the sensors, and we'll also use the OpenCV framework libraries to train the machine learning model. The collected data associated with the drowsiness levels will then be transmitted to a central server for real time analysis. The data will undergo thorough processing and assessment to identify patterns of drowsiness instances. Furthermore, a User-friendly python interface will be developed to provide clients with visual insights into the detected drowsiness instances.

Keywords – *Internet of things (IoT)*.

VT OMNES VNVM SINT

Table of Contents

VT OMNES VNVM SINT

VT OMNES VNVM SINT

List of Figures

ABBREVIATIONS

CNN – Convolutional Neural Networks

DAC – Driver Alert Control

EAR – Eye Aspect Ratio

EEG – electroencephalogram

GSR – Galvanic Skin Response

HRV – Heart Rate Variability

IoT – Internet of Things

MAR – Mouth Aspect Ratio

MLP – Multi-layer Perceptron

RNN – Recurrent Neural Networks

RRV – Respiratory Rtae Variability

RTA – Road Traffic Accidents

SDK – Software Development Kit

SDLP – Standard Deviation of Lane Positioning

STFT - Short-Time Fourier Transform

UML – Unified Model Language

WHO – World Health Organization

DEDICATION

I dedicate this Thesis to the cherished memory of my late father, Mohamed Abu Somo, whose enduring wisdom, unwavering support, and profound love continue to inspire me every day. Though you are no longer with us, your spirit lives on in the values you instilled and the lessons you taught. This achievement is a tribute to your legacy.

To my beloved mother, Zulfa Ali Mohamed, whose boundless strength, selflessness, and encouragement have been my guiding light. Your unwavering belief in my dreams has been the cornerstone of my journey. Thank you for your endless sacrifices and unwavering love.

I also dedicate this work to my grandparents, Zainab Omar and Shekue Athman Sadiq, whose warmth, kindness, and wisdom have shaped my character and fueled my aspirations. Their unconditional love and unwavering support have been a source of strength throughout my life's endeavors.

May the love and lessons imparted by my father, mother, and grandparents continue to illuminate my path and inspire future generation.

CHAPTER 1: INTRODUCTION

1.1 Background of the Study

Driver drowsiness is defined as the state in which a person operating a motor vehicle is too fatigued or sleepy to stay alert, making the driver to be less aware of their surroundings, slow response time and microsleeps (Nauto, 2022). During driving, these symptoms pose a significant risk as they greatly increase the chances of drivers failing to notice exits or traffic signs, drifting into adjacent lanes, or potentially colliding with other vehicles, resulting in an accident (Alajlan & Ibrahim, 2023). According to the National safety and transport authority, the state agency and foundation of traffic safety in Kenya reported that drowsiness causes more accidents than drunk driving during a study conducted by AA Kenya head of Business Development Timothy Keli. He stated that between January and November 2020, 3,123 people were killed in road accidents. Keli added that most accident victims were pedestrians (Mboga, 2020). In recent times, many researchers have proposed systems to be installed in cars with the aim of detecting drowsiness. This initiative is motivated by the urgent necessity to mitigate the frequency of traffic accidents that are caused by driver drowsiness. These researchers have tried different methodologies to detect drivers drowsy state through checking how drivers behave physically and checking their body response (Hussein et al., 2021).

According to Green (2023), Globally, Internet of Things (IoT) usage has increased significantly in recent years, with major corporations such as Cisco, Dexcom, Alarm, Impinj and Intel investing heavily in the technology. These endeavors have resulted in growing benefits for both companies and individuals in terms of their investments. The internet of things (IoT) refers to a collection of devices that can sense data from the environment and share information between devices to achieve smart recognition, location, tracking, monitoring, and management (Patel et al., 2016). Several studies that integrate machine learning models with IoT devices have reported that one of the major challenges is training the models, as machine learning models have a complex architecture and are usually heavyweight. Hence, this creates a challenge when implementing IoT devices like Arduinos and Raspberry Pi's, which are frequently used in detecting drowsiness. These devices have restrictions such as limited memory and modest processing power (Alajlan & Ibrahim, 2023). To address these challenges, a new emerging technology called Machine Learning (ML) has emerged, making it possible to overcome the difficulties of using ML with IoT devices. ML offers a different approach, enabling the execution of complex learning tasks on extremely low-power devices, usually consuming less than even a milliwatt of energy. This capability empowers real-time analysis and understanding of data, leading to significant benefits in terms of speed, privacy, and cost (Banbury et al., 2021). ML applied to IoT devices involves the deployment of machine learning models on devices with limited resources, often utilizing cameras. These cameras are affordable and provide input for ML models, enabling real-time analysis without compromising accuracy. (Warden & Situnayake, 2019).

1.2 Problem Statement

In the 21st century the challenge of driver drowsiness remains a very big problem, significantly contributing to many road accidents. This predicament is particularly pronounced in Kenya where driver fatigue is particularly frequent among long haul truck drivers. Despite the implementation of various governmental measures, such as regulating travel time for public vehicles, reducing nighttime bus drivers and the introduction of an alco-blow based operation on highways to identify intoxicated drivers, driver drowsiness remains a persisting problem. Based on the 2018 global road safety report by World Health Organization (WHO), the number of yearly fatalities resulting from road accidents (RTAs) has risen to 1.3 million (World Health Organization, 2018). Most of these accidents, over 93%, happen in countries with lower and middle incomes. The report also pointed out that road traffic deaths are the leading cause of mortality among individuals aged 5 to 29. Additionally, research on road traffic accidents suggests that more than half of the injuries and fatalities around 50%, involve people aged 15 to 49 years which is usually a phase considered to be economically productive (Macharia et al., 2009) and (Sapkota et al., 2016).

The current attempts to combat this issue have been inefficient in providing a comprehensive drowsiness detection system for drivers, weighing down the implementation of important regulations. While there are a limited number of systems in the market, their high costs make them accessible to only the privileged few who can afford vehicles equipped with such technology. According to Wessel (2022), only companies like Mercedes Benz, which have branded their driver drowsiness detection system as “Attention Assist”,” Driver Condition

Monitor" by Land Rover and "Driver Alert" by Volvo are the only few companies to implement a driver monitoring system. Hence there is an urgent need to develop a cost-effective drowsiness detection solution that caters to the majority, including low-income earners and public transportation services. By leveraging Machine Learning technologies, we can offer a unique solution that will address the issue of drowsiness during driving by utilizing the technologies capabilities which are low power consumption and real time capabilities which will enable precise and responsive drowsiness detection systems. By addressing this gap, the intention is to tackle the alarming rate of accidents linked to drowsiness, thereby contributing to enhanced road safety.

1.3 Objectives of the Study

1.3.1 General Objectives

The main objective of this study is to implement an affordable, reliable and resource efficient drowsiness detection system that can be seamlessly integrated into vehicles using IoT and Machine Learning.

1.3.2 Specific Objectives

1. To review the techniques used on existing drowsiness detection systems during driving using IoT and Machine Learning.
2. To explore the techniques used in drowsiness detection systems during driving using IoT and Machine Learning.
3. To propose a cost-effective, dependable, and efficient system for detecting drowsiness that seamlessly integrates into vehicles by leveraging IoT and Machine Learning technologies.
4. To develop an advanced drowsiness detection system by harnessing the capabilities of Machine Learning technology for seamless integration into vehicles.
5. To test the application.

1.4 Research Questions

1. What are the techniques employed in existing drowsiness detection systems for driving using IoT and Machine Learning?
2. How do existing drowsiness detection systems for driving, incorporating IoT and Machine Learning, utilize various techniques?

3. Can a cost-effective, dependable, and efficient drowsiness detection system be proposed that seamlessly integrates into vehicles, leveraging IoT and Machine Learning technologies?
4. How can an advanced drowsiness detection system be developed, harnessing the capabilities of Machine Learning technologies for seamless integration into vehicles?
5. What is the outcome of testing the application for drowsiness detection in real-world driving scenarios using IoT and Machine Learning?

1.5 Justification

The aim of the study is to come up with a solution that will enhance how we address the problem of drowsiness during driving through the utilization of IoT and Machine Learning. This approach will not only contribute to improved road safety by detecting drowsiness but also provide a more efficient and accurate method of managing drowsiness detection. The proposed system's ability to automatically detect drowsiness patterns and warn drivers in real time aligns with the overreaching goal of enhancing road safety by preventing road accidents. As a result, the customer experience is expected to be greatly improved, as drivers will receive timely alerts and assistance to combat drowsy driving instances. This transparent and responsive approach, enabled by the integration of IoT and Machine Learning, will further contribute to safer roads and a more secure driving environment in general.

1.6 Scope and Limitations

This study aims to leverage the potential of IoT and Machine Learning technologies to develop an efficient drowsiness detection system. This system will utilize sensors and cameras to monitor the driver's eye movements and process this data in real time. The integrated Machine Learning model will then analyze the collected data to accurately identify instances of drowsiness. When the system identifies a drowsy state, it will promptly send a signal or an alert to the driver, enhancing road safety.

To achieve this, a prototype will be developed by utilizing a computer camera to collect data. This device will work together seamlessly to capture real time video images of the driver's face allowing for accurate monitoring of eye movement.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

We are going to look at research on how Machine Learning and IoT technologies can be implemented in drowsiness detection systems. Our aim is to thoroughly investigate how these technologies fit into the field of drowsiness detection. Additionally, we'll explore the specific difficulties that drivers encounter when dealing with drowsiness, especially in crowded urban areas.

Some of the challenges contributing to drowsiness during driving include extended periods behind the wheel, monotony of long journeys, sleep deprivation, and the impact of environmental factors like adverse weather conditions. With respect to these challenges, this chapter will aim to analyze existing applications as these systems act as invaluable guardians, like convenient companions, constantly assessing the driver's condition and providing warnings and quick intervention when signs of drowsiness appear.

The current systems work well, but there's a chance to make them sharper and more efficient. This research aims to enhance drowsiness detection systems to be smarter and more effective in ensuring the safety of drivers and everyone else on the road.

2.2 History of Advancements in Drowsiness Detection System Techniques

The first driver drowsiness detection system was introduced to the market by Volvo which was known as the Driver Alert Control (DAC) where the level of drowsiness was evaluated based on performance aspects by analyzing the driver's driving behavior. According to the project manager for Driver Alert Control (DAC) at Volvo cars, Daniel Levin (2007), he stated that the company did not choose to monitor human behavior as it varies from one person to another but instead investigated the effect of fatigue or decreased concentration has on driving behavior. He later added that the system is based on the cars progressing on the road while providing a real time alert to the driver in case something is likely to go wrong before it’s too late.

Researchers have concluded that there are four commonly employed measures for detecting different stages of drowsiness among drivers. Figure 1 below presents an illustration of these measures currently utilized for classifying driver drowsiness levels. Among these measures, two are directly observed in drivers themselves: image-based and biological-based measures. The third measure is derived from the vehicle's data and is known as the vehicle-based measure. Additionally, researchers have explored a hybrid measure, which combines aspects of at least two of the approaches (Albadawi et al., 2022).

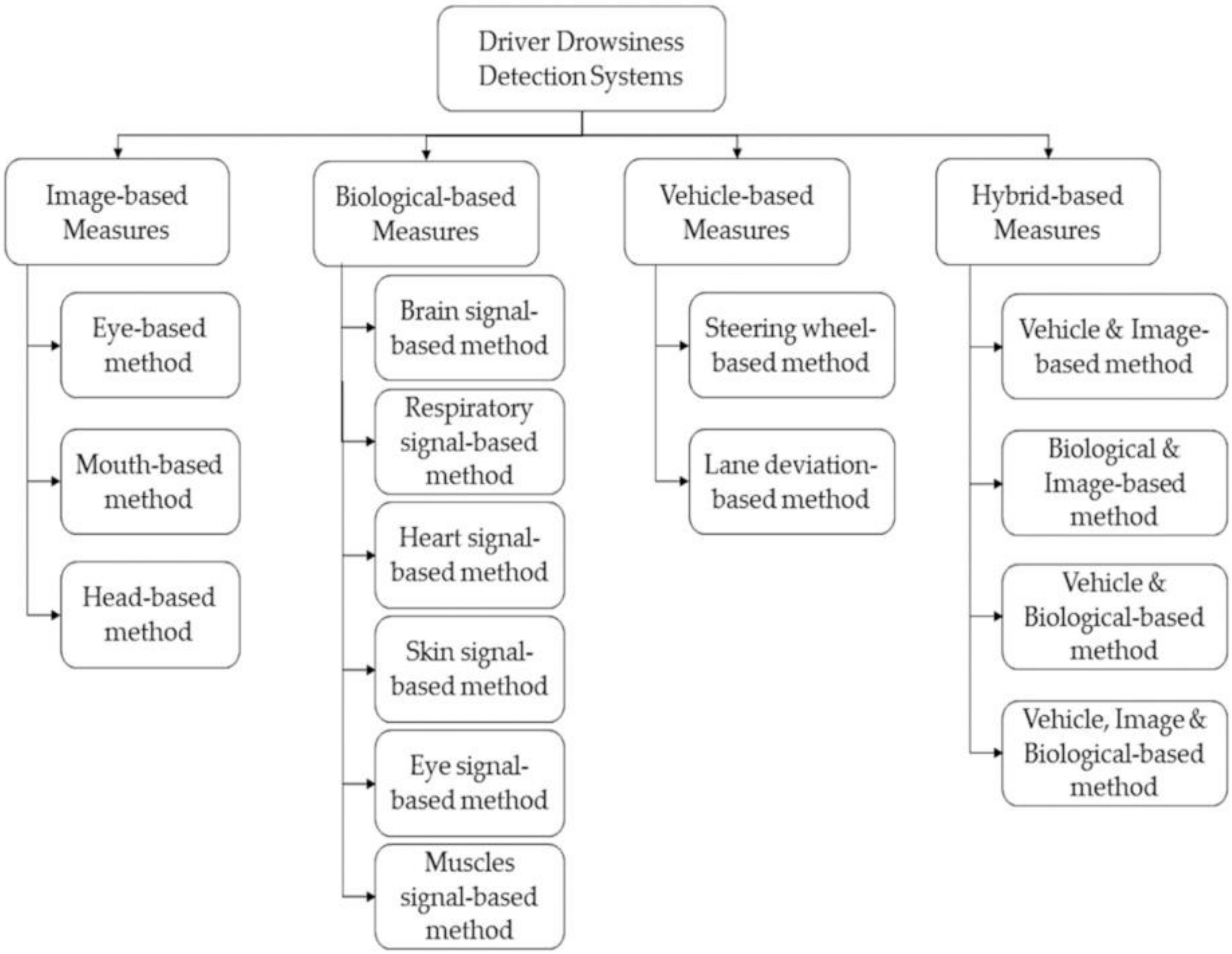


Figure 1 Driver drowsiness detection measures (Albadawi et al., 2022).

The following methods are used in image-based measures during driver drowsiness detection systems. The Eye Drowsiness Detection System is a technology that uses a variety of eye-related parameters and behaviors to assess a person's level of alertness while driving. This system typically uses cameras and computer vision algorithms to track factors such as blinking patterns, duration of eye closure, gaze direction, and pupil dilation. By analyzing these eye-related measurements, the system can detect signs of drowsiness, fatigue or distraction in real time and provide warnings or alerts to the individual to prevent potential accidents or errors that can occur due to loss of alertness (Purnamasari et al., 2021).

A mouth-based method for drowsiness detection involves monitoring a person`s oral movements and expressions to assess their alertness while engaged in tasks like driving or other activities requiring sustained attention. This method typically employs cameras or sensors to track mouth-related parameters such as yawning frequency, mouth opening, and changes in lip and facial muscle movements. By analyzing these mouth-related cues, the system can identify signs of drowsiness or fatigue and provide timely warnings or alerts to the individual to mitigate potential accidents or errors resulting from reduced attentiveness (Alioua et al., 2014).

Head-based methods for detecting drowsiness use a variety of head movements and positions to assess an individual's level of alertness, especially in situations such as driving or demanding tasks. constant concentration. This method often uses sensors or cameras to monitor parameters such as head position, head nodding, or irregular head movements. By assessing these head-related behaviors, the system can identify signs of drowsiness, fatigue or distraction and provide a timely warning or alert to the person to prevent potential accidents that may occur or loss of concentration due to decreased alertness (Teyeb et al., 2014).

The following methods are used in biological-based measures during driver drowsiness detection systems. In brain signal-based method, Passive EEG-based BCI (pBCI) for drowsiness detection is a widely used method to measure and analyze brain activities during wakefulness or sleepiness due to its fine temporal resolution (LaRocco et al., 2020) and (Min et al., 2017). Various methods can be used to analyze EEG signals and determine the level of sleepiness, such as frequency analysis, correlation analysis, and time domain analysis with machine learning (ML) and deep learning (DL).

In respiratory signal-based method, a respiratory signal, which has been obtained using an inductive plethysmography belt, is processed in real time to classify the driver's state of alertness as drowsy or awake. An analysis of the respiratory rate variability (RRV) is employed to detect the fight against drowsiness (Fernández et al., 2019).

In a heart signal-based method, sensors, or devices ae deployed to monitor the drivers heart rate and related cardiovascular parameters to assess their level of alertness while driving. Changes in sleep conditions usually affect the nervous system and subsequently the Heart Rate Variability (HRV), which is defined as fluctuations in the R-wave peak to R-wave peak interval (RRI) on an electrocardiogram trace. In this method, eight HRV features are usually monitored to detect changes in HRV by using a multivariate statistical process control, which is a common anomaly detection method (Fujirawa et al., 2019). The eight commonly monitored HRV features include mean RR interval, standard deviation of NN intervals, very low-frequency, low-frequency, high-frequency, Poincaré plot parameters (SD1 and SD2), and the LF/HF ratio, which collectively provide insights into autonomic nervous system activity and cardiovascular health (Shaffer & Ginsberg, 2017).

In skin - based signal analysis, it involves monitoring various skin-related parameters, such as skin conductance (SC), to assess an individual's alertness during activities like driving. According to Chuwei Ye (2022) SC, can be tracked using comfortable wearable devices like wristbands and rings, making it practical for real-world application. While SC signals can be influenced by daily movements and environmental factors, advanced signal processing techniques are employed to enhance their quality for reliable drowsiness classification by machine learning algorithms (Gianfranchi et al., 2019). Combining SC with other physiological signals, such as electroencephalogram (EEG) or electrocardiogram (ECG), has been explored to achieve higher accuracy in identifying drowsiness states. Hybrid systems that integrate SC with facial features from video or Galvanic Skin Response (GSR) sensors have demonstrated excellent performance in detecting driver drowsiness, offering a non-invasive and efficient solution to enhance road safety by providing timely alerts to fatigued drivers (Amidei et al., 2023).

Driver drowsiness detection based on eye signals involves utilizing computer vision and eye tracking technology to monitor a driver's eye movement and eye lid behavior to assess their level of alertness during driving (Safarov et al., 2023). This method focuses on tracking the driver's eye to determine whether they are open or closed and monitor blinking patterns. When signs of drowsiness such as prolonged eye closure are detected, the system issues an alert or a warning to prevent the likelihood of a potential accident (Borkar, 2015).

In a muscle signal-based method for drowsiness detection, a driver's alertness is monitored on muscle activity typically through an electromyography (EMG), which measures muscle response or electrical activity in response to a nerve stimulation of the muscle (Johns Hopkins Medicine, 2023). This method involves placing sensors in muscle specific areas like the neck or the face area to detect muscle contractions and relaxation patterns. The change in muscle activity is monitored to detect signs of drowsiness, fatigue, or reduced attentiveness. The system then issues an alert when these changes are measured (Satti et al., 2021).

The following methods are used in vehicle-based measures during driver drowsiness detection systems, Steering wheel-based methods for drowsiness detection involve analyzing data collected from the vehicle's steering wheel to assess a driver's level of alertness. Researchers have explored various techniques utilizing steering wheel data for this purpose. For instance, Li et al (2017) introduced a method that calculates the approximate entropy from the Steering Wheel Angle (SWA) signals. They linearized this feature using adaptive piecewise linear fitting and utilized the wrapping distance between linear feature series in a binary classifier to determine drowsiness states. In another approach, frequency, and time-frequency features of SWA were extracted using techniques like Short Time Fourier Transform (STFT) and Wavelet Transform (WT) (Haupt et al., 2011) for drowsiness detection. Ahlstrom et al, (2018) employed a driving simulator to gather eleven steering wheel-related features and used multivariate analysis of variance to identify parameters significantly correlated with drowsiness levels, ultimately reducing the number of parameters to five.

In the lane deviation-based method, sensors placed strategically within a vehicle, such as the driver's seat and steering wheel, to apply vehicle-based measures. Among these measures, the two most prevalent are Standard Deviation of Lane Positioning (SDLP) and Steering Wheel Movement (SWM). SDLP relies on a front-mounted camera to track the vehicle's lane position, enabling the system to discern whether the driver is alert or drowsy based on lane deviation patterns (Doudou et al., 2019). This approach leverages lane tracking technology to assess driver attentiveness and is a commonly used method in the field of drowsiness detection (Bajaj et al., 2023).

In summary, over the years, drowsiness detection systems have gotten much better thanks to new technologies. From basic methods, we now have advanced systems that make driving safer by detecting a driver's fatigue levels when they are on the road.

2.3 Challenges Experienced by Drivers Leading Them to Drowsiness During Driving

Several challenges can lead drivers to become drowsy while driving:

2.3.1 Sleep Deprivation

Not getting enough sleep is a major reason why people feel extremely tired during the day, leading to risky behaviors while driving like microsleeps. While adults should ideally aim for seven to nine hours of nightly sleep, many don't meet this recommendation (Suni & Rehman, 2023).

2.3.2 Sleep Disorders

Sleep disorders like sleep apnea can disrupt and reduce the quality of a person's sleep, leading to daytime drowsiness. These disorders are often undiagnosed and if not treated, can result in ongoing tiredness (Suni & Rehman, 2023).

2.3.3 Alcohol and Drug Use

Alcohol consumption can induce drowsiness and impair reaction time and decision-making, elevating the likelihood of car accidents (Suni & Rehman, 2023).

2.3.4 Medications

Many medications, including sleep aids, can lead to drowsiness. This effect can persist into the next morning, and it's a potential side effect of medications used to treat various medical conditions (Suni & Rehman, 2023).

2.3.5 Nighttime Driving

Nighttime driving poses challenges due to the body's natural circadian rhythms. As it gets darker, our bodies prepare for sleep, leading to the secretion of melatonin, decreased body temperature, and slower digestion. This can result in slower reaction times, fatigue, and difficulty concentrating while driving at night, affecting overall alertness, and increasing the risk of drowsiness-related incidents (Burns, 2023).

2.3.6 Timing and High-Risk Groups

Auto accidents caused by drowsy driving are most common during the early morning hours between midnight and 6 a.m. and during the mid-afternoon, times when sleepiness tends to peak. While drowsy driving can affect anyone, certain groups are at higher risk of such accidents, including individuals with professions that involve extensive driving like long-haul truckers and bus drivers, those working long or irregular hours, individuals with significant sleep issues such as insomnia, and teenagers, who often have limited driving experience and insufficient sleep (Suni & Rehman, 2023).

2.4 Existing Drowsiness Detection System During Driving Applications

Locally, there are not any existing drowsiness detection systems but as for on the global scope there exists a variety of applications and research in this field. Numerous research studies have employed a range of deep learning models, including AlexNet, CNN, VGG-16, and RNN, for the purpose of identifying drowsiness in drivers based on videos and images captured through IoT (Internet of Things) devices.

2.4.1 An intelligent Mobile App to Detect Drowsy Driving with Artificial Intelligence (Sleepy Eyes)

In this application, Xiao & Sun (2021) introduced an innovative approach to detect and mitigate drowsy driving incidents. The primary objective was to develop a reliable algorithm that utilizes eye openness as a key indicator to assess a driver's alertness. This method harnessed the capabilities of Google Firebase's eye detection algorithm, coded in Dart language and utilizing the Flutter Camera package plugin to access the smartphone's front-facing camera. When the user initiated the application by pressing the start button, the algorithm automatically captured 10 photos per second, subjecting each image to analysis through Google Firebase's eye openness detection feature. In this approach, Google Firebase's eye detection could effectively measure the degree of eye openness, assigning a value between 0 and 1, where 0 indicated closed eyes and 1 indicated fully open eyes. By using this data, Xiao & Sun (2021) performed calculations based on the information provided by Google Firebase to determine if the driver was exhibiting signs of drowsiness. The algorithm included a specifically designed calculation method, detailed below, to determine when it was necessary to alert the user about their drowsy driving state. The algorithm begins by taking an image of the user and storing it temporarily, while the face detector checks for any faces and retrieves the user's eye value. If the eye value is below 0.3, indicating drowsiness, a variable called "closed_to_total" is incremented based on

```
closed_to_total = closed_to_total*(3/4) + 1/4;
```

If the eye value is above 0.3, the calculation proceeds differently.

```
closed_to_total = closed_to_total*(15/16);
```

This would lead to decreasing the progress bar slowly to encourage the user to rest, while a notification sound plays when "closed_to_total" surpasses 0.5. This calculation process ensures timely notifications to prevent drowsy driving, as illustrated in the user interface below.

Figure 2 Sleepy Eyes User interface

2.4.2 Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques

In this application, Jabbar et al (2018), utilized the use of a Multilayer Perceptron Classifier (MLP) for data processing. The MLP is a neural network consisting of interconnected nodes that map input data to output classes. This architecture is depicted in the figure below.

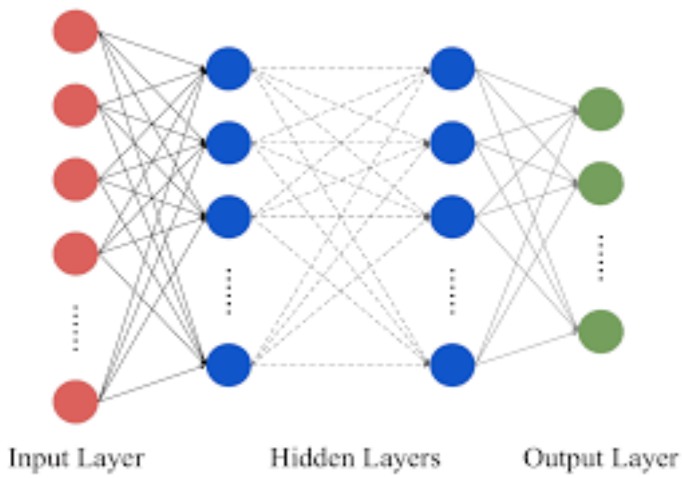


Figure 3 MLP Architecture.

To train the MLP, the authors of this study employed a set of training data containing input and output vectors, allowing the model to learn parameters iteratively until it achieved the desired input-output mapping. These parameters, referred to as the set theta, encompass weight matrices and bias vectors.

The dataset used for this application is the National Tsing Hua University (NTHU) Driver Drowsiness Detection Dataset, featuring recordings of 22 subjects under various driving scenarios and lighting conditions. Infrared (IR) illumination is utilized to acquire IR videos with a resolution of 640 X 480 at 30 frames per second. The dataset includes both training and testing components.

Figure 4 NTHU dataset.

VT OMNES VNVM SINT

The model preparation process involved five key steps:

1. Extracting videos from the NTHU Database,
2. Extracting images from video frames,
3. Extracting landmark coordinates from images using the Dlib library,
4. Training the algorithm
5. Model extraction.

After the model had been trained, it proceeded to evaluate a driver's drowsiness by analyzing facial landmarks and was subsequently stored for utilization in a mobile app. This app made use of the mobile camera to capture images of the driver's face, processed them through the Dlib Library via the Java Native Interface (JNI) framework, identified facial landmarks, and determined drowsiness based on the trained model. When drowsiness was identified, the app alerted the driver through visual and audio signals.

2.4.3 Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer-Vision and Eye-Blink Analyses for Enhanced Road Safety

In this application, Safarov et al (2023), proposed a comprehensive approach for the detection of driver drowsiness, which involved simultaneous assessments of eye-blinks and yawning. The drowsiness levels of drivers were categorized into three distinct states: one based on yawning, another based on eye-blinking, and a third that combined both factors, as demonstrated in the figure below.

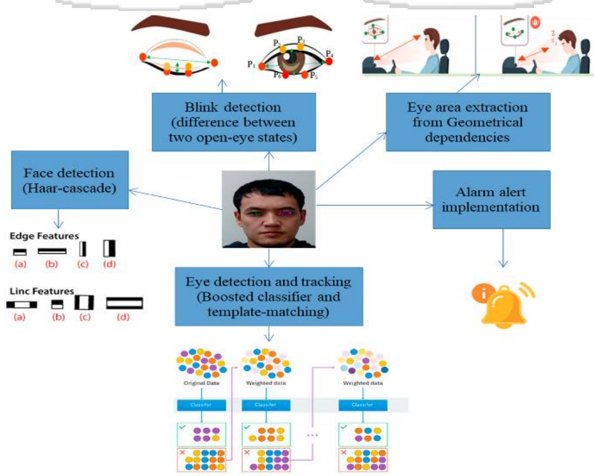


Figure 5 The proposed method with algorithm application, eye tracking, and distance measurement of eye states and between the camera and the driver's head location.

To enhance blink accuracy, iris-tracking deep learning algorithms are integrated into the model, aiding in the classification of closed and open eyes. By ensuring alignment between "closed eye" classifications and landmark metric computations, the model minimizes false positives, resulting in improved accuracy and reduced overestimated drowsy alerts. The application focused on detecting the driver's face and eyes using a medium-pipe face mesh model, concentrating on a single face for drowsiness detection. Blink detection relied on distance-based measurements derived from eye landmark movements, normalized based on eyelid movement relative to eye size. The application solution also leveraged electrooculography (EOG), a well-established method in fields like neuroscience and human-computer interaction (HCI). EOG recorded electrical signals from eye movements through strategically placed electrodes, offering a cost-effective, high-resolution solution for monitoring driver drowsiness and contributing significantly to road safety.

2.5 Gaps In The Existing Solutions

In the first application, the main challenge the researchers experienced was figuring out how the app could always collect information from the user through the phone's camera without any problems, since previous studies did not focus on their implementations on smartphone devices (Xiao & Sun.,2021).

In the second application, the major limitation of the study occurred when drivers' faces assumed different positions, such as turning their heads entirely to the right or left. The Dlib Framework, employed for facial landmark extraction, struggled to detect, and process the images in such instances. Consequently, the images related to these head positions had to be excluded from the dataset used in this research (Jabbar et al., 2018).

Finally, in the third application, the primary challenge of this application was when the drivers were wearing sunglasses. This prevented the model's ability to detect facial landmarks and, consequently, its capacity to accurately analyze eye blinks. This was due to the model’s heavy reliance on facial landmarks, which may have been altered when the sunglasses were present (Safarov et al., 2023).

A solution to mitigate these gaps in the existing systems is to implement a Machine Learning and IoT solution. This will enable lightweight machine learning models to run on edge devices hence enabling real time processing and analysis while IoT facilitates the efficient collection of this information. Through this combination, the proposed system will be able to adapt to various challenges and improve on its accuracy while enhancing the overall robustness during drowsiness detection during driving.

2.6 Conceptual Framework

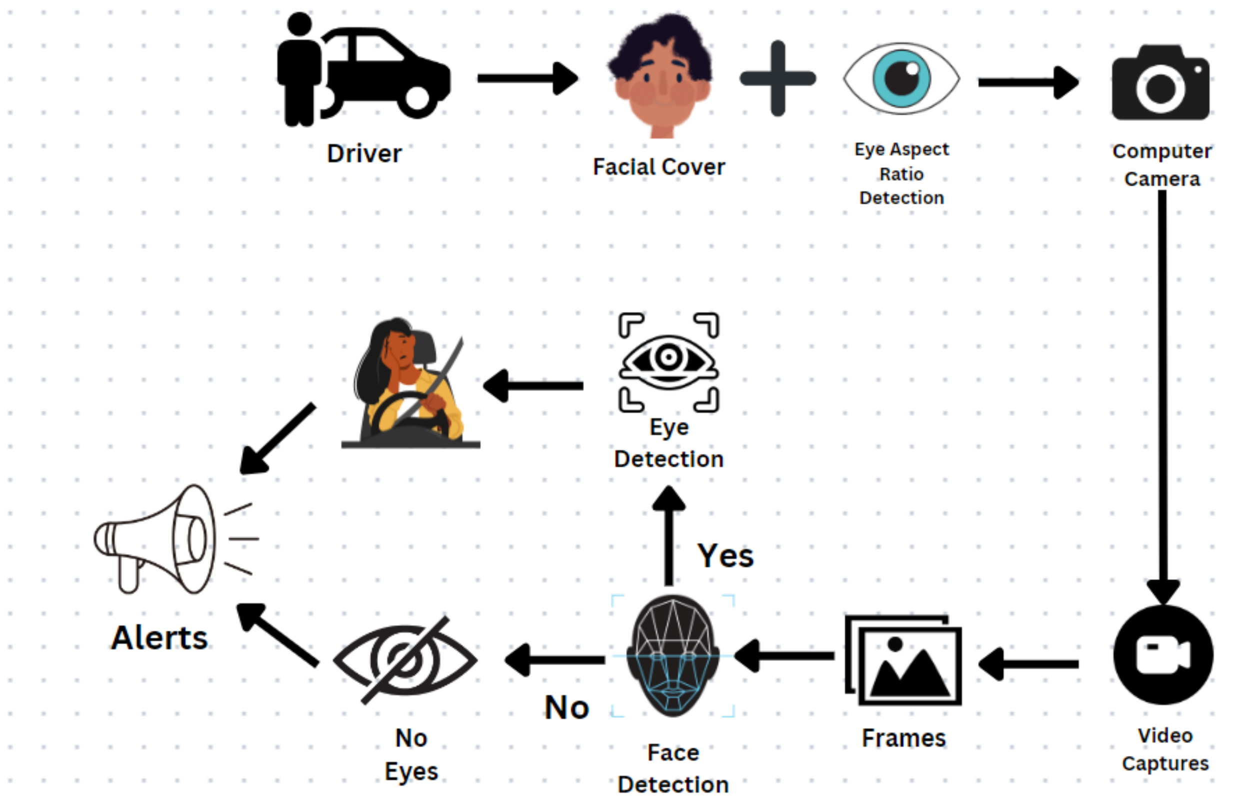


Figure 6 Conceptual Framework

According to the figure above, the needs of the system are provided by the driver where a pi camera sensor collects the facial cover and eye aspect ratio (EAR) in real-time in form of videos and sends the information to the system. The data is then processed where the video captures are converted into frames. A Machine Learning algorithm then can detect the facial landmarks which in this case are the eyes to determine if they are open or closed. If they are closed the system sends an alert to the driver.

CHAPTER 3: METHODOLOGY

3.1 Introduction

According to Bouchrika (2023), the primary objective of research isn't just about collecting information, it is rather about exploring and providing solutions to unanswered questions, thereby contributing to existing knowledge in a specific field. In this chapter we are going to explore the key elements that make up the basis of our research. These include how we planned our study, who and where we studied, how many people were involved, and how we collected and analyzed data. Furthermore, we are going to explain methods we will use to create our research system and how we plan to share our findings with others. Lastly, we will look at the ethical considerations of our research to show that we are following the right rules and principles. These elements together will form the core of our research approach.

3.2 Research Design

The primary objective of this research is to develop a drowsiness detection system and produce, test, and design a prototype that will be utilized to solve the problem of drowsiness during driving. Research design refers to the set of methods and techniques that a researcher selects to carry out their study (Bhat, 2023). There are main types of research including, basic research, correlational research, applied research, experimental research, exploratory research, grounded research, descriptive research, phenomenological research, quantitative research, and qualitative research (Physiopedia, 2023).

In this study, we will employ an applied research approach, which encompasses various phases. These phases involve identifying a market demand for the product, creating a product with the potential to fulfill those demands, constructing a prototype, and evaluating whether the prototype successfully meets the desired criteria concerning cost, environmental impact, and profitability when it is introduced to the market (Gillis, 2022). This study aims to develop a drowsiness detection system capable of collecting data and transmitting it into a Machine Learning model for analysis.

3.3 Population and location

According to Bhandari (2023), a population is an entire group in which a researcher draws conclusions about while a sample is the specific group where we will collect data from. The population for this study will encompass individuals of both genders, specifically focusing on truck drivers and bus drivers. This diverse group of drivers will provide valuable insights into the research objectives and help ensure a comprehensive understanding of the factors under investigation. The research will take place in Nairobi County, Kenya.

3.4 Sample Size

This study will utilize a probabilistic sampling approach, where randomization will be employed to ensure that every element within the population has an equal opportunity to be part of the selected sample. To achieve this, the researcher will implement a method known as stratified sampling. This technique involves categorizing the population elements into distinct groups based on their similarities and differences. Each group will consist of elements that are like each other but different from those in other groups (Singh, 2018). The researcher will target a maximum of five people who will have prototype installed in their cars and give feedback. This number of people will be ideal since it will be costly to develop the prototype and hence few people who will be selected will represent others.

According to CEIC DATA (2021), as of December 2021 the number of registered vehicles was reported at 2,095,088 units. This was an increase from the previous number of 1,987,589 units from the previous year. They also added that the number of registered lorries, trucks and heavy vehicles was at 159,128 units. The sampling formula that will be used for random sampling is shown in the equation below.

VT OMNES VNVM SINT

$$n = \frac{\frac{z^2 * P * (1 - P)}{e^2}}{1 + \frac{z^2 * P(1 - P)}{e^2 N}}$$

Where:

n = Sample Size

z= z-score (we are working with 1.96 a confidence of 95%)

e= error margin of 5% or 0.05

p = standard deviation (0.5)

N = Population size

The result of this has shown the researcher that the work will have a stratum of 418 people where they will be selected from.

3.5 Data Collection Methods

The sole purpose of this research is to collect data from the research subjects who are in line with our topic of interest. On the other hand, an ideal instrument yield's objective, accurate, sensitive, efficient, and relevant results. The following tools will be used in this study.

3.5.1 Survey

Interviews and questionnaires will be conducted as research instruments to collect data from truck drivers. The procedure will be carried out by having multiple people respond to questions while the interviewer will take notes on the user's feedback.

3.5.2 Prototyping

The prototyping phase will involve sharing the developed application modules with the public to gather constructive feedback, especially regarding user satisfaction and the presence of desired features. This iterative process will involve refining the system based on user input and engaging colleagues and classmates to assist in its development. Their insights and suggestions will play a valuable role in enhancing the smart metering system.

VT OMNES VNVM SINT

3.5.3 Documents Review

As part of this research, we will examine various documents, including related studies in the field of drowsiness detection systems. This review aims to offer valuable insights into existing implementations, providing critical feedback, and expanding our comprehension of effectively executing the proposed solution.

3.6 Data Analysis Methods and Presentations

Information collected from participants will be presented in a numerical format using statistical data derived from quantitative research techniques. The primary goal of this study is to acquire precise data to ensure the accuracy of the results. After collecting and analyzing the data, the findings will be visually represented using pie charts and bar graphs.

3.7 Research Quality

Both the reliability and validity of the study will contribute to the study's overall quality. Validity is used to describe the accuracy of the conclusions drawn from an evaluation, involving inferences made from data relevant to score interpretation. On the other side, reliability indicates repeatability and stability over time. In the case of tests, lower reliability implies more measurement errors, making another definition of reliability the degree to which a test is devoid of measurement errors (Bruin, 2010).

Correctly applying the methodology will guarantee reliability in the study. The researcher will carefully plan the methods to ensure each phase of the study progresses smoothly. Additionally, reliability will be ensured by standardizing the research conditions to minimize the impact of external factors on the data collection process, as variations in such factors can influence the outcomes.

On the other hand, ensuring validity will involve the careful selection of suitable measurement methods. The researchers will make certain that the methods and measurement techniques chosen are of high quality and effectively assess the necessary aspects for the study. Furthermore, the researcher will conduct extensive literature review in line with the research objectives to leverage existing knowledge. Additionally, validity will be upheld by employing appropriate sampling methods tailored to the study's requirements, as suggested earlier. The utilization of the stratified sampling technique, as previously outlined, will be instrumental in fulfilling all the study's research goals. The researchers will also ensure a clear definition of the study's population concerning the scope, geographical distribution, and profession (Paul C. Price, 2017).

Finally, the robustness and consistency of the information collected during the study will be verified. This will be done by comparing our findings with existing research and seeking guidance from our academic supervisor to ensure the quality and reliability of our data.

3.8 System Development Methodology

The study will adopt agile software development methods, which involve breaking down the work into smaller, manageable phases and concentrating on short-term planning. The project's objectives, scope, and the number of phases will be defined upfront. Each phase will last from one to four weeks and will encompass activities such as requirements analysis, design, coding, testing, and validation. Agile methodology is chosen due to its practical approach, emphasis on teamwork and skill development, the ability to demonstrate functionality, and reduced resource needs.

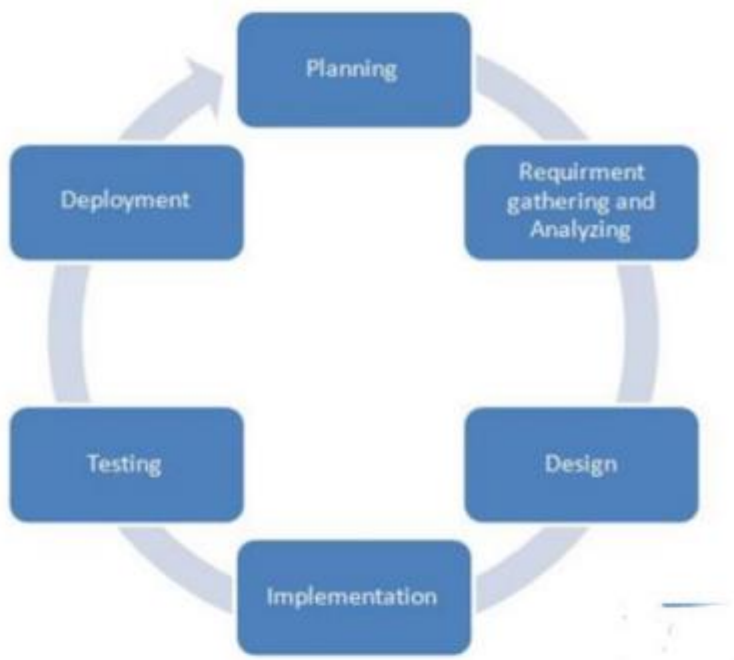


Figure 7 Agile Development System Methodology (Lithmee, 2018).

3.8.1 Planning

In this stage, we will set out all the project requirements and goals. Planning will help us organize the project's various phases and clarify the roles and responsibilities of each participant or organization involved. During the planning phase, we'll also consider the financial and resource requirements for implementing the solution, including both hardware and software tools, which we'll discuss further during the implementation stage.

3.8.2 Requirements Gathering and Analysis

During this phase, we will concentrate on identifying the technologies required for building a drowsiness detection system. To do this, we will conduct a thorough review of existing literature to pinpoint these technologies. We'll also examine device manuals and datasheets to gain insights into their structures, pin configurations, and operational principles. Additionally, we will identify the most suitable software and libraries for use in the development process. This comprehensive evaluation will guide us in determining the most effective approach for creating both the hardware and software components, ultimately leading to the final product.

3.8.3 Design

The system's design will encompass both its architectural and visual aspects. The architectural component will rely on sensors, controllers, and other physical devices. In contrast, the visual design will employ computer programs to create structure charts, data flow diagrams, and Hierarchical Input Process Output (HIPO) diagrams. These various diagrams serve different purposes: data flow diagrams illustrate how data moves in and out of the system, structure charts offer more detailed breakdowns of the system into functional modules, and HIPO diagrams provide a holistic view of the system and its modules.

3.8.4 Implementation

In this phase, a prototype will be developed by constructing the system in separate components, which will later be integrated and tested indoors to assess if they align with the set objectives and requirements. This development process will involve a combination of hardware and software tools. The hardware aspect will include custom-made sensors and microcontrollers. For software development, an integrated development environment (IDE) like Visual Studio will be utilized for creating firmware and web applications. Programming languages such as C++ and Python will be employed at the hardware level. Additionally, Firebase will be set up to store data and host the web application.

VT OMNES VNVM SINT

3.8.5 Testing

In this phase, testing will be done by deploying the prototype to a vehicle environment. The test parameters will be whether the device sends data from the camera and processes the images to determine the level of alertness during driving. The system will be expected to be a portable plug and play device and the data collected will be real time and accurate.

3.8.6 Deployment

During this stage, the device will be taken to the field for data collection purposes. Users will receive guidance on utilizing the web application for data visualization. The system's deployment will specifically take place in collaboration with Strathmore University, located within Nairobi County.

3.9 Ethical Considerations

To ensure the study's credibility and the integrity of its findings, obtaining institutional approval will be essential. Strathmore University, which holds certification, will provide the necessary ethical clearance for the project. Throughout the research process, several ethical considerations must be addressed, including issues of confidentiality, validity, voluntary participation, informed consent, potential risks, and research methodologies. To comprehensively address these ethical concerns, the research team will take the following measures:

The researchers will establish a trust agreement, which will serve as a framework for interactions between them and the participants. This agreement will ensure that both parties provide informed and unequivocal consent to meet the study's requirements.

In line with ethical guidelines, particularly the third principle recommended by the Economic and Social Research Council (ESRC), stringent measures will be in place to safeguard the confidentiality of information shared by research participants and protect the identities of respondents (Smith, 2003). In cases where complete confidentiality is not feasible, the principle of anonymity will be strongly encouraged.

The study will prioritize the safety and security of all participants by providing them with necessary safety equipment, training, or access to consultation services, as needed.

The research will strictly adhere to informed consent guidelines established by the National Commission for Science, Technology, and Innovation (NACOSTI), as mandated by Kenyan law.

These ethical safeguards aim to ensure the integrity of the research process and the well-being of all participants involved.

CHAPTER 4: SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

This chapter outlines the system requirements, both functional and non-functional, the system analysis and the system design diagram which were used in the development of the system.

4.2 System Requirements

System requirements analysis refers to the process of determining the user expectation of a new system. Some of the system requirements reviewed in this project include:

4.2.1 Functional Requirements

Functional requirements focus on how the software works and specify the desired behavior of the system. A drowsiness detection system leveraging IoT and Machine Learning will involve a blend of hardware sensors, embedded software, and machine learning models. Functionally, it will entail the acquisition of data, real time processing, machine learning model development, drowsiness detection, alerting mechanisms, integration/connectivity, and a user interface.

In the Data Acquisition stage, it will involve integrating sensors like a camera to capture eye and mouth movements. Real time processing is crucial for minimizing the latency and on device processing using Machine Learning techniques.

For the drowsiness detection part, the relevant features which are the eye and mouth aspect ratios are extracted and classified using machine learning models. The system then utilizes static thresholds predetermined during the model development phase. These thresholds are set based on extensive testing and analysis to ensure detection performance across a range of users and conditions.

Alerts are then promptly issued in real time when drowsiness is detected. Users receive an immediate alert message to prompt appropriate action, enhancing safety and preventing potential accidents due to drowsiness behavior during driving.

4.2.2 Non-Functional Requirements

The non-functional requirements of the project include:

I. System Usability – the user interface of the system is easy to use, and it provides the user a quality user experience when interacting with the system.

II. System Availability – this is done by ensuring that the system is hosted on a reliable real time platform which in this case is Firebase cloud database.

III. System Performance – by using firebase performance monitoring SDK to collect performance data from the application, review and analyze that data in the firebase console so that we can understand in real time where the performance of your app can be improved so that we can use the information to fix the performance issues.

IV. System Security – the application uses encrypted password and communication channels from the IoT devices to the firebase database.

V. System Scalability - The system should be able to work even in a changed size or scale.

VI. System Portability - The system should be used in other operating systems rather than the in created without requiring major rework.

4.2.3 Domain Requirements

The following are the domain requirements for this study:

- Windows 10 operating system.
- Central Processing Unit (CPU) with Core i5 2.4 GHz.
- 8 GB of Random Access Memory (RAM).
- 8 GB of free hard disk space.

4.3 System Architecture

The system architecture outlines the connections among different components of the system, illustrating how they collaborate to achieve its functionality. It also details how the system's structure, design, and user requirements are accounted for. The figure below illustrates the subsystems and how they are connected to each other to achieve the drowsiness detection system. The sensor module consists of a camera sensor which is used to collect data related to the physiological state of the user particularly focusing on the eye and mouth movements. The data is then sent to a real time firebase database system where it is then fetched during the data preprocessing stage. In the data pre-processing module, the raw data collected, undergoes preprocessing to extract relevant features such as the eye aspect ratio and mouth aspect ratio which will then serve as input for the subsequent machine learning algorithms. In the machine learning module, it includes the trained machine learning model responsible for analyzing the preprocessed data and classifying the user's drowsiness levels based on learned patterns features.

In the drowsiness classification module, the output from the machine learning algorithm determines the level of drowsiness categorizing it as either drowsy or not drowsy. Finally, in the alerting mechanism module, the component triggers alerts in real time to notify the users through a text to speech auditory message cue prompting them to take appropriate action.

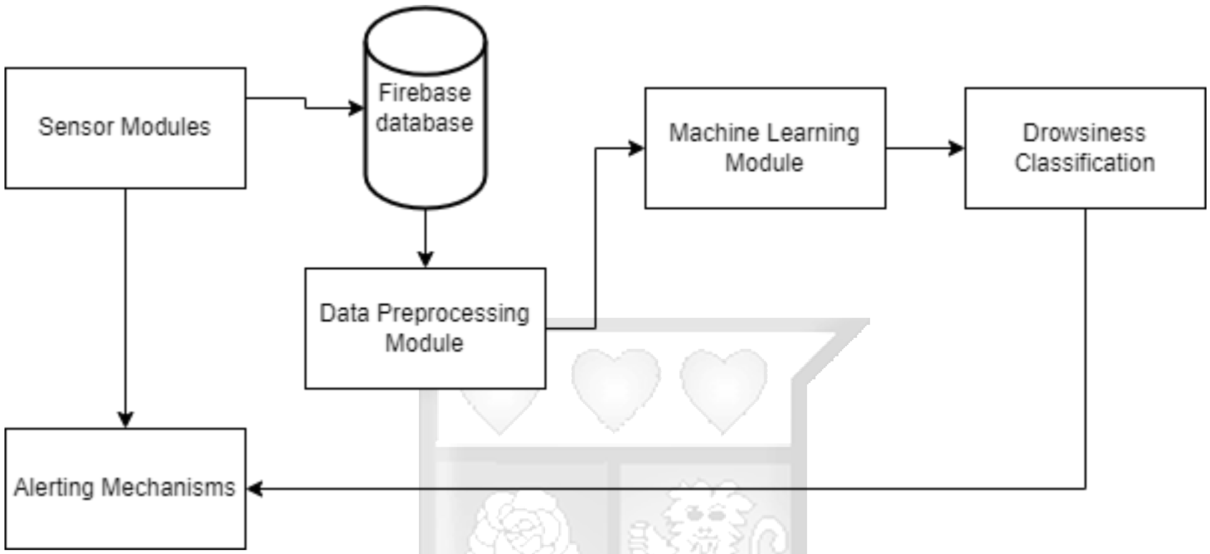


Figure 8 System Architecture

4.4 System Design

System design involves utilizing UML diagrams, which are graphical representations illustrating the static structure of the system. These diagrams depict system elements such as classes, attributes, methods, and relationships between different objects, providing a visual overview of the system's architecture and components (Jin, 2022). Various types of diagrams, including system sequence diagrams, use case diagrams, partial domain models, context and data flow diagrams, entity diagrams, and class diagrams, were employed throughout different stages of the design process. These diagrams were utilized to depict, explain, and document the functionality of the system comprehensively.

4.4.1 Use case model

A use case model is a visual representation of the interactions between an actor and the system (Daly, 2023). A system is usually divided into use cases and actors using design diagrams. To properly arrange the various stake holders for the new proposed system, use case diagrams depicting system behavior were created as illustrated in the figure below.

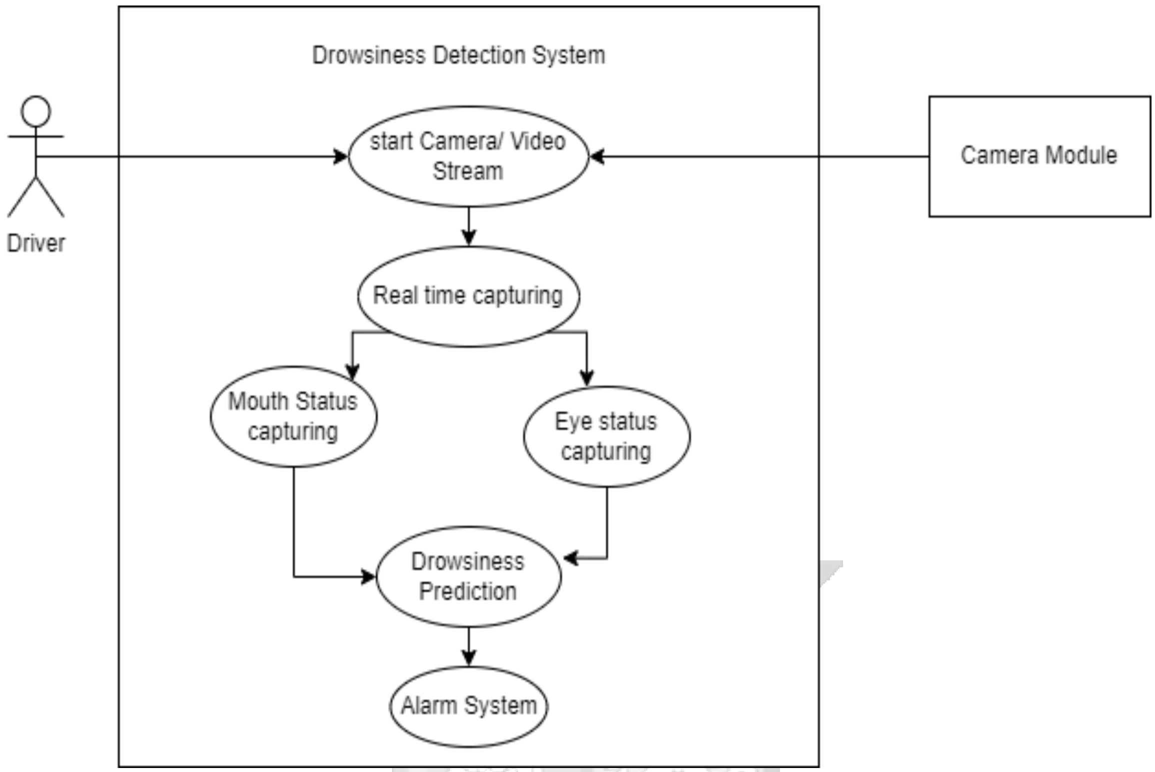


Figure 9 Drowsiness Detection System Use Case Diagram

4.4.2 Use Case Descriptions

4.4.2.1 Real time capturing

This is a detailed process of how real time capturing is implemented into the system. The table below shows the scenario.

Use Case	**Real time Capturing**
Description	A user stands Infront of a camera and his facial landmarks are highlighted accurately and captured in real time
Source	Driver
Inputs	Eye Aspect Ratio and Mouth Aspect Ratio
Preconditions	The user is in a well-lit room. The user has enabled Camera and Audio Settings. The user is above 18 years old.
Postconditions	The system captures the user’s physiological features.
Flow of events	The system loads up the detector and predictor model.

	The system starts the video stream. The user stands in front of the camera. The system highlights the facial landmarks to be captured in real-time. The system captures the user's physiological data and sends it to a database.

4.4.2.2 Real time Alert Functionality

This use case entails instantly notifying the user upon detection of drowsiness. The table below shows the scenario.

Use Case	**Real time Alert Functionality**
Description	A user is instantly notified if drowsiness is detected by the system
Source	System
Inputs	Drowsy (1) / Non drowsy (0) labels
Preconditions	The system captures the users' physiological markers (Eye and Mouth Aspect Ratios). The system assigns labels of either drowsy or not drowsy depending on the users' physiological state.
Postconditions	The system Alerts the user using a text to speech functionality implemented in the system.
Flow of events	The system does feature extraction on the facial physiological features (Eyes and Mouth). The model performs a prediction based on the facial landmarks and assigns labels as either drowsy (1) or not drowsy (0).

	If the drowsy label (1) is assigned. The system calls the alert method and an alert through text to speech is given to the user to notify them of their state.

4.4.3 Data Flow Diagrams

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement (Nolle, 2021).

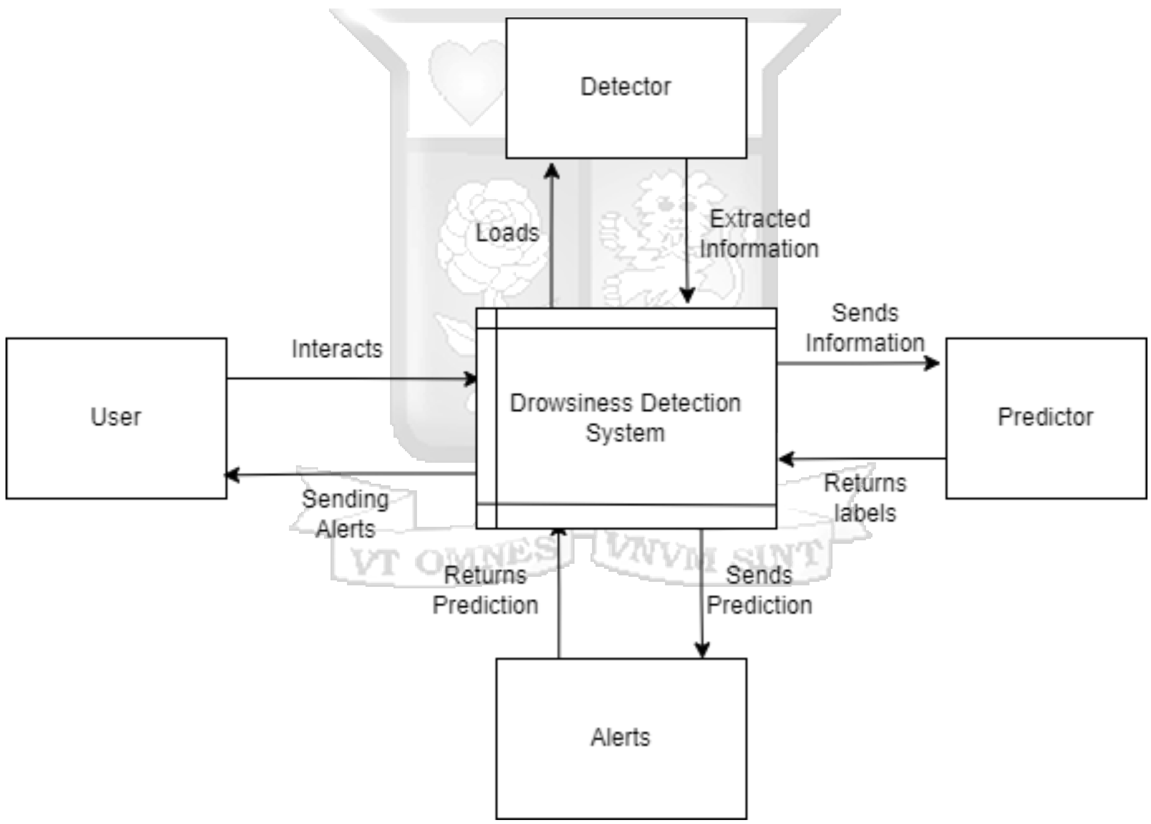


Figure 10 Drowsiness Detection system Data Flow Diagram

4.4.4 Sequence Diagram

A Sequence diagram is a type of interaction diagram that illustrates the order, and the way processes interact with each other (Thoutam et al., 2022). The figure below shows the scenario.

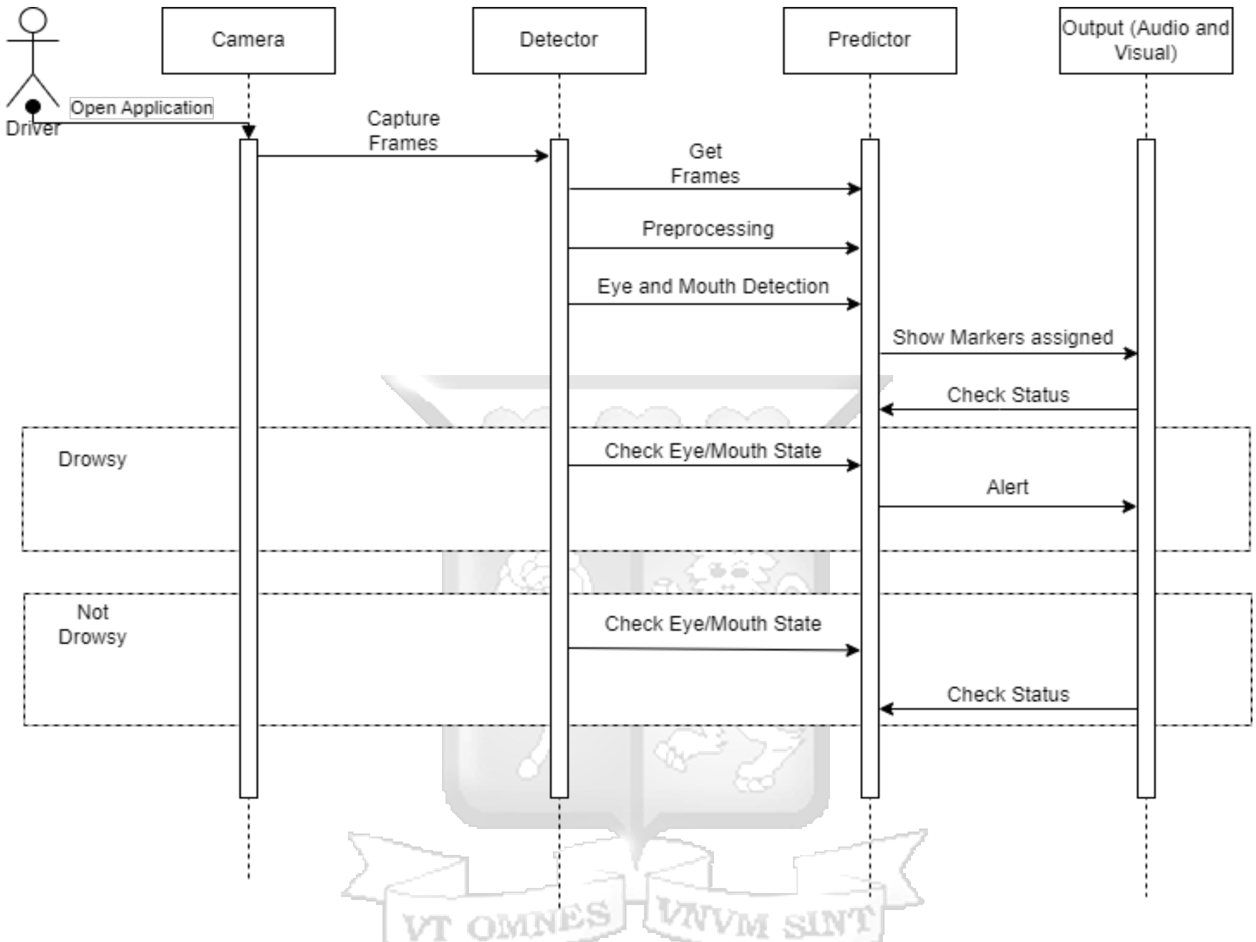


Figure 11 Drowsiness Detection System Sequence Diagram

4.4.5 Flowchart Diagram

A flowchart is a graphical representation that depicts the sequence of steps, decisions, and actions involved in a process or workflow. The figure below shows the scenario.

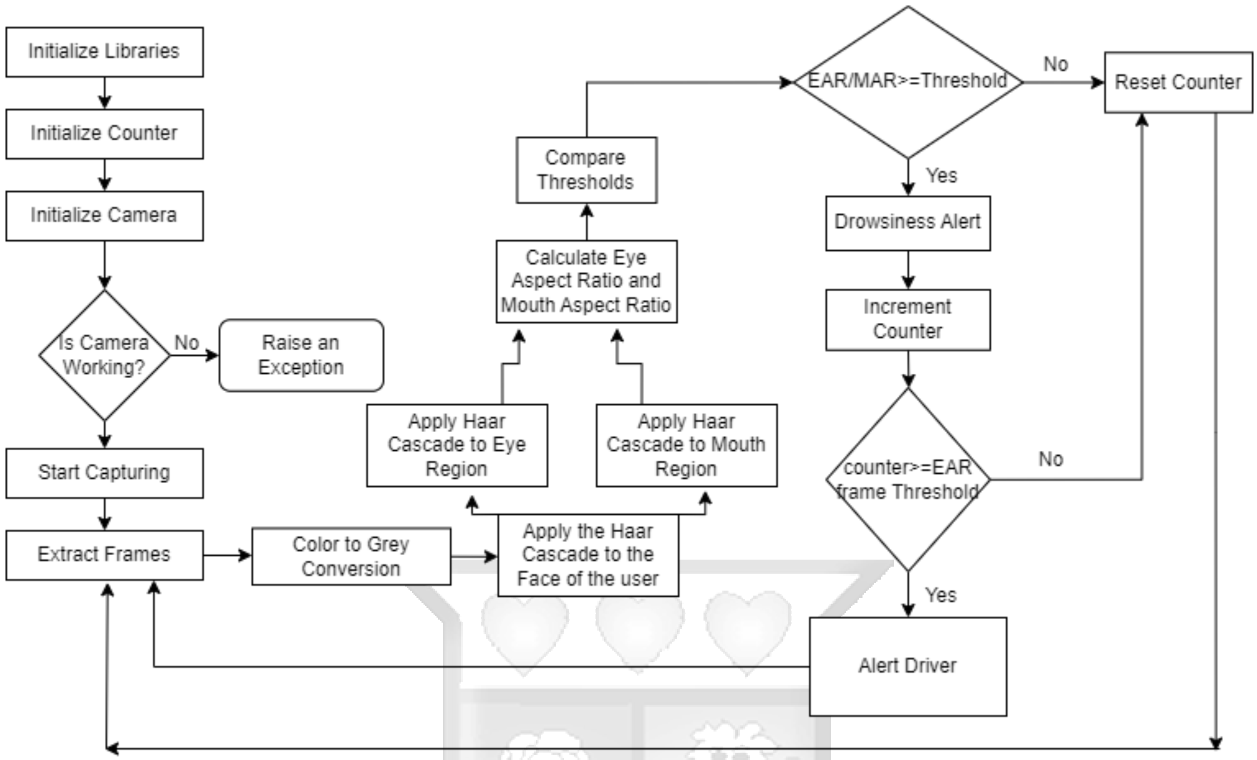


Figure 12 Drowsiness Detection System Flowchart Diagram

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Introduction

System implementation refers to the phase in the software development cycle where the designed system is constructed, coded, tested, and deployed into a live environment. This involves translating the system design specifications into a functional system through programming, configuration, and integration of various systems (Royce, 2021). As for system testing, it is a phase in the software development lifecycle where the entire system is evaluated to ensure that it meets the specified requirements, and it works correctly (Murch, 2012).

5.2 Development Environment

To ensure a smooth implementation process, an appropriate development environment was carefully chosen. The necessary elements for the development process, including hardware, software, and cloud services, are outlined as follows:

5.2.1 Hardware Requirements

Hardware	Description
HP EliteBook 840 G5	The system is running on a Intel(R) Core (TM) i5-7300U CPU @ 2.60GHz 2.71 GHz, With 8GB of RAM on a 64-bit operating system, x64-based processor with an inbuilt camera module.

5.2.2 Software Requirements

The table below illustrates the software requirements for the drowsiness detection system.

Software	Description
Operating System	Windows 10 and above / Raspberry Pi Ubuntu
VSCode	Vscode IDE version 1.7* or higher
Programming Language	Python 3.11.3

5.2.3 Cloud Service Requirements

The table below illustrates the cloud services used to implement the drowsiness detection system.

Cloud Services	Description
Google Firebase	Firebase real-time database was used as a NoSQL database. Google provides this service as a free service for running tests.

5.3 Database Configuration

In this study, a Google Firebase Realtime database was created for storing the drowsiness data collected by the camera sensors for preprocessing.

5.3.1 Setting up the Firebase Database

The first step involves navigating to the firebase console platform (console.firebase.google.com) as shown in the figure below.

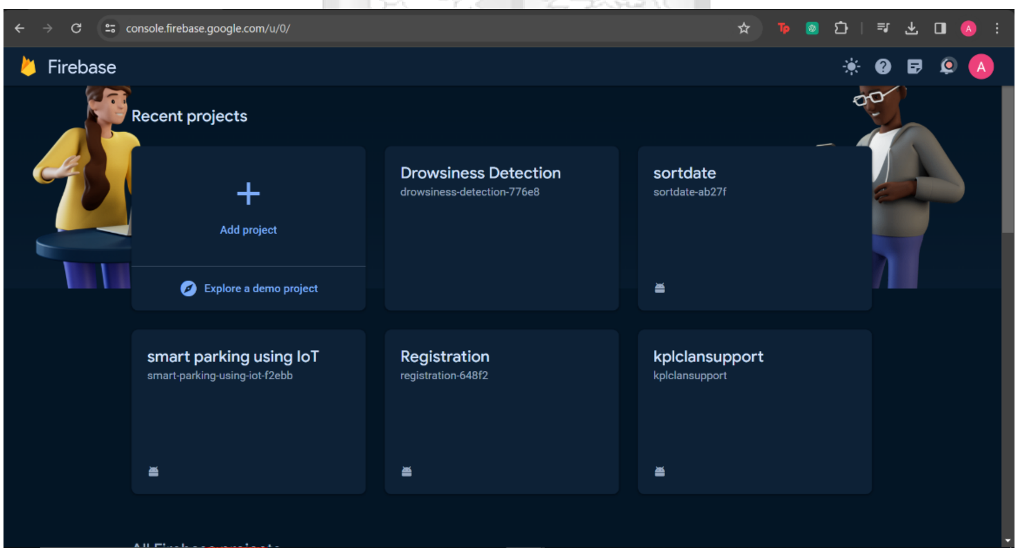


Figure 13 Firebase Console Window

The next step is to click on "Add Project" and follow the prompts to create a new project where we will give our project a name and select our preferred firebase service as shown in the figure below.

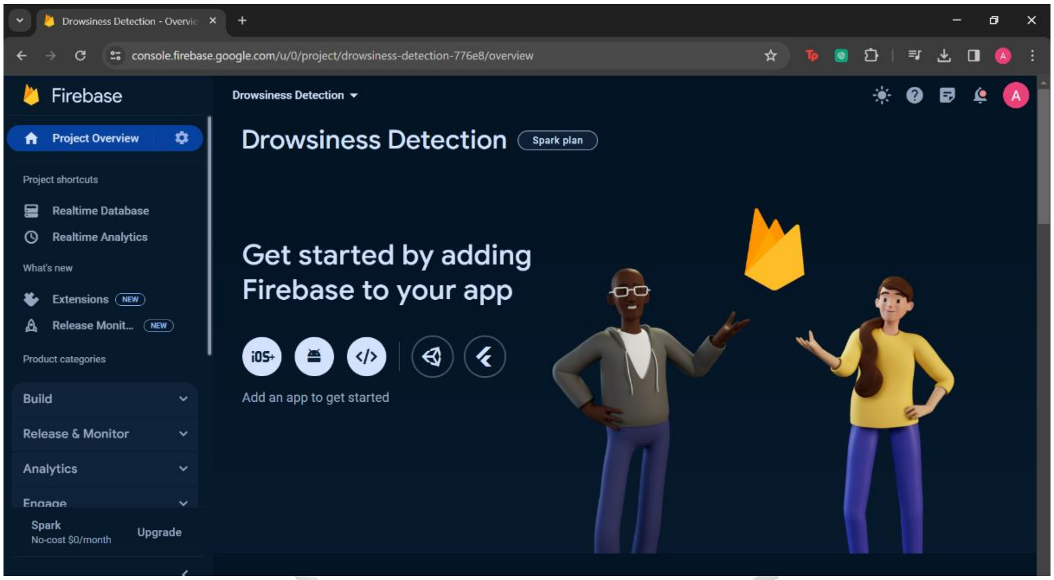


Figure 14 Drowsiness Detection Console Page

The next step is to initialize our application by installing the Firebase SDK on our code by adding the necessary dependencies and libraries to our project. This will involve initializing Firebase into our application by adding the Firebase configuration provided in the Firebase Console.

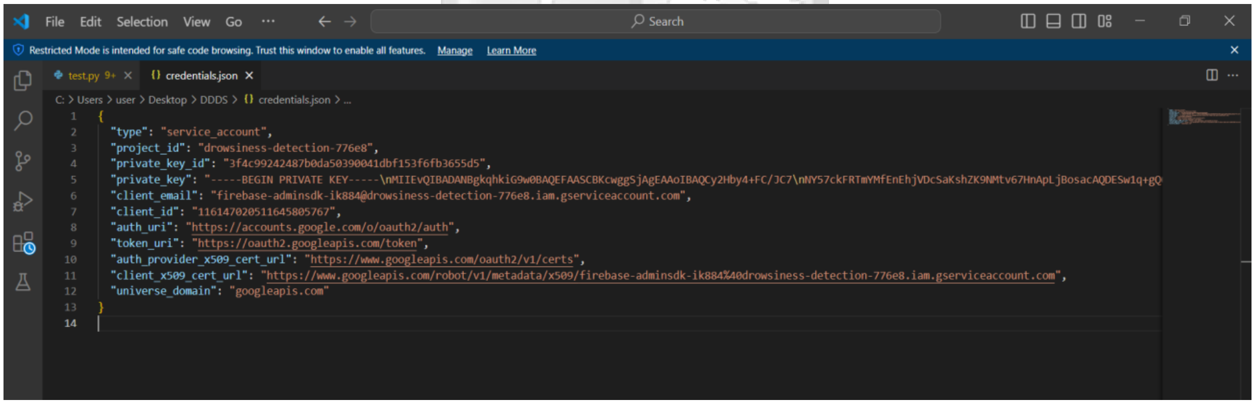


Figure 15 Firebase Credentials

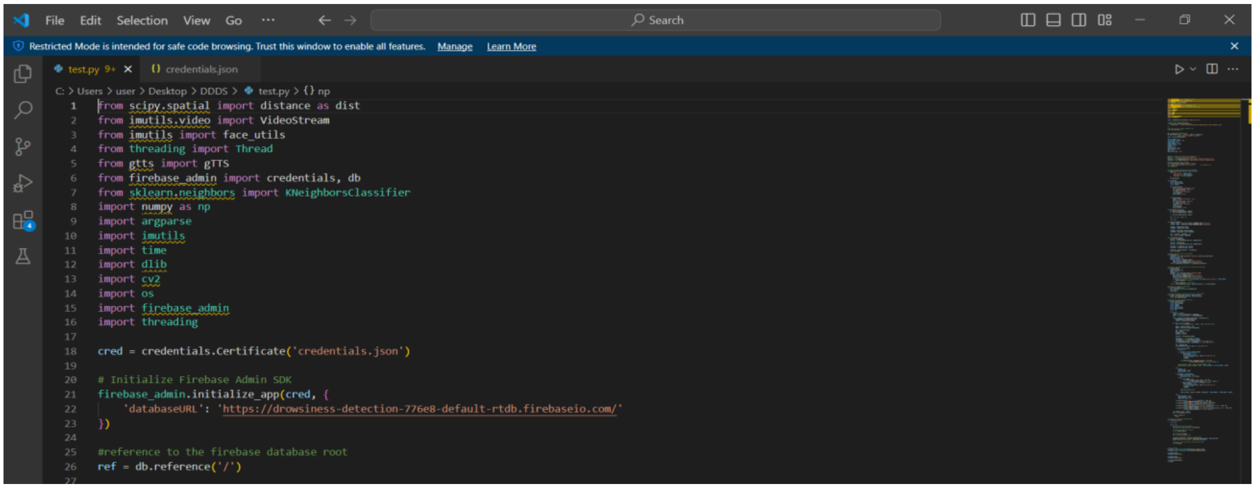


Figure 16 Firebase Dependencies

The data from the sensors is then sent to the Firebase Realtime Database after initialization, facilitating real-time updates and synchronization across connected clients. This is shown well in the figure below.

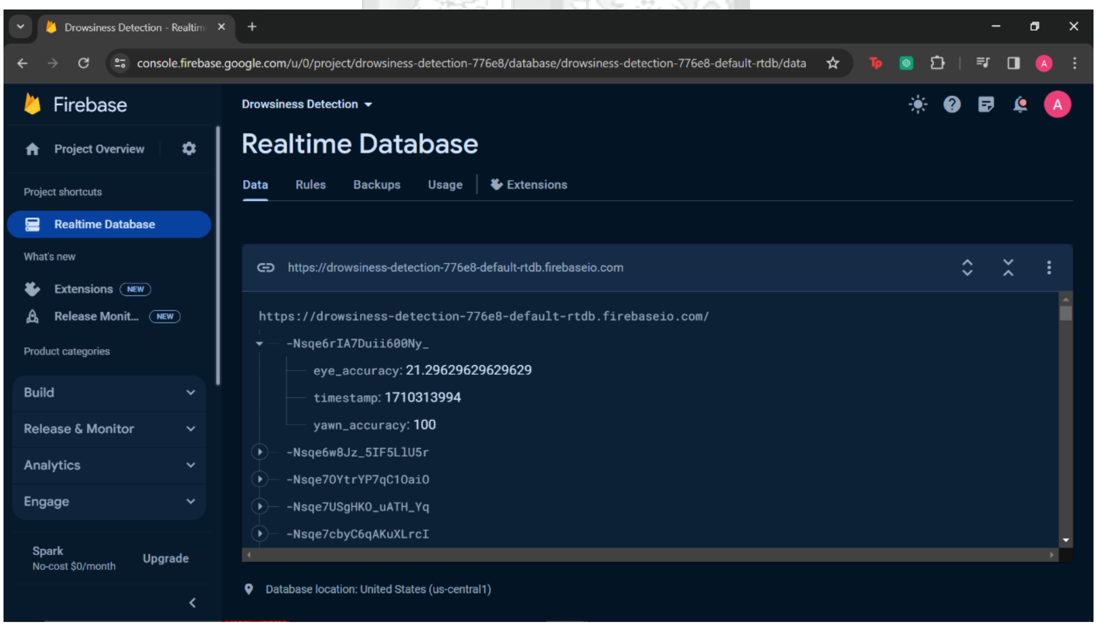


Figure 17 Firebase Real time Database.

5.4 Real time Data processing Implementation

Real-time data processing in the drowsiness detection system involves several key steps to accurately analyze sensor inputs and detect signs of drowsiness in users.

5.4.1 Data Acquisition

The system continuously collects sensor data which are the eye movements and mouth movements from the connected sensors/camera in real-time. This is shown in the figure below.

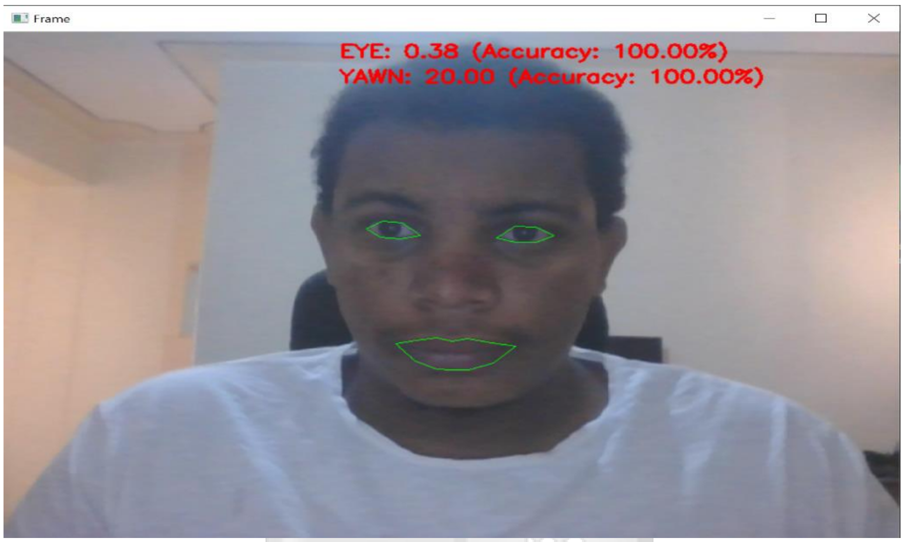


Figure 18 Collecting Data in Real Time through a Video Stream

5.4.2 Data Preprocessing and Feature Extraction

In the data preprocessing and feature extraction stage of the drowsiness detection system, Haar cascading and facial landmark detection played crucial roles in extracting relevant features from the input sensor data, which was obtained from cameras capturing images of the user's face. Here's an explanation of how these techniques is applied:

5.4.2.1 Haar Cascading

Haar cascading is a machine learning-based technique used for object detection in images (Jaiswal, 2023). It involves training a classifier to identify specific patterns, known as Haar-like features, that are characteristic of the object of interest. A Haar feature is essentially calculations that are performed on adjacent rectangular regions at a specific location in a detection window. The calculation involves summing the pixel intensities in each region and calculating the differences between the sums (Mittal, 2024). Here are some examples of Haar features below.

VT OMNES VNVM SINT

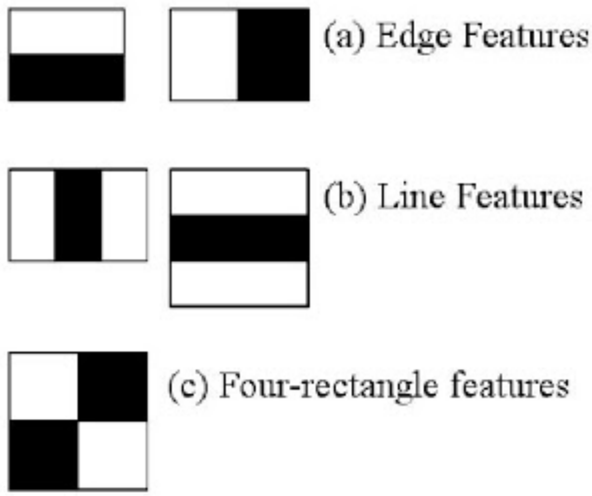


Figure 19 Types of Haar Features (Mittal, 2024).

These features within the image simplify the detection of edges or lines or identify regions with abrupt changes in pixel intensities. This is shown in the figure below.

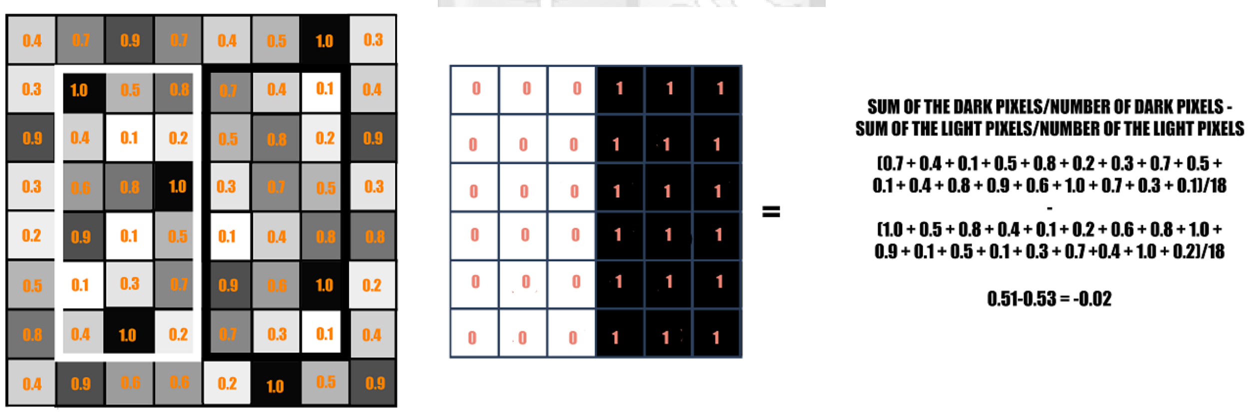


Figure 20 Haar Calculation Values (Behera, 2020).

In the figure above, The left rectangle represents an image with pixel values ranging from 0.0 to 1.0. The center rectangle depicts a Haar kernel, where light pixels are situated on the left and dark pixels on the right. The Haar calculation involves determining the difference between the averages of pixel values in the darker and lighter regions. If this difference is close to 1, an edge is detected by the Haar feature (Behera, 2020).

5.4.2.2 Facial Landmark Detection

Once the face region is identified using Haar cascading, facial landmark detection techniques are applied to localize the key facial landmarks which are the eyes and mouth of the user. The method used for the facial landmark detection involved the use of a shape predictor algorithm which is the dlib library's facial landmark predictor. The dlib library is employed to estimate the coordinates (x, y) of 68 facial points, mapping various facial features on a person's face as illustrated in the image below (Jose, 2021).

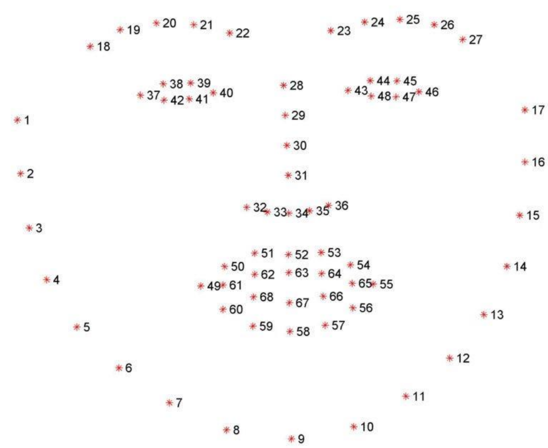


Figure 21 Dlib Facial Landmark (Jose, 2021).

5.4.2.3 Facial Feature Extraction

With the facial landmarks localized, various features related to drowsiness detection are then computed from their coordinates. For example:

VT OMNES VNVM SINT

Eye Aspect Ratio (EAR): Eye Aspect Ratio (EAR) is calculated from the distances between the inner and outer corners of the eyes to detect eye closure.

From Figure 21. above, you can notice that each eye is represented by using six (6) landmark points. The Eye Aspect Ratio for a single eye is hence calculated using the formula below (Pandey, 2021).

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 22 Eye Aspect Ratio formula (Pandey, 2021).

Where an open eye will have more EAR and a closed eye will have less EAR.

Figure 23 Open and Closed Eye EAR (Pandey, 2021).

Mouth Aspect Ratio (MAR): Mouth Aspect Ratio (MAR) is computed from the distances between the corners of the mouth to detect changes in mouth shape associated with yawning or drowsiness. This is achieved by calculating the distance between the top lip and the bottom lip.

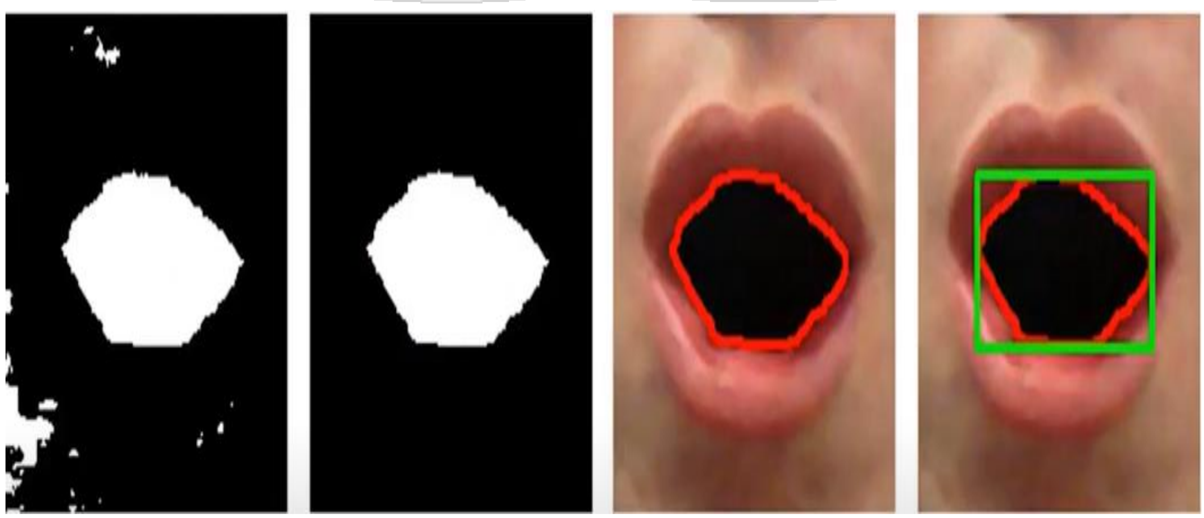


Figure 24 Mouth Aspect Ratio

These extracted features serve as inputs to the machine learning algorithms that will be used to classify the user's alertness level and trigger alerts when signs of drowsiness are detected.

5.4.3 Algorithm Implementation

After the feature extraction stage, we then set to analyze these features and classify them based on the user’s drowsiness levels. By leveraging OpenCV’s library, I utilized the K-Nearest Neighbors (KNN) algorithm as one of the primary classification techniques. During the training phase, I assigned labels to the collected data based on predefined thresholds. For instance, if the Eye Aspect Ratio (EAR) is greater than or equal to the threshold, I labeled it as one (1), indicating drowsiness; otherwise, I labelled it as (0), indicating not drowsy. With this labeled dataset, I trained the KNN algorithm, where each example represented a set of features extracted from sensor data alongside the corresponding to the drowsiness levels of the user. As I collect and process sensor data in real-time, I deployed the trained KNN classifier to make instant predictions by comparing feature vectors of new instances to those of training examples, I was able to determine whether the user is drowsy or not drowsy based on the calculations from the EAR and MAR formulas. The diagram below shows examples of screens that show whether the user is drowsy or not.

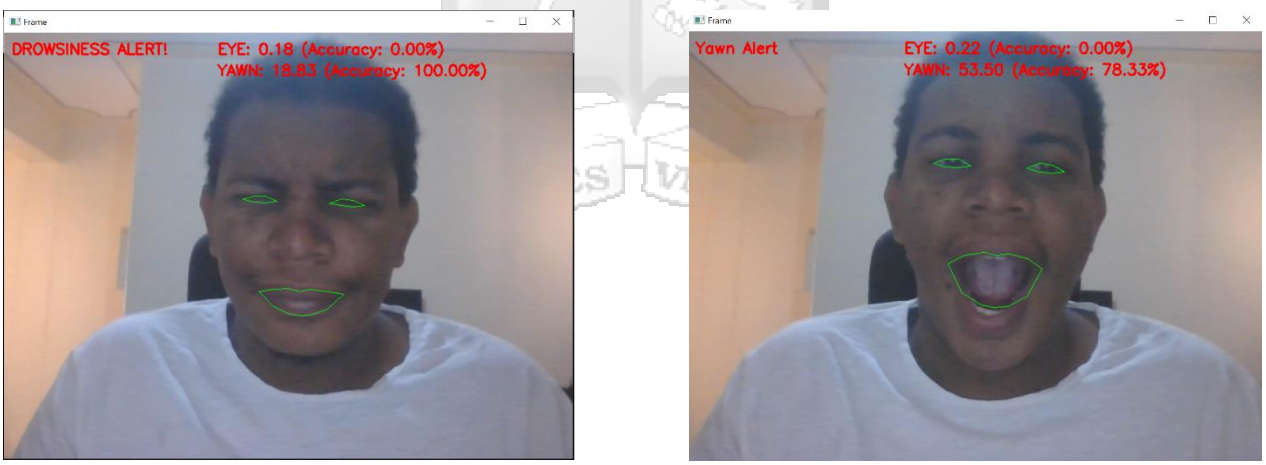


Figure 25 Drowsy or Not Drowsy Frame Examples

5.5 System Testing

Following the integration of all system components, a comprehensive evaluation was conducted to ensure seamless functionality. Throughout the development stages, repeated rounds of functional testing were undertaken to validate the fulfillment of all requirements and guarantee smooth system operation. Moreover, this testing procedure rigorously addressed and resolved any identified bugs or issues to ensure optimal performance and reliability of the system.

5.5.1 Functional Requirements

The table below shows the testing functional requirements.

Test Cases	Pre-Conditions	Test Data	Expected Results	Pass/Fail
Sensor/Camera Calibration	Sensors are calibrated connected and calibrated correctly	Camera readings	Accurate Camera readings	Pass
Face Detection	Video stream contains faces	Video Frames	Correct localization of faces	Pass
Facial Landmark Detection	Detected faces are correctly localized	Facial Images	Correct identification of facial landmarks	Pass
Feature Extraction	Facial Landmark detection is successful	Extracted facial features	Eye and mouth features are extracted successfully	Pass
Classification Accuracy	Training data is properly labeled	Feature Vectors and corresponding labels	Accurate classification of drowsiness levels	Pass
Real time processing	Continuous stream of sensor data is available	Real time sensor data	Timely prediction of drowsiness levels	Pass
Robustness Testing	Variations in lighting conditions and	Diverse test data	Consistent performance under different	Pass

VT OMNES VNVM SINT

	user demographics		conditions	
Integration Testing	All systems components are properly integrated	Combined system functionality	Seamless interaction between components	Pass
Error Handling	Unexpected errors or anomalies occur	Erroneous inputs	Graceful handling of errors and anomalies	Pass

5.5.2 Compatibility Testing

This test was done to verify the accessibility of the system across different operating systems. The tested operating systems include Windows and Ubuntu Linux. The table below illustrates this.

Operating System	Compatibility
Windows 10 Pro Version 22H2	YES
Linux Ubuntu	YES

5.5.3 Validation Testing

The drowsiness detection system was validated through a rigorous testing and analysis to guarantee that it satisfied the requirements. The validation was conducted to ensure that there was accuracy and dependability of data collected by the system. This validation was to gain assurance that the drowsiness detection system data is accurate and will contribute to a more secure and efficient method of ensuring road safety.

CHAPTER 6: DISCUSSION OF RESULTS

6.1 Introduction

In this chapter, we delve into a comprehensive analysis of the performance results obtained from evaluating our drowsiness detection system. Utilizing a dataset comprising 62 two-second videos collected from the system. Through rigorous examination, we obtained a breakdown of 28 true positives (TP), 19 true negatives (TN), 7 false positives (FP), and 8 false negatives (FN). These performance metrics serve as crucial indicators of the system's efficiency in accurately identifying and mitigating instances of driver drowsiness.

Furthermore, we delve into the scenarios employed to evaluate the measures, encompassing diverse real-world situations that drivers may encounter. These scenarios include instances of:

- True positives where the system correctly detects the driver's drowsiness, triggering an alert to prevent potential accidents.
- False positives, the system incorrectly identifies the driver as drowsy, resulting in unnecessary alerts and disruptions.
- True negatives represent scenarios where the system correctly identified the driver as alert, avoiding unnecessary alerts and ensuring smooth operation.
- False negatives depict instances where the system fails to detect the driver's drowsiness, leading to a missed opportunity to intervene and prevent potential accidents.

6.2 Results Of The Study

Through an in-depth analysis of key performance metrics including accuracy, recall, precision, and F1 score, we aim to gain valuable insights into the system's strengths, weaknesses, and overall effectiveness. By interpreting these results in the context of real-world scenarios, we seek to inform decision-making processes and identify potential areas for improvement, ultimately contributing to the enhancement of road safety and the prevention of accidents caused by driver fatigue.

6.2.1 Accuracy Evaluation

Accuracy is a metric that measures how often a machine learning model correctly predicts the outcome.

VT OMNES VNVM SINT

Formula:

$$Accuracy = \frac{True\ Positive + True\ Negatives}{Total}$$

Calculation:

Given the values provided:

- True Positives (TP) = 28
- True Negatives (TN) = 19
- False Positives (FP) = 7
- False Negatives (FN) = 8
- Total = TP + TN + FP + FN = 28 + 19 + 7 + 8 = 62

Using the formula:

$$Accuracy = \frac{28+19}{62} = 0.758$$

Interpretation:

The calculated accuracy value of approximately 0.758 indicates that the drowsiness detection system achieved an accuracy rate of approximately 75.8%. This means that about 75.8% of the instances in the dataset were correctly classified by the system as either drowsy or not drowsy.

6.2.2 Recall Assessment

Recall, also known as sensitivity, quantifies the frequency with which a machine learning model accurately identifies positive instances (true positives) out of all the genuine positive samples present in the dataset.

Formula:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Calculations:

Given the values provided:

- True Positives (TP) = 28

- False Negatives (FN) = 8

Using the formula:

$$Recall = \frac{28}{28 + 8} = 0.778$$

Interpretation:

The calculated recall value of approximately 0.778 indicates that the drowsiness detection system achieved a recall rate of approximately 77.8%. This means that about 77.8% of all instances of actual drowsiness in the dataset were correctly identified by the system as drowsy. A higher recall means that the model is good at identifying correct data.

6.2.3 Precision Analysis

Precision measures the accuracy of positive predictions made by a classification model. It evaluates the proportion of correctly identified positive instances (true positives) out of all instances predicted as positive by the model. Precision provides insights into the model's ability to avoid false positive predictions.

Formula:

$$Precission = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Calculation:

Given the values provided:

- True Positives (TP) = 28
- False Positives (FP) = 7

Using the formula:

$$Precision = \frac{28}{28 + 7} = 0.800$$

Interpretation

The calculated precision value of approximately 0.800 indicates that the drowsiness detection system achieved a precision rate of approximately 80.0%. This means that about 80.0% of the instances predicted as drowsy by the system were genuinely drowsy.

6.2.4 F1 Score Evaluation

The F1 score is a harmonic average of precision and recall, providing a balanced measure of a classification model's performance.

Formula:

$$\text{F1 Score} = 2 * \frac{Precission * Recall}{Precission + Recall}$$

Calculation:

Given the values provided:

- Precision = 0.800
- Recall = 0.778

Using the formula

$$\text{F1 Score} = 2 * \frac{0.800 * 0.778}{0.800 + 0.778} = 0.7886$$

Interpretation:

The calculated F1 score of approximately 0.7886 indicates that the drowsiness detection system achieved a balanced performance in terms of precision and recall. This value reflects the harmonic mean of precision and recall, providing an overall measure of the system's effectiveness in identifying and mitigating instances of driver drowsiness.

6.3 Validation of the System

This task's main aim was to determine if the system met its proposed performance and how effective it was in collecting data and analyzing it to detect drowsiness during driving. The study involved coming up with a prototype where real time data was collected from sensors and passed through a machine learning model for analysis. The results of the study showed that this drowsiness detection system could be an effective and appropriate method of preventing road accidents.

VT OMNES VNVM SINT

CHAPTER 7: CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORKS

7.1 Conclusions

After this study, several conclusions were drawn as follows:

1. **Effectiveness in drowsiness detection** – The system demonstrated a promising effectiveness in identifying instances of driver drowsiness, as evidenced by high recall and precision rates. This suggested that the system can accurately detect drowsiness and issue timely alerts to prevent potential accidents on the road.
2. **Reliability in real world scenarios** – Through testing, the system exhibited great efficiency and robustness, indicating its reliability in diverse environments and lighting conditions. This reliability is crucial for ensuring the system's practical utility and effectiveness in real-world applications.
3. **Potential for road safety enhancement** – the findings of this study suggest that deploying this drowsiness detection system has the potential to significantly enhance road safety by preventing the issues arising from driver fatigue which leads to drowsiness. This is achieved through providing timely alerts, therefore reducing the likelihood of accidents related to drowsiness.
4. **Areas for Optimization and Enhancement** – despite the systems effectiveness, this study also identified potential areas for optimization and enhancement including fine tuning of algorithm parameters, refinement of data preprocessing techniques, and validation of drowsiness scenarios. This enabled the system to enhance its performance and usability which ensure its effectiveness in the real world.

7.2 Recommendations

After this study, the subsequent recommendations were drawn:

1. Car Manufacturing companies should embrace the use of drowsiness detection systems especially using IoT and Machine learning ones, because the infrastructure already exists. This will enable real time collection of data, optimize operations, and enhance road safety for drivers.

2. The government should put up policies to push for the introduction and adoption of drowsiness detection systems to reduce the number of road accidents related to drowsiness.
3. Drivers should be involved in the deployment process and should also be trained on the benefits of this technology, its efficiency, and its transparency.
4. Further research should be conducted on drowsiness detecting algorithms and how this can be adopted in not only urban cities but also villages to improve the overall sustainability of both urban and rural areas.

7.3 Future Works

Drowsiness detections systems with IoT and Machine Learning promises technology that can help improve and enhance road safety. Work under drowsiness detection systems is expected to evolve in several directions in the future.

1. Advanced sensor technology – investigation of advanced sensor technologies such as multimodal sensors for example heart rate monitors and EEG sensors, will be able to enhance the accuracy and reliability of drowsiness detection. This can be done by attaching these sensors on the steering wheelbase in order to collect the physiological information of the driver in real time.
2. Data Analysis – Incorporation of data analysis since more data will be collected by the drowsiness detection sensors. This will enable data scientists to use data analytic techniques to have more insight into drowsiness detection and areas of improvement.
3. Contextual awareness – This is enhancing the systems contextual awareness by integrating environmental factors, road conditions, traffic patterns and driver context into the drowsiness detection process. This will be achieved by developing adaptive algorithms that dynamically adjust alert thresholds and response strategies based on contextual information to reduce the number of false alarms which will further improve the user acceptance rates.
4. Diversified Dataset - Future implementations of the drowsiness detection system should prioritize the incorporation of a diversified dataset. This entails gathering data from a wide array of real-world scenarios, including diverse demographics, environmental conditions, driving contexts, and cultural factors. By incorporating such diversity into the

dataset, the system can better generalize across different driving conditions and populations, leading to improved performance, reduced bias, and increased real-world relevance.

REFERENCES

Ahlstrom, C., Forsman, P. M., Nordbakke, S., Patel, M., Thiffault, P., Radun, I., Yang, G., Guosheng, Y., Guo, Z., & Jens, B. (2018, July 23). *Drowsiness monitoring based on steering wheel status*. Transportation Research Part D: Transport and Environment. https://www.sciencedirect.com/science/article/abs/pii/S1361920917306582

Alajlan, N. N., & Ibrahim, D. M. (2023, June 18). *DDD tinyml: A tinyml-based driver drowsiness detection model using Deep Learning*. MDPI. https://www.mdpi.com/1424-8220/23/12/5696#:~:text=The%20study%20in%20%5B38%5D%20proposed,used%20to%20notify%20him%2Fher.

Albadawi, Y., Takruri, M., & Awad, M. (2022, March 7). *A review of recent developments in driver drowsiness detection systems*. MDPI. https://www.mdpi.com/1424-8220/22/5/2069

Alioua, N., Amine, A., & Rziza, M. (2014, August 6). *Driver's fatigue detection based on yawning extraction*. International Journal of Vehicular Technology. https://www.hindawi.com/journals/ijvt/2014/678786/

Amidei, A., Spinsante, S., Iadarola, G., Benatti, S., Tramarin, F., Pavan, P., & Rovati, L. (2023, April 15). *Driver drowsiness detection: A machine learning approach on skin conductance*. Sensors (Basel, Switzerland). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10143251/

Bajaj, J. S., Kumar, N., Kaushal, R. K., Gururaj, H. L., Flammini, F., & Natarajan, R. (2023, January 23). *System and method for driver drowsiness detection using behavioral and sensor-based physiological measures*. MDPI. https://www.mdpi.com/1424-8220/23/3/1292#B8-sensors-23-01292

Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazel, A., Holleman, J., Huang, X., Hurtado, R., Kanter, D., Lokhmotov, A., Patterson, D., Pau, D., Seo, J., Sieracki, J., Thakker, U., Verhelst, M., & Yadav, P. (2021, January 29). *Benchmarking tinyml systems: Challenges and direction*. arXiv.org. https://arxiv.org/abs/2003.04821

Behera, G. S. (2020, December 29). *Face detection with Haar Cascade*. Medium. https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08

Bhandari, P. (2023, June 21). *Population vs. sample: Definitions, Differences & Examples*. Scribbr. https://www.scribbr.com/methodology/population-vs-sample/#:~:text=A%20population%20is%20the%20entire,t%20always%20refer%20to%20people.

Bhat, A. (2023, September 12). *Research Design: What it is, Elements & Types*. QuestionPro. https://www.questionpro.com/blog/research-design/

Borkar, A. (2015, January 1). *Real time eye blinking detection and tracking using opencv*. Academia.edu. https://www.academia.edu/74455089/Real_Time_Eye_Blinking_Detection_and_Tracking_Using_Opencv

Bouchrika, I. (2023, May 15). *How to Write Research Methodology: Overview, Tips, and Techniques*. Research.com. https://research.com/research/how-to-write-research-methodology#:~:text=Determining%20the%20validity%20of%20your,that%20support%20your%20research%20approach.

Bruin, I. (2010). Exploring how objects used in a picture vocabulary test influence validity: Chapter three. University of Pretoria. Retrieved from https://repository.up.ac.za/bitstream/handle/2263/25218/00front.pdf.

Burns, P. (2023, May 19). *How night driving contributes to Drowsiness*. Burns, Cunningham & Mackey, P.C. https://www.bcmlawyers.com/how-night-driving-contributes-to-drowsiness/#:~:text=The%20Dangers%20of%20Nighttime%20Driving&text=As%20it%20gets%20darker%20outside,this%20time%20is%20often%20counterproductive.

CEIC DATA. (2021). *Kenya number of registered vehicles*. CEIC. https://www.ceicdata.com/en/indicator/kenya/number-of-registered-vehicles

Daly, N. (2023, May 4). *What is a use case & how to write one: Wrike*. Blog Wrike. https://www.wrike.com/blog/what-is-a-use-case/

Doudou, M., Bouabdallah, A., & Berge-Cherfaoui, V. (2019, September 12). *Driver drowsiness measurement technologies: Current research, Market Solutions, and Challenges - International Journal of Intelligent Transportation Systems Research*. SpringerLink. https://link.springer.com/article/10.1007/s13177-019-00199-w

Fernández, F. G., Chimeno, M. F., Castro, J. R., & González, M. A. G. (2019, June 24). *Driver drowsiness detection based on respiratory signal analysis*. IEEE Xplore. https://ieeexplore.ieee.org/document/8744224

Fujirawa, K., Abe, E., Kamata, K., Nakayama, C., Suzuki, Y., Yamakawa, T., Hiaraoka, T., Kano, M., Sumi, Y., Masuda, F., Matuso, M., & Katodani, H. (2019, June). *Heart rate variability-based driver drowsiness detection and its validation with EEG*. IEEE transactions on bio-medical engineering. https://pubmed.ncbi.nlm.nih.gov/30403616/

Gianfranchi, E., Cellini, N., Patron, E., Tagliabue, M., & Sarlo, M. (2019, July 23). *Stressing the accuracy: Wrist-worn wearable sensor validation over different condition*. Wiley Online Library. https://onlinelibrary.wiley.com/doi/10.1111/psyp.13441

Gillis, A. S. (2022, August 30). *What is Product Development? Definition and Examples*. CIO. https://www.techtarget.com/searchcio/definition/product-development-or-new-product-development-NPD

Haupt, D., Honzik, P., Raso, P., & Hyncica, O. (2011). *Steering wheel motion analysis for detection of the driver's drowsiness*. Researchgate.net. https://www.researchgate.net/profile/Daniel-Haupt/publication/261849547_Steering_wheel_motion_analysis_for_detection_of_the_driver%27s_drowsiness/links/5566af1308aeccd77735a6ed/Steering-wheel-motion-analysis-for-detection-of-the-drivers-drowsiness.pdf

Hussein, M. K., Salman, T. M., Miry, A. H., & Subhi, M. A. (2021). Driver Drowsiness Detection Techniques: A Survey. In 2021 1st Babylon International Conference on Information Technology and Science (BICITS) (pp. 45-51). Babil, Iraq. doi: 10.1109/BICITS51482.2021.9509912.

Jabbar, R., Khalifa, K. A., Kharbeche, M., Alhajyaseen, W., Jafari, M., & Jiang, S. (2018, April 24). *Real-time driver drowsiness detection for Android application using Deep Neural Networks Techniques*. Procedia Computer Science. https://www.sciencedirect.com/science/article/pii/S1877050918304137

Jaiswal, A. (2023, August 10). *Guide to haar cascade algorithm with object detection example*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/

Jin. (2022, May 3). *System design-UML class diagram*. Medium. https://interviewnoodle.com/system-design-uml-class-diagram-c69dbade0854

Johns Hopkins Medicine. (2023, April 24). *Electromyography (EMG)*. https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/electromyography-emg

Jose, I. (2021, June 2). *Facial mapping (landmarks) with dlib + python*. Medium. https://towardsdatascience.com/facial-mapping-landmarks-with-dlib-python-160abcf7d672

LaRocco, J., Le, M. D., & Paeng, D.-G. (2020, August 24). *A systemic review of available low-cost EEG headsets used for drowsiness detection*. Frontiers. https://www.frontiersin.org/articles/10.3389/fninf.2020.553352/full?ref=https%3A%2F%2Fgithubhelp.com

Levin, D. (2007). *Volvo Cars introduces new systems for alerting tired and distracted drivers*. Volvo Car USA Newsroom. https://www.media.volvocars.com/us/en-us/media/pressreleases/12130

Li, Z., Li, S. E., Li, R., Cheng, B., & Shi, J. (2017, March 2). *Online detection of driver fatigue using steering wheel angles for real driving conditions*. MDPI. https://www.mdpi.com/1424-8220/17/3/495

Lithmee. (2018, April 3). *Difference between SDLC and agile methodology*. Compare the Difference Between Similar Terms. https://www.differencebetween.com/difference-between-sdlc-and-vs-agile-methodology/

Macharia, W., Njeru, E., Muli-Musiime, F., & Nantulya, V. (2009). *Severe road traffic injuries in Kenya, quality of care and access*. African Health Sciences. https://www.ajol.info/index.php/ahs/article/view/43772

Mboga, J. (2020). *Fatigue kills more than drink driving, safety expert says*. The Standard. https://www.standardmedia.co.ke/business/nairobi/article/2001354215/fatigue-kills-more-than-drink-driving-safety-expert-says

Min, J., Wang, P., & Hu, J. (2017). *Driver fatigue detection through multiple entropy fusion analysis in an EEG-based system*. PLOS ONE. https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0188756

Mittal, A. (2024, January 18). *Haar Cascades, explained*. Medium. https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d

Murch, R. (2012). *The Software Development Lifecycle-A Complete Guide*. Richard Murch.

Nauto. (2022). *What is Drowsy Driving? - definition*. https://www.nauto.com/glossary/what-is-drowsy-driving

Nolle, T. (2021, October 18). *What is a data flow diagram (DFD)?*. Data Management. https://www.techtarget.com/searchdatamanagement/definition/data-flow-diagram-DFD

Patel, K. K., Patel, S. M., & Scholar, P. (2016). Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. International journal of engineering science and computing, 6(5).

Paul C. Price, R. S. J. (2017, August 21). *RELIABILITY AND VALIDITY OF MEASUREMENT*. Research methods in psychology. https://opentext.wsu.edu/carriecuttler/chapter/reliability-and-validity-of-measurement/

Pandey, D. (2021, April 21). *Eye aspect ratio(ear) and drowsiness detector using dlib*. Medium. https://medium.com/analytics-vidhya/eye-aspect-ratio-ear-and-drowsiness-detector-using-dlib-a0b2c292d706

Physiopedia. (2023). *Types of Research*. https://www.physio-pedia.com/Types_of_Research

Purnamasari, P. D., Kriswoyo, A., Ratna, A. A. P., & Sudiana, D. (2021, October 19). *Eye based drow*siness detection system for driver - journal of electrical engineering & technology. SpringerLink. https://link.springer.com/article/10.1007/s42835-021-00925-z

Royce, W. W. (2021b, February 2). Managing the development of Large Software Systems (1970). MIT Press. https://direct.mit.edu/books/edited-volume/5003/chapter-abstract/2657056/Managing-the-Development-of-Large-Software-Systems?redirectedFrom=PDF

Safarov, F., Akhmedov, F., Abdusalomov, A. B., Nasimov, R., & Cho, Y. I. (2023a, July 17). *Real-time deep learning-based drowsiness detection: Leveraging computer-vision and eye-blink analyses for Enhanced Road Safety*. MDPI. https://www.mdpi.com/1424-8220/23/14/6459

Safarov, F., Akhmedov, F., Abdusalomov, A. B., Nasimov, R., & Cho, Y. I. (2023b, July 17). *Real-time deep learning-based drowsiness detection: Leveraging computer-vision and eye-blink analyses for Enhanced Road Safety*. MDPI. https://www.mdpi.com/1424-8220/23/14/6459#:~:text=There%20are%20several%20factors%20underlying,driving%20%5B1%2C2%5D.

Sapkota, D., Bista, B., & Adhikari, S. R. (2016, November 29). *Economic costs associated with motorbike accidents in Kathmandu, Nepal*. Frontiers. https://www.frontiersin.org/articles/10.3389/fpubh.2016.00273/full

Satti, A. T., Kim, J., Yi, E., Cho, H.-Y., & Cho, S. (2021, July 27). *Microneedle array electrode-based wearable EMG system for detection of driver drowsiness through steering wheel grip*. Sensors (Basel, Switzerland). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8347525/#:~:text=The%20muscle%20activity%20signal's%20magnitude,drop%20and%20alerts%20the%20driver.

Shaffer, F., & Ginsberg, J. P. (2017, September 28). *An overview of heart rate variability metrics and norms*. Frontiers in public health. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5624990/

Singh, S. (2018, December 25). *Sampling techniques*. Medium. https://towardsdatascience.com/sampling-techniques-a4e34111d808

Smith, D. (2003). *Five Principles for Research Ethics*. Monitor on Psychology. https://www.apa.org/monitor/jan03/principles

Suni, E., & Rehman, A. (2023, June 1). *Drowsy driving: Dangers and how to avoid it*. Sleep Foundation. https://www.sleepfoundation.org/drowsy-driving

Teyeb, I., Jemai, O., Zaied, M., & Amar, C. B. (2014, January 1). *A drowsy driver detection system based on a new method of head posture estimation*. SpringerLink. https://link.springer.com/chapter/10.1007/978-3-319-10840-7_44

Thoutam, N., Sakhare, M., Jagtap, P., Jagtap, S., & Lingayat, A. (2022). *Time driver-drowsiness detection system using facial features*. International Journal of Creative Thoughts (IJCRT). https://ijcrt.org/papers/IJCRT2201305.pdf

Warden, P., & Situnayake, D. (2019). Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers. O'Reilly Media.

Wessel, R. (2022). *What are driver drowsiness detection systems?: Tomtom newsroom*. TomTom. https://www.tomtom.com/newsroom/explainers-and-insights/driver-drowsiness-detection-systems/

WHO, W. (2018). *Global status report on road safety 2018*. World Health Organization. https://www.who.int/publications/i/item/9789241565684

Xiao, T., & Sun, Y. (2021, October 5). *An intelligent mobile app to detect drowsy driving with Artificial Intelligence*. Computer Science & Information Technology (CS & IT). https://www.academia.edu/55796440/An_Intelligent_Mobile_App_to_Detect_Drowsy_Driving_with_Artificial_Intelligence

Ye, C., Li, W., Li, Z., Maguluri, G., Grimble, J., Bonatt, J., Miske, J., Iftimia, N., Lin, S., & Grimm, M. (2022, September 26). *Smart Steering Sleeve (S3): A non-intrusive and integrative sensing platform for Driver Physiological Monitoring*. MDPI. https://www.mdpi.com/1424-8220/22/19/7296

VT OMNES VNVM SINT

APPENDICES

Appendix A: Code Snippets

```python
from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
from gtts import gTTS
from firebase_admin import credentials, db
from sklearn.neighbors import KNeighborsClassifier
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
import os
import firebase_admin
import threading

cred = credentials.Certificate('credentials.json')

# Initialize Firebase Admin SDK
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://drowsiness-detection-776e8-default-rtdb.firebaseio.com/'
})

#reference to the firebase database root
ref = db.reference('/')
```

```python
ap = argparse.ArgumentParser()
ap.add_argument("-w", "--webcam", type=int, default=0,
                help="index of webcam on system")
args = vars(ap.parse_args())

EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 30
alarm_status = False
alarm_status2 = False
saying = False
COUNTER = 0
eye_accuracy = 0.0
yawn_accuracy = 0.0
knn = None
yawn_start_time = None



print("-> Loading the predictor and detector...")
#detector = dlib.get_frontal_face_detector()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')

print("-> Starting Video Stream")
vs = VideoStream(src=args["webcam"]).start()
#vs= VideoStream(usePiCamera=True).start()       //For Raspberry Pi
time.sleep(1.0)

```

```
# Function to push accuracy data to Firebase
def push_accuracy(eye_accuracy, yawn_accuracy):
    data = {
        'eye_accuracy': eye_accuracy,
        'yawn_accuracy': yawn_accuracy,
        'timestamp': int(time.time())
    }
    ref.push(data)

def alarm(msg):
    global alarm_status
    global alarm_status2
    global saying

    while alarm_status:
        print('Predicted Drowsiness: [1]')
        tts = gTTS(text=msg, lang='en')
        tts.save("temp.mp3")
        os.system("start temp.mp3")
        time.sleep(10)
        alarm_status = False


    if alarm_status2:
        print('Predicted Drowsiness: [1]')
        saying = True
        tts = gTTS(text=msg, lang='en')
        tts.save("temp.mp3")
        os.system("start temp.mp3")
```

```
        saying = False
        time.sleep(10)
        alarm_status2 = False

def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

    C = dist.euclidean(eye[0], eye[3])

    ear = (A + B) / (2.0 * C)

    return ear

def final_ear(shape):
    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

    leftEye = shape[lStart:lEnd]
    rightEye = shape[rStart:rEnd]

    leftEAR = eye_aspect_ratio(leftEye)
    rightEAR = eye_aspect_ratio(rightEye)

    ear = (leftEAR + rightEAR) / 2.0
    return (ear, leftEye, rightEye)
```

```python
def lip_distance(shape):
    top_lip = shape[50:53]
    top_lip = np.concatenate((top_lip, shape[61:64]))

    low_lip = shape[56:59]
    low_lip = np.concatenate((low_lip, shape[65:68]))

    top_mean = np.mean(top_lip, axis=0)
    low_mean = np.mean(low_lip, axis=0)

    distance = abs(top_mean[1] - low_mean[1])
    return distance

# Function to fetch data from Firebase
def fetch_data():
    snapshot = ref.order_by_child('timestamp').limit_to_last(10).get()
    eye_accuracies = []
    yawn_accuracies = []
    for key, value in snapshot.items():
        eye_accuracies.append(value['eye_accuracy'])
        yawn_accuracies.append(value['yawn_accuracy'])
    return np.mean(eye_accuracies), np.mean(yawn_accuracies)
```

```python
# Function to prepare the data for training the k-NN model
def prepare_data():
    eye_accuracies = []
    yawn_accuracies = []
    labels = []
    snapshot = ref.order_by_child('timestamp').get()
    for key, value in snapshot.items():
        eye_accuracies.append(value['eye_accuracy'])
        yawn_accuracies.append(value['yawn_accuracy'])
        # Determine drowsiness based on some criteria
        if value['eye_accuracy'] < EYE_AR_THRESH and value['yawn_accuracy'] > YAWN_THRESH:
            labels.append(1)  # Drowsy
        else:
            labels.append(0)  # Not drowsy
    return np.array([eye_accuracies, yawn_accuracies]).T, np.array(labels)

# Function to train the k-NN model
def train_knn_model(X, y):
    knn = KNeighborsClassifier(n_neighbors=3)
    knn.fit(X, y)
    return knn

# Function to predict drowsiness using the trained model
def predict_drowsiness(eye_accuracy, yawn_accuracy):
    X_new = np.array([[eye_accuracy, yawn_accuracy]])
    return knn.predict(X_new)
```

```
# Function for video stream processing
def process_video_stream():
    global alarm_status
    global alarm_status2
    global saying
    global COUNTER
    global eye_accuracy
    global yawn_accuracy
    global knn

    while True:
        frame = vs.read()
        frame = imutils.resize(frame, width=800)
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        rects = detector.detectMultiScale(gray, scaleFactor=1.1,
            minNeighbors=5, minSize=(30, 30),
            flags=cv2.CASCADE_SCALE_IMAGE)

        for (x, y, w, h) in rects:
            rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))

            shape = predictor(gray, rect)
            shape = face_utils.shape_to_np(shape)

            eye = final_ear(shape)
            ear = eye[0]
            leftEye = eye [1]
            rightEye = eye[2]
```

```python
            distance = lip_distance(shape)

            leftEyeHull = cv2.convexHull(leftEye)
            rightEyeHull = cv2.convexHull(rightEye)
            cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
            cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

            lip = shape[48:60]
            cv2.drawContours(frame, [lip], -1, (0, 255, 0), 1)

            if ear < EYE_AR_THRESH:
                COUNTER += 1

                if COUNTER >= EYE_AR_CONSEC_FRAMES:
                    if alarm_status == False:
                        alarm_status = True
                        t = Thread(target=alarm, args=('Stop and rest',))
                        t.deamon = True
                        t.start()

                    cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

                # eye_accuracy = 100 - (ear / EYE_AR_THRESH * 100)
                eye_accuracy = max(0, min(100, ((ear - EYE_AR_THRESH) / EYE_AR_THRESH * 100)))
```

VT OMNES VNVM SINT

```python
            else:
                COUNTER = 0
                alarm_status = False
                eye_accuracy = 100

            if (distance > YAWN_THRESH):
                if yawn_start_time is None:
                    yawn_start_time = time.time()
                else:
                    if time.time() > 4:
                        cv2.putText(frame, "Yawn Alert", (10, 30),
                                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
                    if alarm_status2 == False and saying == False:
                        alarm_status2 = True
                        t = Thread(target=alarm, args=('Stop and rest',))
                        t.deamon = True
                        t.start()

                    yawn_start_time = None

                    yawn_accuracy = max(0, min(100, ((distance - YAWN_THRESH) / YAWN_THRESH * 100)))

            else:
                alarm_status2 = False
                yawn_accuracy = 100
                yawn_start_time = None
```

```python
            cv2.putText(frame, "EYE: {:.2f}".format(ear), (300, 30),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            cv2.putText(frame, "YAWN: {:.2f}".format(distance), (300, 60),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            cv2.putText(frame, f"EYE: {ear:.2f} (Accuracy: {eye_accuracy:.2f}%)", (300, 30),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            cv2.putText(frame, f"YAWN: {distance:.2f} (Accuracy: {yawn_accuracy:.2f}%)", (300, 60),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

        cv2.imshow("Frame", frame)
        key = cv2.waitKey(1) & 0xFF

        if key == ord("q"):
            break

# Function for Firebase implementation
def firebase_process():

    global knn

    while True:
        # Fetch the data from Firebase
        eye_accuracy, yawn_accuracy = fetch_data()

        # Prepare the data for training
        X, y = prepare_data()
```

```python

            # Train the k-NN model
            knn = train_knn_model(X, y)

            # Predict drowsiness using the trained model
            predicted_drowsiness = predict_drowsiness(eye_accuracy, yawn_accuracy)
            print("Predicted Drowsiness:", predicted_drowsiness)

            # Sleep for some time before fetching data again
            time.sleep(10)


# Start threads
vs_thread = threading.Thread(target=process_video_stream)
firebase_thread = threading.Thread(target=firebase_process)

vs_thread.start()
firebase_thread.start()

vs_thread.join()
firebase_thread.join()

cv2.destroyAllWindows()
vs.stop()
```

Appendix B: Ethics Review Certificate

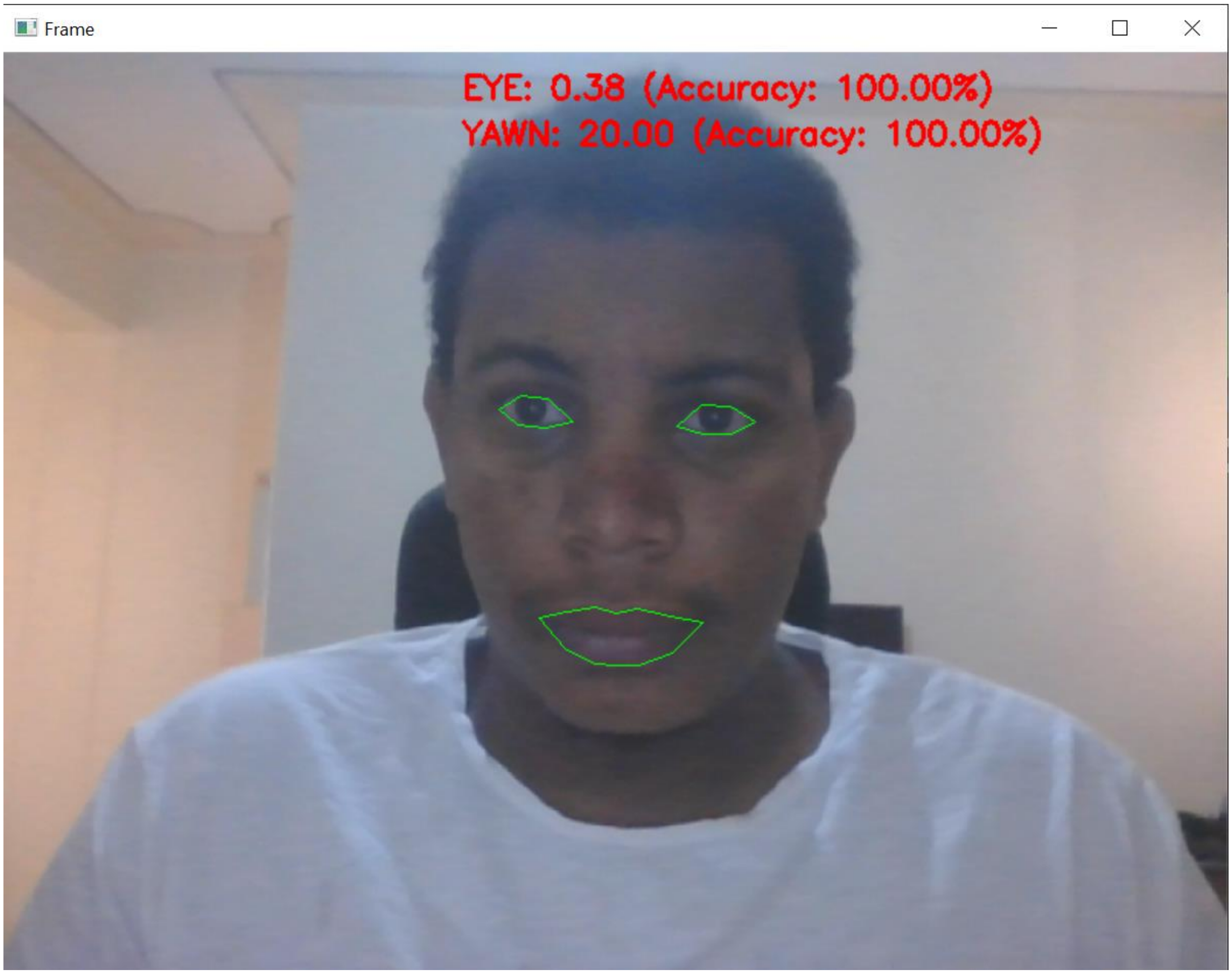
Appendix c: Study Budget

Category	Amount (ksh)	Budget Justification
Project Tools		
1 PC (Core i5 processor with a 2.4GHz Speed, 12GB RAM, and 500 GB Hard Drive)	55500	
TOTAL Project Tools	55500	
Equipment		
Stationery (pens, papers)	500	Ksh. 500
Printing fees	500	Ksh. 500 @ Ksh. 5 per page
TOTAL Equipment	1000	
Other Direct Costs		
Ethical Review Fees	10000	Fees for ethical review certificate
TOTAL OTHER DIRECT COSTS	10000	
TOTAL DIRECT COSTS	66500	
TOTAL PROJECT COST (DIRECT + INDIRECT COST)	66500.00	

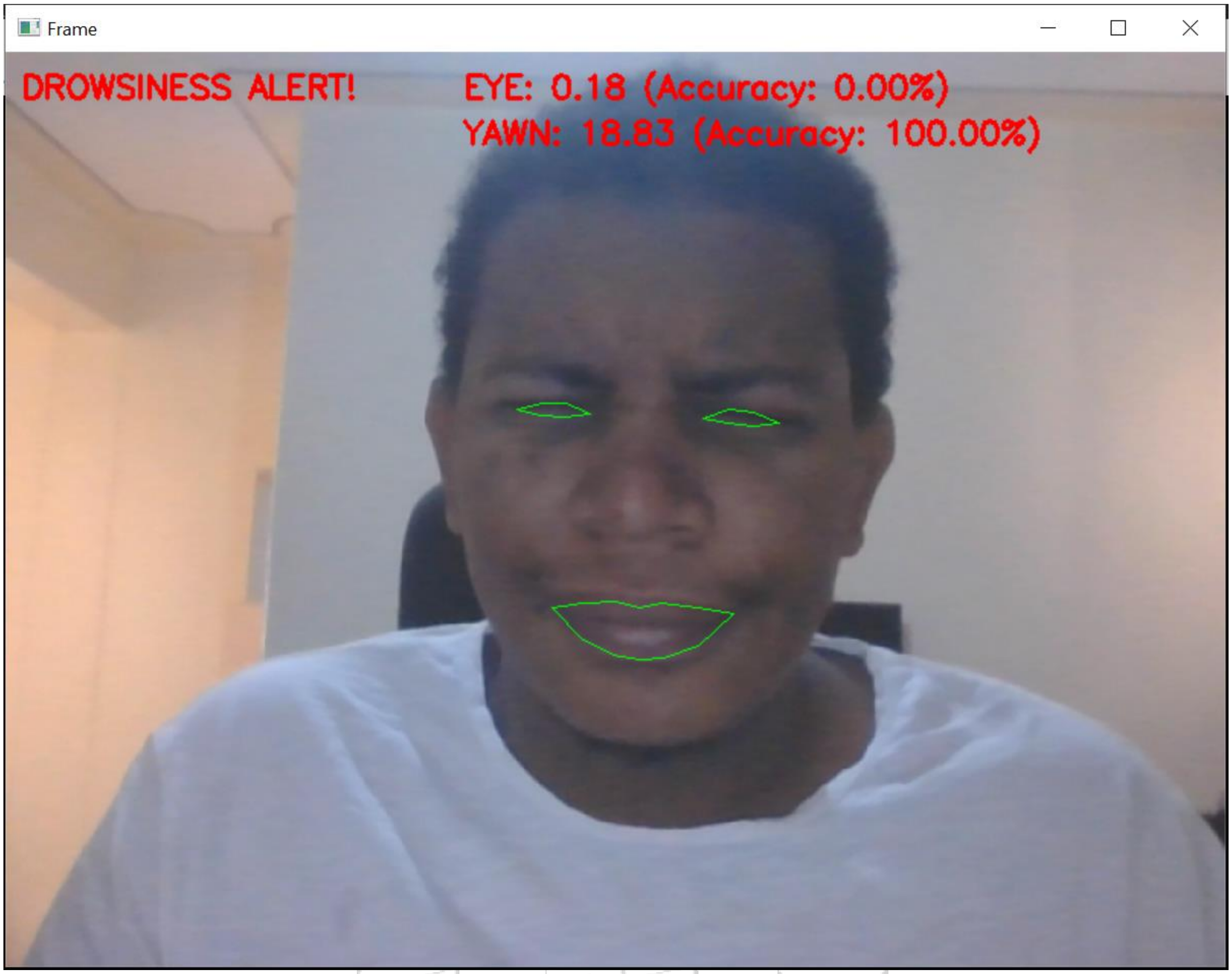
Appendix D: System Implementation Screenshots

This section shows screen shots of the implemented system.

Appendix D.1: Video Stream

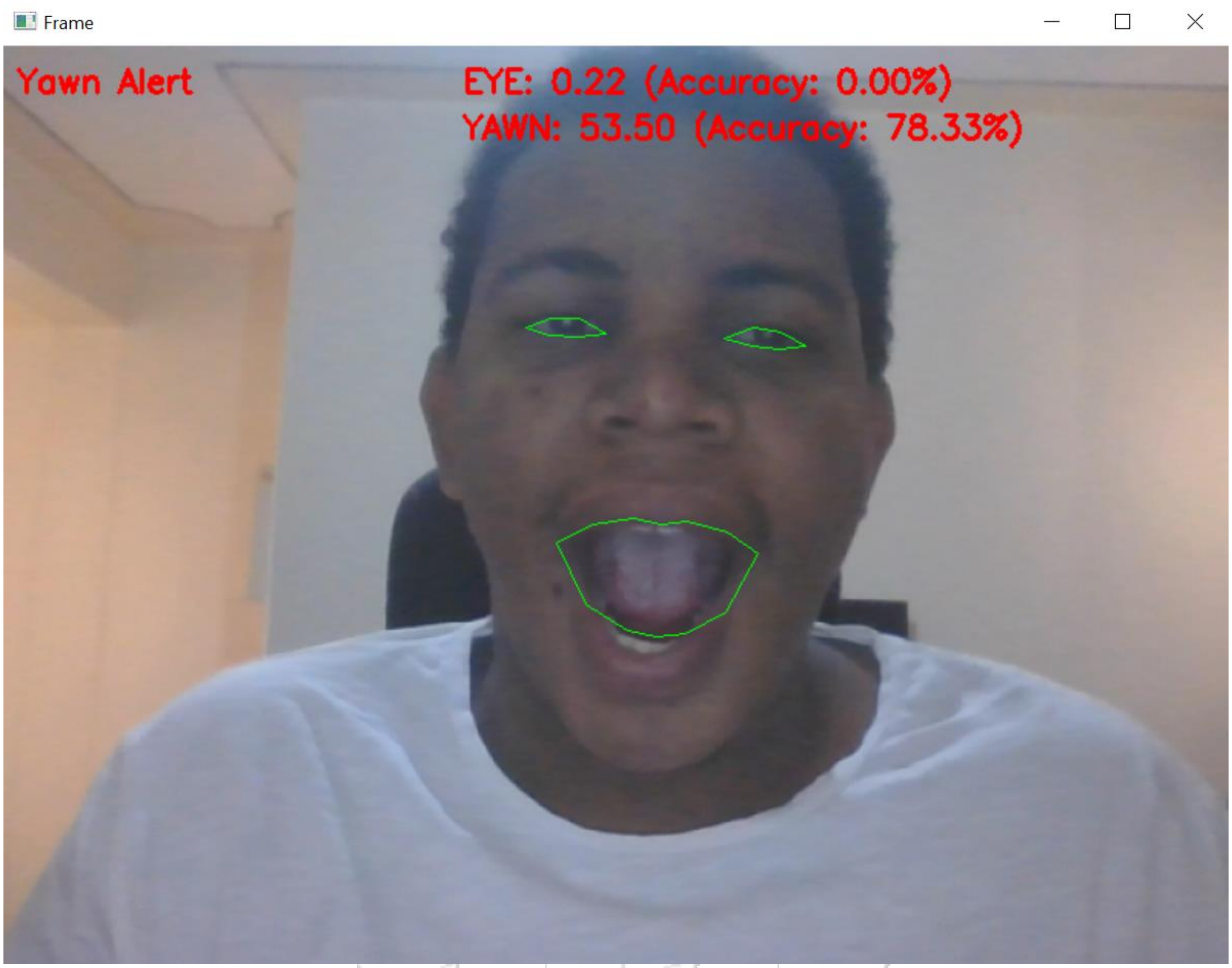


Appendix D.2: Eyes Closed (Drowsiness Alert)



VT OMNES VNVM SINT

Appendix D.3: Mouth Open (Yawn Alert)



VT OMNES VNVM SINT