(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

A Project Report on

IOT-Integrated System for Drowsiness detection using Facial Landmarks

Submitted by,

Aditya Gaikwad	**(Exam Seat No. B238039)**
Yash Gaikwad	**(Exam Seat No. B238040)**
Harshal Tawade	**(Exam Seat No. B234005)**
Prasad Chavan	**(Exam Seat No. B234100)**

Guided by,

Ms. Vinodini Gupta

A Report submitted to MIT Academy of Engineering, Alandi(D), Pune, An Autonomous Institute Affiliated to Savitribai Phule Pune University in fulfillment of the requirements of

BACHELOR OF TECHNOLOGY in

Computer Engineering

School of Computer Engineering

MIT Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Alandi (D), Pune – 412105

(2023–2024)

MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

CERTIFICATE

It is hereby certified that the work which is being presented in the Fourth Year Project Design Report entitled **"IOT-Integrated System for Drowsiness detection using Facial Landmarks"**, in fulfillment of the requirements for the award of the Bachelor of Technology in Computer Engineeringand submitted to the **School of Computer Engineering** of **MIT Academy of Engineering, Alandi(D), Pune, Affiliated to Savitribai Phule Pune University (SPPU), Pune**, is an authentic record of work carried out during Academic Year **2023–2024** Semester **VII**, under the supervision of **Ms. Vinodini Gupta**, **School of Computer Engineering**

Aditya Gaikwad	**(Exam Seat No. B238039)**
Yash Gaikwad	**(Exam Seat No. B238040)**
Harshal Tawade	**(Exam Seat No. B234005)**
Prasad Chavan	**(Exam Seat No. B234100)**

Ms. Vinodini Gupta
Project Advisor

Mr. Pranav Shriram
Project Coordinator

Dr. R.M. Goudar
Dean, SCE

External Examiner

DECLARATION

We the undersigned solemnly declare that the project report is based on our own work carried out during the course of our study under the supervision of **Ms. Vinodini Gupta**.

We assert the statements made and conclusions drawn are an outcome of our project work. We further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this Institute/University or any other Institute/University of India or abroad.
3. We have followed the guidelines provided by the Institute in writing the report.
4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Aditya Gaikwad	**(Exam Seat No. B238039)**
Yash Gaikwad	**(Exam Seat No. B238040)**
Harshal Tawade	**(Exam Seat No. B234005)**
Prasad Chavan	**(Exam Seat No. B234100)**

Abstract

The major aim of this project is to develop a drowsiness detection system by monitoring the eyes; it is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident. In such a case when drowsiness is detected, a warning signal is issued to alert the driver. This detection system provides a noncontact technique for judging different levels of driver alertness and facilitates early detection of a decline in alertness during driving. In such a case when fatigue is detected, a warning signal is issued to alert the driver. The system also has the additional4 feature of slowing down the vehicle if the driver fails to respond to the alarm and ultimately stops the vehicle.

Keywords: driver drowsiness detection, OpenCV, driver fatigue, Raspberry pi

Acknowledgment

We feel much honored in presenting this dissertation report on **"IOT-Integrated System for Drowsiness detection using Facial Landmarks"** in such an authenticable form of sheer endurance and continual efforts of inspiring excellence from various coordinating factors of cooperation and sincere efforts drawn from all sources of knowledge. We express our sincere gratitude to **Ms. Vinodini Gupta** who is the Project guide. She guided us throughout the project and constantly motivated providing suggestions for improvement. We express our sincere gratitude to **Dr. Vaishali Wangikar**, and **Mrs. Prajakta Ugale** who are our Project reviewer for helping us in rectifying our errors and providing suggestions to improve. We are also very thankful to **Mr. Pranav Shriram**, Project coordinator for motivating us throughout the project. We are thankful to **Dr. R. M. Goudar**, SCET Dean for encouraging us. We wish to express our profound gratitude to MITAOE for supporting and providing all the facilities, which would have made it possible for us to complete the dissertation report. We extend our thanks to all our colleagues who have given their full cooperation and valuable suggestions for our dissertation report work.

Aditya Gaikwad
Yash Gaikwad
Harshal Tawade
Prasad Chavan

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

IoT devices are increasingly becoming integral to our daily lives, offering a remarkable feature of generating vast amounts of data. This data holds significant potential for utilization in domains like Artificial Intelligence and Machine Learning. However, it is crucial to address certain concerns, such as cyber-security, that arise with the widespread adoption of IoT. The Internet of Things refers to the interconnection of various objects in our everyday lives, encompassing electronics, software, and sensors, which can access and communicate via the Internet. An important application of IoT is Drowsiness Detection, a safety system found in modern vehicles. This system actively identifies signs of driver drowsiness or fatigue, prompting them to take a break. Prolonged periods of monotonous driving on highways or expressways can lead to exhaustion and a decline in the driver's focus. Numerous studies conducted worldwide have consistently indicated that driver inattention is the primary cause behind approximately 25% of all road accidents.

1.2 Motivation

Driver fatigue is a major contributing factor to the rising number of road accidents in today's society, and its true impact is likely underestimated. Extensive research has shown a strong correlation between driver drowsiness and accidents, emphasizing the urgent need for measures to mitigate the dangers associated with it. The prevalence of drowsiness as a leading cause of severe traffic accidents in our daily lives underscores the importance of developing an effective system that can detect drowsiness and intervene before accidents occur. Prominent automakers such as Toyota, Ford, and Mercedes-Benz are already incorporating advanced safety technologies into their vehicles to prevent accidents caused by driver drowsiness. This progressive trend is anticipated to enhance the intelligence of cars and significantly decrease such accidents. Driven by the compelling statistics on drowsiness-related accidents, our research endeavors to offer an enhanced and systematic approach to detecting drowsiness.

1.3 Project Idea

The designed drowsiness detection and correction system enables rapid identification of sleepiness. It effectively distinguishes between normal eye blinking and tiredness, preventing drivers from getting drowsy while driving, even in low-light conditions. By monitoring the status of the driver's eyes, the system can accurately determine whether they are open or closed. If the eyes remain closed for approximately two seconds, an alarm is triggered, alerting the driver, and the vehicle's speed is reduced. This proactive measure will lead to a decrease in the number of accidents and enhance both driver and vehicle safety. While sleepiness detection systems are typically found only in expensive and luxurious automobiles, the implementation of such a system in regular vehicles can greatly contribute to driver safety.

1.4 Proposed Solution

Every year, approximately 1.3 million lives are lost due to car accidents, making them the leading cause of death. These accidents often occur when drivers become distracted or drowsy. The development of high-speed highways has reduced the room for error for drivers. Every day and night, countless individuals travel long distances on these roads, putting themselves at risk. Fatigue and distractions, such as phone calls or engaging with passengers, greatly increase the chances of an accident. To address this issue, we propose a system that actively alerts drivers when they exhibit signs of distraction or drowsiness. By utilizing image processing techniques on facial landmarks captured through a camera, we can effectively detect and prevent distraction or drowsiness. This innovative system can be easily installed in vehicles, offering a portable solution for enhanced driver safety.

1.5 Project Report Organization (Chapter wise summary)

Introduction chapter consists of various ideas regarding the implementation of the project with a certain background study done. In the Literature review chapter, various IEEE papers have been studied and information is gathered. The future scope and breakdown of the problem statement in simple terms are discussed in the chapter after the literature review. Basic requirements to design the system and methodology used to design the project is mentioned in the following chapters followed by the conclusion of the project.

Chapter 2

Literature Review

2.1 Related work And State of the Art (Latest work)

In a method proposed by (Picot et al., 2012)*Antoine Picot [16]* Drowsiness detection systems employ brain and eye activity. The channel of the electroencephalogram (EEG) used to track brain activity approaches for diagnosis and the EEG-based sleepiness detection uses fuzzy logic. Used blinking detection and characterization to track observed ocular activity Utilizing electrooculography (EOG) channels, blinking characteristics are extracted.
(He et al., 2015) *William Choi et.al. [9]* research suggests a sleepiness detection system based on Google Glass. The glass had a proximity sensor connected to track how often people blink their eyes. Driving. This was put to the test by 23 experienced drivers, including 13 women and 10 men. It took place from 8 am to 8 pm. The viability of utilizing Google Glass to identify operator tiredness and establishing that the threshold algorithm for proximity sensors can accurately detect eye blinks, which may place sleepy drivers at larger risks than distracted drivers.
In the method proposed by (Valsan A et al., 2021) *Valsan A, V. et.al [19]*, a Real-time video of the driver is captured by the proposed system utilizing a digital camera. The driver's face is discernible in every frame of the footage thanks to

various image processing techniques. Using a single shape predictor, facial landmark points on the driver's face are located, and then the ratios of the eyes to the mouth and the frequency of yawning are calculated. Based on the values of these characteristics, drowsiness is identified. The thresholds are established using an adaptive thresholding approach. Algorithms for machine learning were also applied offline.

(Kumar & Patra, 2018)*A. Kumar et.al.[12]* proposed a method in which a camera records the footage, and using image processing methods, the driver's face is recognized in each frame. Facial landmarks on the detected face are identified, and tiredness is then determined using created adaptive thresholding based on the eye aspect ratio, mouth opening ratio, and nose length ratio values. There are other offline implementations of machine learning algorithms.

(Miranda et al., 2018)*Menchie Manda et. al. [14]* suggests The technology continually monitors the driver's eyelid movements, and once it detects sleepiness, it instantly notifies him with a random-sounding siren. Through internet connectivity, the web application instantly delivers the report to the car owner.

2.2 Limitation of State of the Art techniques

- The instruments that measure blood pressure, heart rate, breathing rate, and other bodily functions are used in the current sleepiness detection systems. The usage of these gadgets may make the motorist uncomfortable while driving. cannot guarantee that while driving, the drivers always wear these gadgets. may malfunction or become lost, which might cause a poor degree of accuracy in the outcome.

- Low light circumstances do not lend themselves to the current technology producing good outcomes. Lower accuracy is the result of the inability to recognize the driver's face and eyes in low or dim lighting situations.

- Also the existing systems only generate alerts after detecting drowsiness but there is no prevention approach to avoid accidents.

- It is challenging to assess the condition of the eye if the person wears spectacles. Because it depends on light, reflections from spectacles might produce the effect of an opened eye when the eye is closed.

- If more than one face appears in the window, the camera may pick up more faces, which might result in undesirable output. because of the various conditions of the many faces.

- Highly expensive

- Not portable

- Not Reliable

2.3 Discussion and future direction

In the proposed method, a camera is positioned in front of the driver to record the video, and the images were obtained by extracting the video frames from that. Additionally, the Haar cascade classifier is used to find the Face in the frames. The facial landmarks from the retrieved photos were identified, and from that, the aspect ratios of the mouth and eyes were calculated. The outcome of these computations and certain machine learning algorithms determines whether or not the driver is sleepy. If the result says the driver is sleepy then some preventive measures will be taken to avoid accidents. This measure includes the alert to the driver as well as the passenger and the speed of the vehicle will get reduced to zero.

2.4 Concluding Remarks

There are three types of driver drowsiness detection methods now in use: physiological, vehicle-based, and behavioral. The various assessments use a variety of factors to categorize driver tiredness.

2.4.1 Physiological Measures :

By focusing on the heart rate, pulse rate, brain activity, and body temperature, physiological measurements may be used to assess the health of the driver. The most common physiological measurements include a variety of signals, including electrocardiography, electromyography, electrooculography, and electroencephalography (ECG). The EEG technique uses the electrical activity of the human brain to measure brain waves that are utilized in many different applications, such as the diagnosis of epilepsy and the monitoring of sleep problems. EOG is a method for measuring the cornea-retinal standing potential that exists between the front and the rear of the human eye. It records the movement of the eye and measures the cornea-retinal level. Through surface electrodes that are kept in touch with the driver's skin, EMG analyses and documents the electrical activity of the muscles. Driving-related cardiac activity, such as the driver's heart rate, rhythm, and electrical activity may be monitored and evaluated using the ECG method.

2.4.2 Vehicle Based Measures :

A tired motorist may behave differently behind the wheel than a typical driver would. Driver tiredness may be predicted by keeping an eye on a few metrics such as lane departure, steering movement, rapid changes in acceleration, gas pedal, and brake pedal. A typical driver will have a predictable driving style, but if he veers slightly out of his lane or does any other vehicle-based action, it might trigger an unwarranted alarm. Due to the potential for greater false positives, only a small number of investigations have employed the vehicle-based measure.

2.4.3 Behavioural Based :

Behavioral assessments based on frequent yawning, eye closure, eye blinking, and facial expressions have been widely employed by researchers. Recent years have seen a significant increase in the use of machine learning algorithms to detect driver sleepiness through the use of collected photos or videos. Researchers have used various lightweight algorithms to improve accuracy while decreasing execution time and cost.

Chapter 3

Problem Definition and Scope

3.1 Problem statement

IoT-based drowsiness detection system with speed control mechanism.

3.2 Goals and Objectives

- To design a system to detect the driver's drowsiness by continuously monitoring the driver
- To alert the driver on the detection of drowsiness by using a buzzer or alarm
- Reduce the Speed of the Vehicle on Drowsiness Detection

3.3 Scope and Major Constraints

The current system is focused on the speed control mechanism after drowsiness detection. Further, the system accuracy can be raised by doing the adaptation under poor light and extending the face-capturing range.

Description	Constraint
Light dependency	Proper ambient Light
OpenCV scan range(Webcam to face)	23.5 cm
Face Orientation	Maximum 60 degrees
Multiple Faces	Not more than one

Table 3.1: Major Constraints

3.4 Software and Hardware Requirements

3.4.1 Hardware requirements:

- Raspberry pi
- Buzzer
- Breadboard
- DC motor
- SD card
- Pi camera
- Jumper cable

3.4.2 Software requirements:

- Python:
  - Python 3 Libraries
  - Numpy
  - Pygame
  - Dlib
  - Imutils
  - OpenCV, etc
- Operating System: Windows or Ubuntu

3.5 Expected Outcomes

- System should be able to detect the driver's drowsiness by continuously monitoring the driver
- The system should be able to give an alert on the detection of drowsiness by using a buzzer or alarm
- The system should be able to Reduce the Speed of the Vehicle on Drowsiness Detection

---

Chapter 4

System Requirement Specification

4.1 Overall Description

In Chapter No 4, we described the system's required proposed block diagram, user case diagram, sequence diagram, and activity diagram. In this section, project duration, start date, and end date are mentioned.

4.1.1 Block diagram/ Proposed System setup

The proposed system setup is shown in Figure 4.1 Real-time video is captured through the camera and frames are extracted from the video. The face, eye, and mouth are tracked and drowsiness is detected. Alert will be given along with vehicle speed reduction.

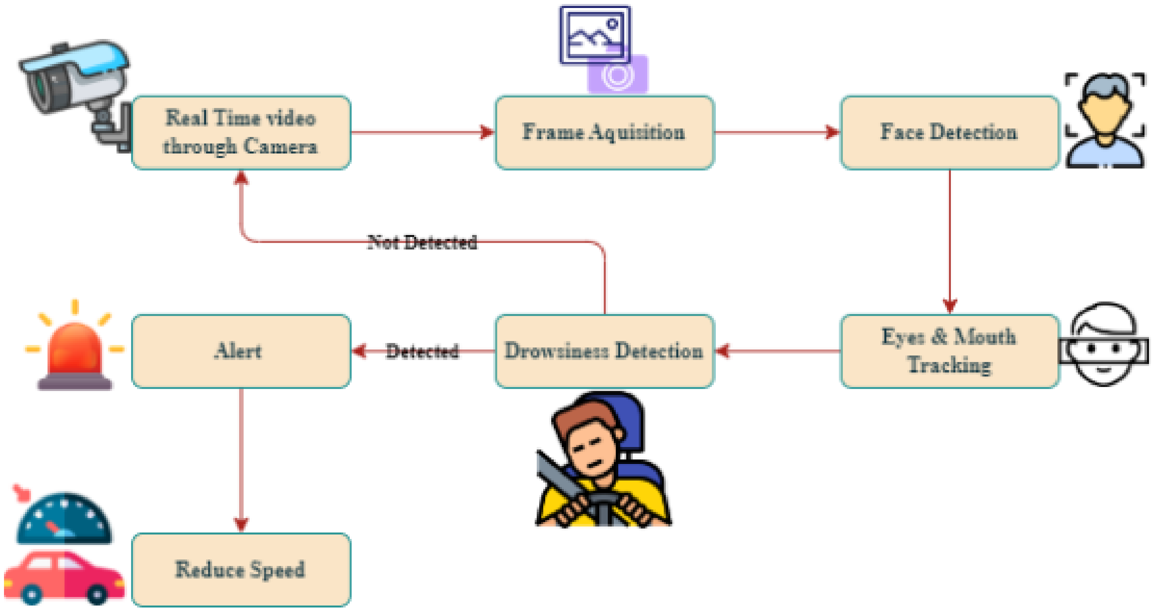


Figure 4.1: Block Diagram

4.1.2 Use Case Diagram

Use case diagram is shown in the Figure 4.2. It indicates the scope of the system and the high-level function of the system.

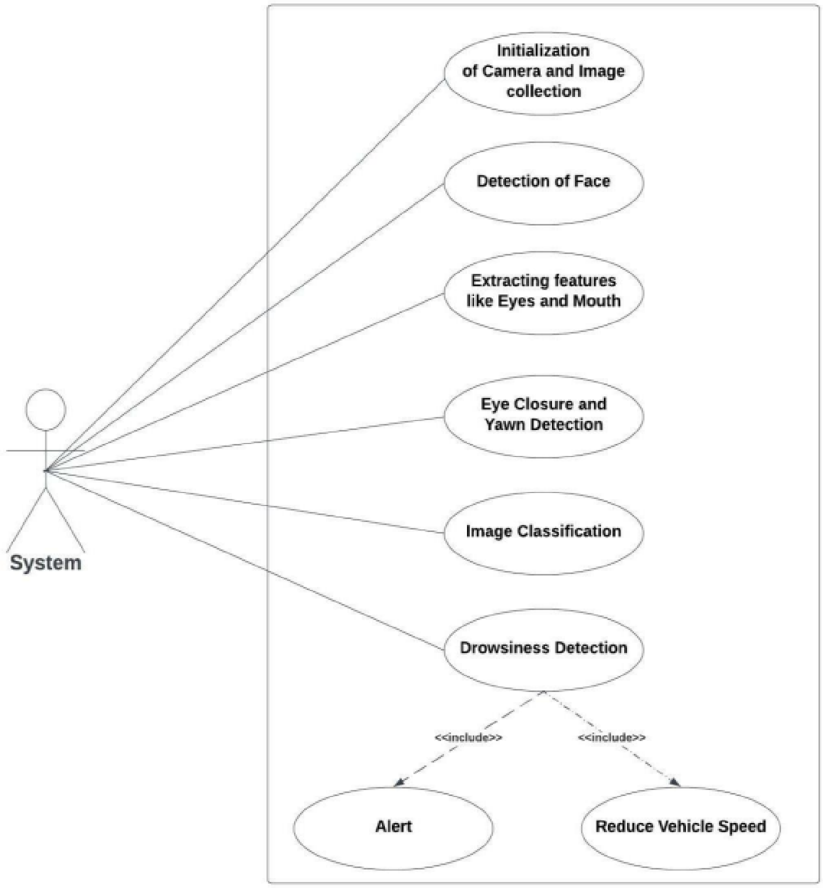


Figure 4.2: User Case Diagram

4.1.3 Sequence Diagram

The sequence Diagram is shown in Figure 4.3. It shows the sequence of operations performed by the system. Driver, System, and Vehicle are the three major elements of the proposed system.

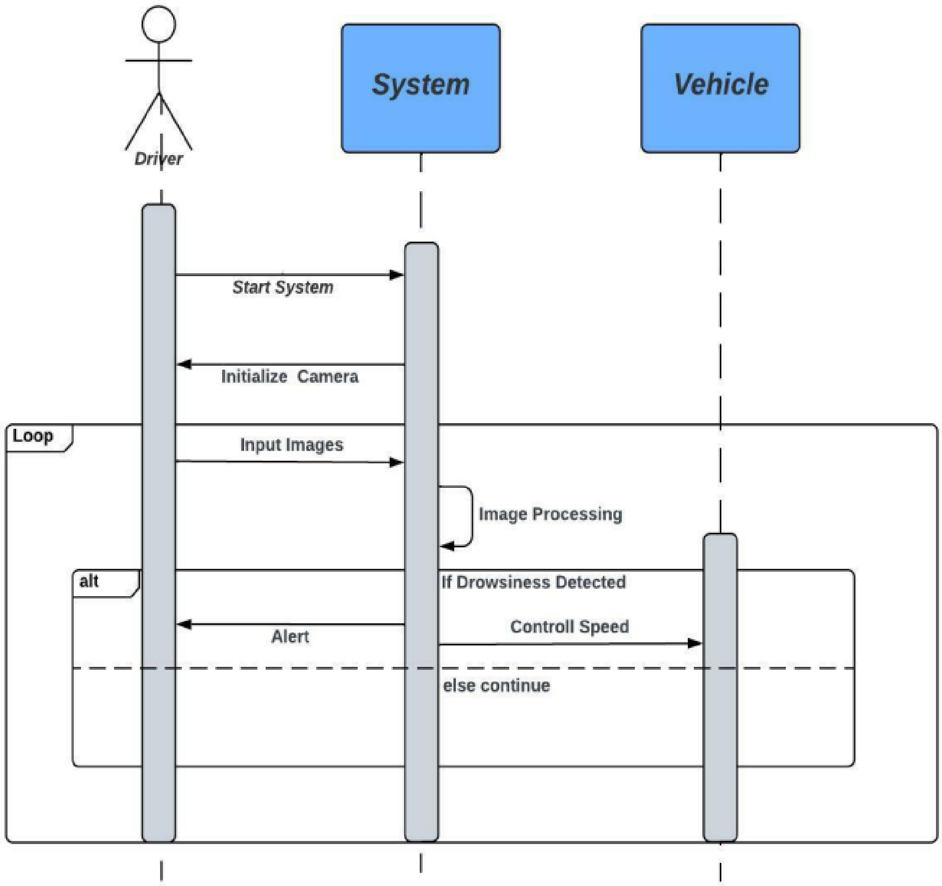


Figure 4.3: Sequence Diagram

4.1.4 Activity Diagram

Activity Diagram is shown in the Figure 4.4. It indicates the detailed working of the system.

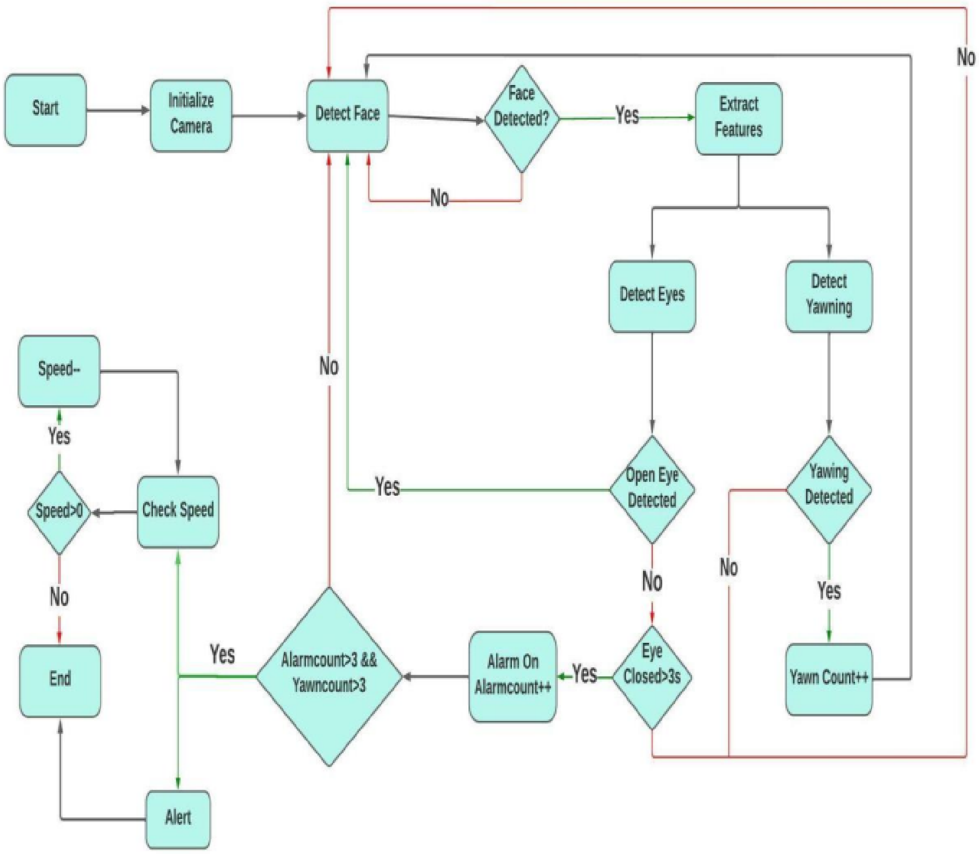


Figure 4.4: Activity Diagram

4.2 Project Planning

In figure4.5 the project planning section describes the overall project implementation planning. The first step is Problem Identification, where we identify a specified problem. Task 2 is the literature survey where we study recent research papers related to our topic. Task 3 is system requirements, gathering all that is required to complete the system. Task 4 is user analysis where we find our user's requirements and needs in our proposed system. Task 5 System design. After completing all these tasks, we will design the front end of a website.

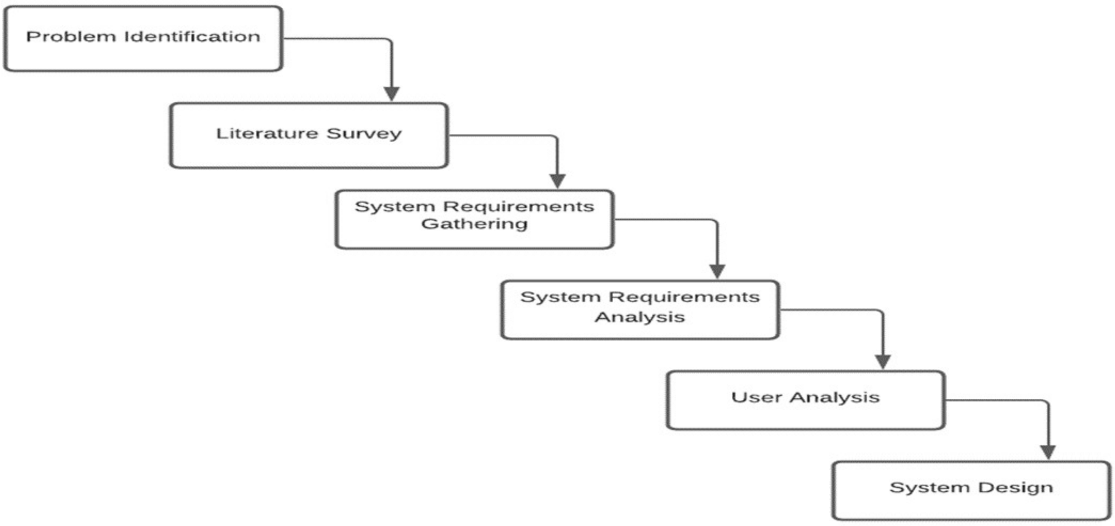
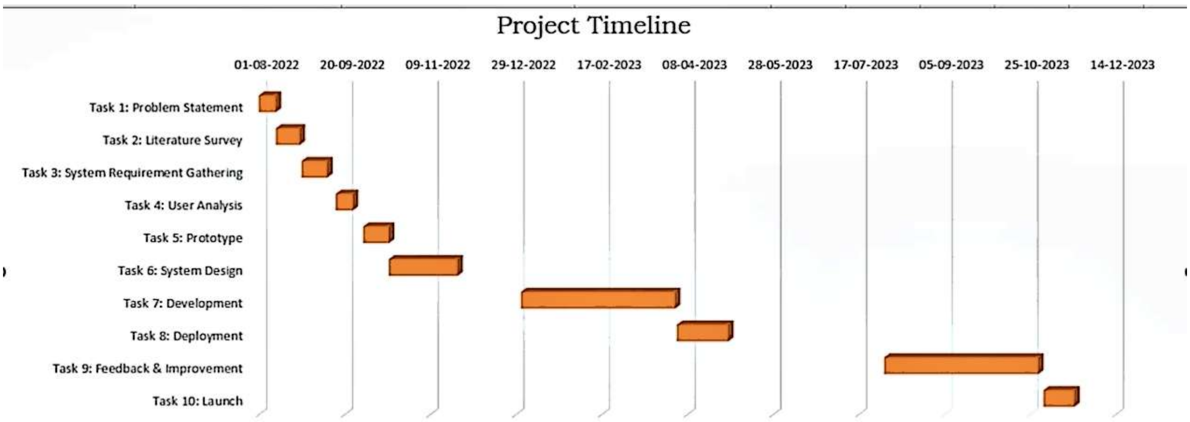


Figure 4.5: Project Planning

4.2.1 Project Timeline

In figure4.6 project Timeline shows an overall timeline of a project task-wise in the Gantt chart. It shows a timeline of a particular task throughout the project and it will help to track the project. In this section project duration, start date, and end date are mentioned. This timeline is for three-semester project execution from problem Identification to the launch of the project. There are 10 tasks in total out of which the first four tasks should be completed in the V semester and tasks 7 and task 8 should complete in semester VI and the last two tasks that is Feedback and improvement and task 10 Launch should be done in the VIIth semester. Using this chart we can manage our project execution plan and track the process throughout it.



	Task 10: Launch	Task 9: Feedback & Improvement	Task 8: Deployment	Task 7: Development	Task 6: System Design	Task 5: Prototype	Task 4: User Analysis	Task 3: System Requirement Gathering	Task 2: Literature Survey	Task 1: Problem Statement
Start Date	02-11-2023	01-08-2023	02-04-2023	01-01-2023	16-10-2022	01-10-2022	15-09-2022	26-08-2022	11-08-2022	01-08-2022
Duration	18	90	30	90	40	15	10	15	14	10

Figure 4.6: Project Timeline

Chapter 5

Proposed Methodology

5.1 System Architecture

User will interact with the website by doing different task to do this task user need to create new login credentials. After creating login credentials user gets a dashboard where the user can do different activities as per requirements. The first option user gets on a dashboard is to apply for a pass. Then ticket booking, track bus, view pass, and profile view as shown in Figure 5.1

5.2 Mathematical Modeling

Every algorithm relies on a set of instructions that form its fundamental structure. In the Viola-Jones algorithm, these instructions are represented by Haar features, which consist of rectangular kernels, as depicted in Figure 5.2. During the algorithm's development phase, researchers introduced rotated Haar features, shown in Figure 5.3 and Figure 5.4, as well as asymmetric configurations. Additionally, instead of calculating a simple difference, they proposed assigning specific weights to each subregion and computing the corresponding values as a weighted sum of pixels from different types of regions.

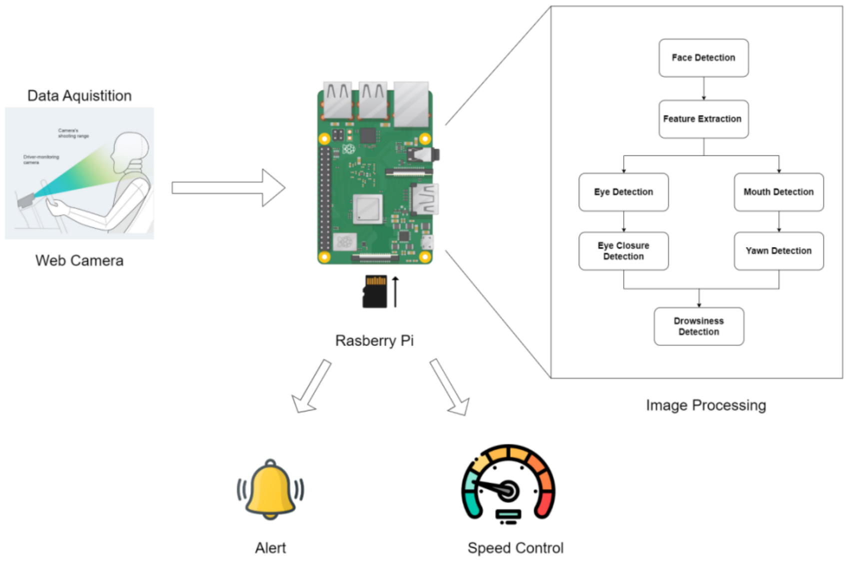


Figure 5.1: Proposed Block Diagram

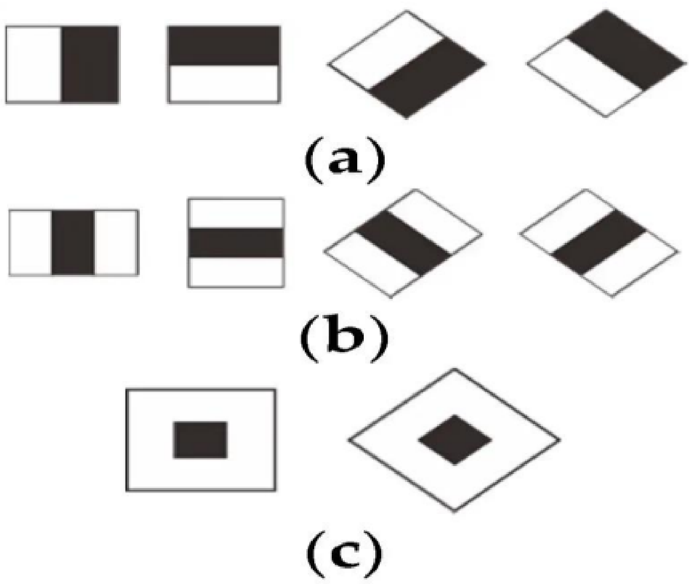


Figure 5.2: Sample of Haar Features

$$feature_I = \sum_{i \in I = \{1, \ldots, N\}} \omega_i \cdot RecSum(r_i)$$

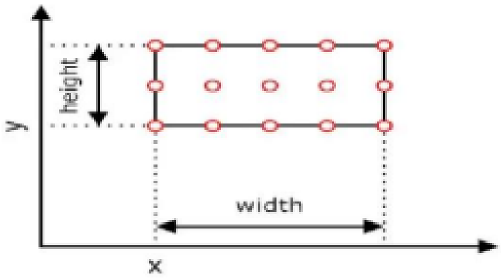


Figure 5.3: Initial Feature state

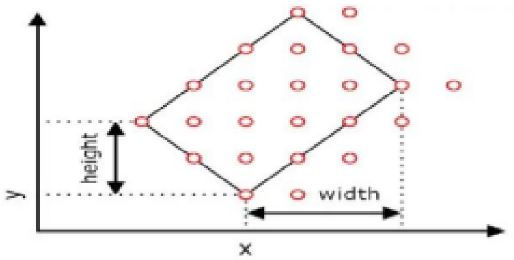


Figure 5.4: After 45° Rotation

In order to determine the class for each cascade, the sum of the quantities assigned by the weak classifiers within that cascade is calculated. Each weak classifier contributes two quantities based on whether the attribute value associated with that classifier is greater or lesser than a specified threshold.

$$\sum_{i=1}^{k}\left(w_I \cdot \sum_{u=i+R_I y}^{R_I y+R_I height-1} \sum_{v=j+R_I yx}^{R_I y+R_I width-1} A_{uv} \right) < norm(i,\ j) \cdot \ threshold(t)$$

Eye Aspect Ratio (EAR) :

Figure 5.5: Eye Landmarks

We can detect the eyes using facial landmarks as shown in Figure 5.5. The Right eye uses [37,42] and the Left eye uses [43,45]. The following formula is used to calculate the eye-aspect ratio. There are a total of 6 spots on the eyes: 2 on the top eyelid, 2 on the lower eyelid, and 2 in the center of the eye. In order to compute the vertical distance for the eye-aspect ratio, multiply the horizontal distance from the center of the eye by two.

$$EAR = ||P5 - P1|| - ||P4 - P2||/2(||P0 - P3||)$$

When an eye is open, the EAR is generally constant, and when it is closed, it approaches zero. Based on the EAR computed in the preceding stage, the choice of the eye state is taken. The eye condition is categorized as ”closed” if the distance is zero or nearly zero; otherwise, it is categorized as ”open.”

Mouth Aspect Ratio (MAR):

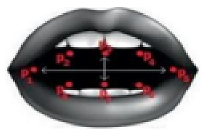


Figure 5.6: Mouth Landmarks

We can detect mouth by using landmarks [61,68] as shown in Figure 5.6. A formula that is presented below is used to determine the mouth aspect ratio. To do this, we need eight locations on the lips as references: two on the upper, two on the lower, and two in the center. The vertical distance in the mouth aspect ratio is determined by dividing by two the horizontal distance from the center of the mouth.

$$MAR = ||P2 - P8|| - ||P3 - P7|| - ||p4 - p6||/3(||P1 - P5||)$$

5.3 Approach

Step 1: Data Collection: Take an image as input using the camera.

Step 2: Data Preprocessing: From these images, the facial region will get detected and then the system will detect the eyes from the facial region.

Step 3: Decision Making: Images will be fed to the classifier and the classifier will classify the state of driver's eye and predict if the eyes are open or closed. Based on the result, the system will attempt tasks such as alerting and controlling the speed of the vehicle.

Step 4: Performance Measure: In this phase, we will measure and analyze the performance and accuracy of the model based on the results.

Step 5: Deployment: If the model is giving accurate results as per expectation, it will go for deployment stag

---

Chapter 6

Implementation

6.1 System Implementation

System implementation is the process of converting a designed system into a fully operational and functional system. This ensures that the information system meets quality standards and is in a usable state.

6.2 Experiment/Implementation Parameters

6.2.1 Libraries Used

Figure 6.1 shows the libraries used for data extraction and image processing. `scipy.spatial.distance` is imported as `dist`. It provides functions for calculating distances between points or arrays.

`imutils.video` is imported as `VideoStream`. It allows accessing video streams from various sources.

`imutils.face_utils` is imported. It contains various utility functions for working

```python
from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
import os
import serial
import subprocess
script_path = 'emailmodule.py'
```

Figure 6.1: Libraries Used for Image Processing

with facial landmarks.

`threading.Thread` is imported for creating a separate thread for video processing.

`numpy` is imported as `np`. It provides support for numerical operations on multi-dimensional arrays and matrices.

`argparse` is imported for parsing command-line arguments.

`imutils` provides various utility functions for image and video processing.

`time` is imported for time-related functions.

`dlib` is a library for machine learning, including facial landmark detection.

`cv2` is the OpenCV library for computer vision and image processing.

`os` is imported for operating system-related functions.

`serial` is imported for serial communication with external devices.

`subprocess` is imported for creating additional processes.

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

    while alarm_status:
        print('call')
        s = 'espeak "'+msg+'"'
        os.system(s)

    if alarm_status2:
        print('call')
        saying = True
        s = 'espeak "' + msg + '"'
        os.system(s)
        saying = False
```

Figure 6.2: Alarm and Message Function

6.2.2 def alarm

This function is designed to continuously speak the provided message using the *espeak* text-to-speech engine while the *alarm_status* variable is true. Additionally, if *alarm_status2* is true, it will speak the message once and set a flag to indicate that the speaking process is ongoing. Which is shown in Figure 6.2

```python
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
```

Figure 6.3: Function for Eye Aspect Ratio

6.2.3 def_eye_aspect_ratio

Calculate the Euclidean distance between two points, `eye[1]` and `eye[5]`, representing the vertical extent of the eye (A).

Calculate the Euclidean distance between two points, `eye[2]` and `eye[4]`, representing the horizontal extent of the eye (B).

Calculate the Euclidean distance between two points, `eye[0]` and `eye[3]`, representing the distance between the top and bottom landmarks of the eye (C).

Calculate the EAR by dividing the sum of A and B by twice the value of C.

The division by 2.0 normalizes the values by taking the average distance between the vertical and horizontal landmarks, relative to the height of the eye.

Return the calculated EAR.

By using this function, you can obtain the eye aspect ratio for a given eye and use it for further analysis or applications, such as determining if the eye is open or closed, detecting fatigue or drowsiness, or tracking eye movements. Which is

```
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)
```

Figure 6.4: Final Calculation of EAR

shown in Figure 6.3

6.2.4 final_ear

The function `final_ear(shape)` calculates the final Eye Aspect Ratio (EAR) for both the left and right eyes using the provided facial landmarks. The function takes a `shape` object, typically representing the facial landmarks detected on a face, as input. Which is shown in Figure 6.4

```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
```

Figure 6.5: Calculation of Distance between top and bottom Lips

6.2.5 lip_distance

The function `lip_distance(shape)` calculates the vertical distance between the top and bottom lips using the provided facial landmarks. The function takes a `shape` object, which likely represents the facial landmarks detected on a face, as input. Which is shown in Figure 6.5

6.3 User Interface

The user interface is used to communicate user with the computer system. The primary goal of the user interface is to provide a tentative and efficient user experience.
All the messages about each safety test like seat belt detection, alcohol detection, and drowsiness detection, would be passed through the LED Display. Upon each unsuccessful check, the driver has to act accordingly to go for the next check.
Once the user enters the drowsiness detection stage, he will be monitored continuously and alerted for every drowsiness count. If the driver enters a completely drowsiness state, the safety alarm system and speed control mechanism will start working.

Figure 6.6: LED Display to the driver

6.4 Functional Implementation

As shown in Figure 6.7 for detection of whether the user is alcoholic or not a Gas sensor is used. MQ-3 Gas sensor is used to detect whether the driver is drunk or not. MQ-3 sensor is highly sensitive to alcohol with a simple circuit and faster response time. It works on both AC and DC voltage.

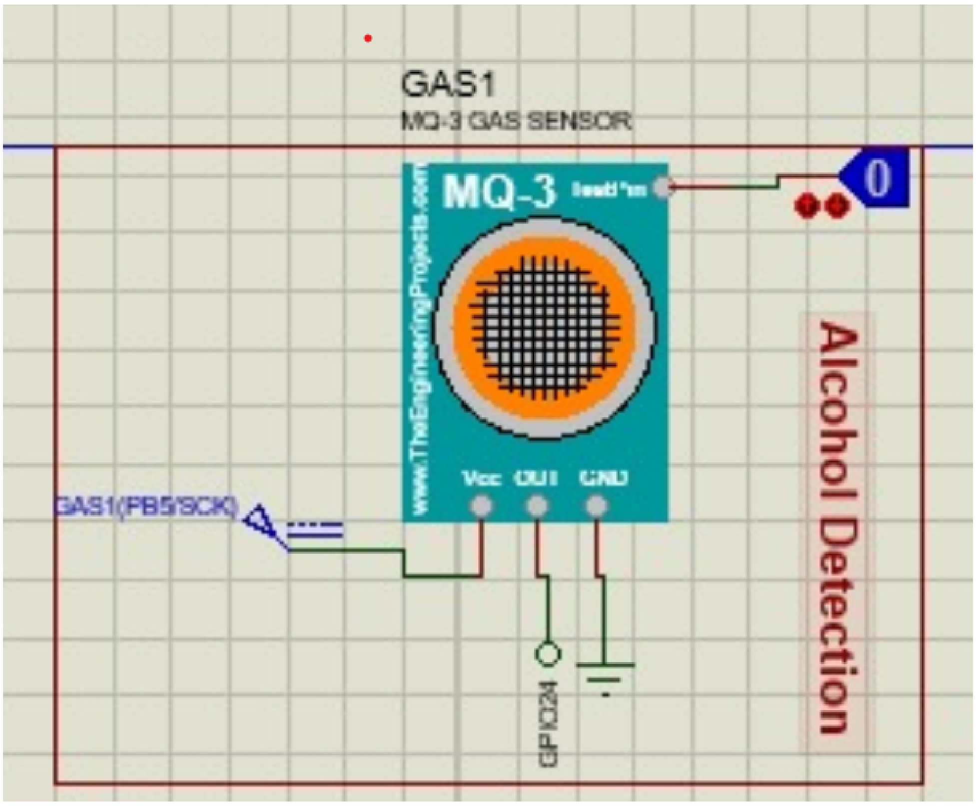


Figure 6.7: Alcohol Detection Implementation

If the sensor detects the driver to be drunk then the vehicle does not start and the message for the same is displayed to the driver on the LED screen as shown in Figure 6.8. Otherwise, the system checks for seat belts.

Figure 6.9 shows the implementation of Seat Belt Detection. After the system checks the alcohol and passes the check, then it goes for checking whether the driver is wearing a seat belt or not.

If the seat belt is detected then the driver can start the vehicle. If not driver is instructed to wear a seat belt through the Display as shown in Figure 6.10.

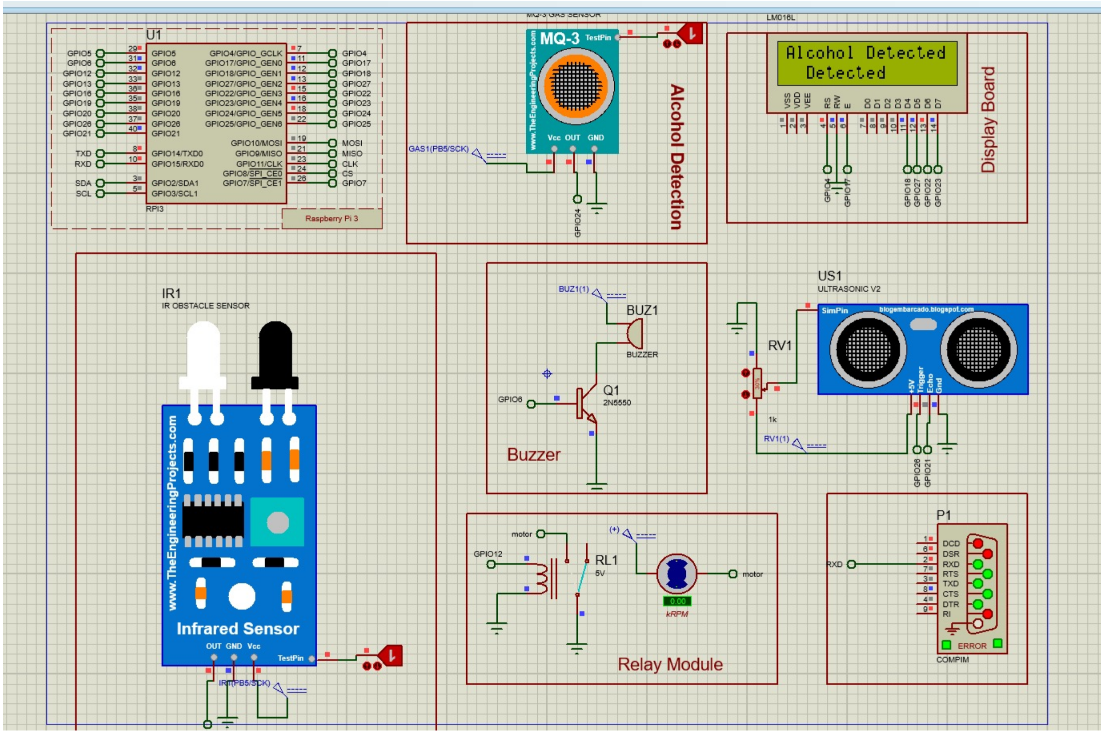


Figure 6.8: Driver Detected to be Alcoholic

After the system has passed the two checks for seat belt and alcohol detection, then the system will be continuously monitoring the driver through the camera, as shown in Figure 6.11 and count the drowsiness.

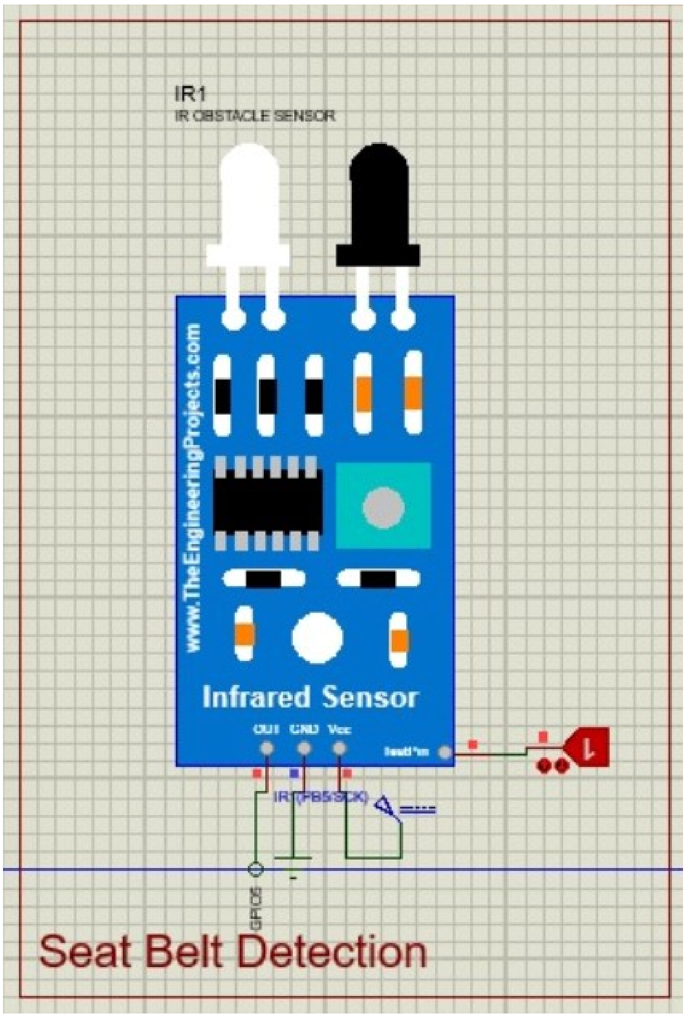


Figure 6.9: Seat Belt Detection

After the drowsiness is detected, the buzzer will start ringing and the speed of the vehicle will be controlled. At the same time, LEDs will start glowing to indicate to other vehicle drivers on the road about the drowsy driver.

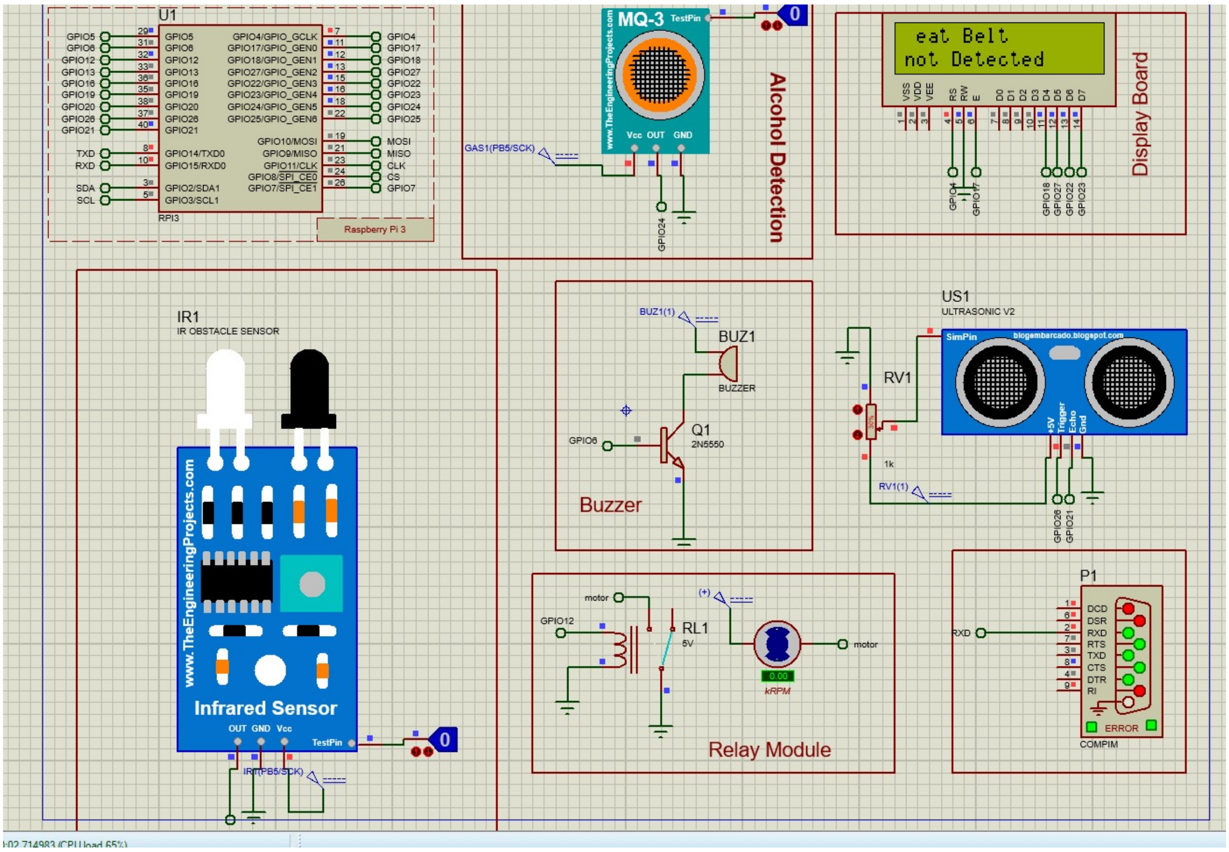


Figure 6.10: no Seat Belt Detected

6.5 Project Code

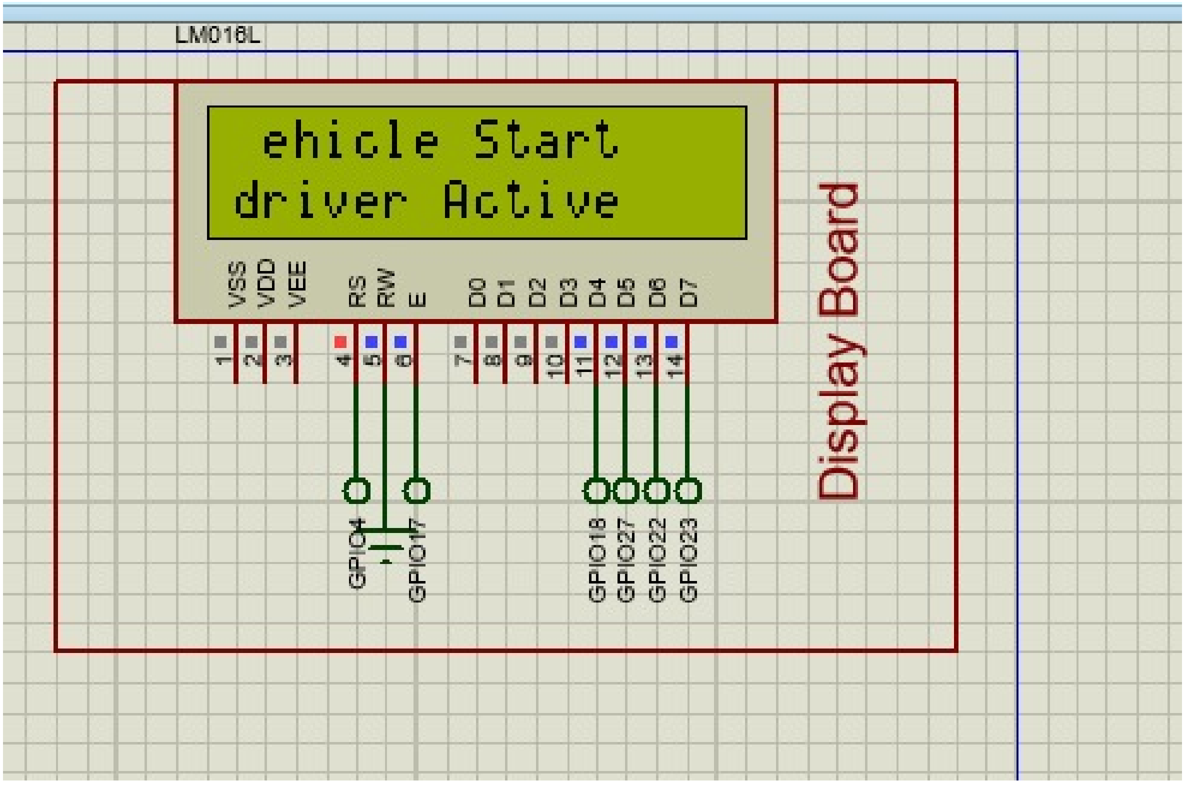


Figure 6.11: Active Driver Status

6.6 Output

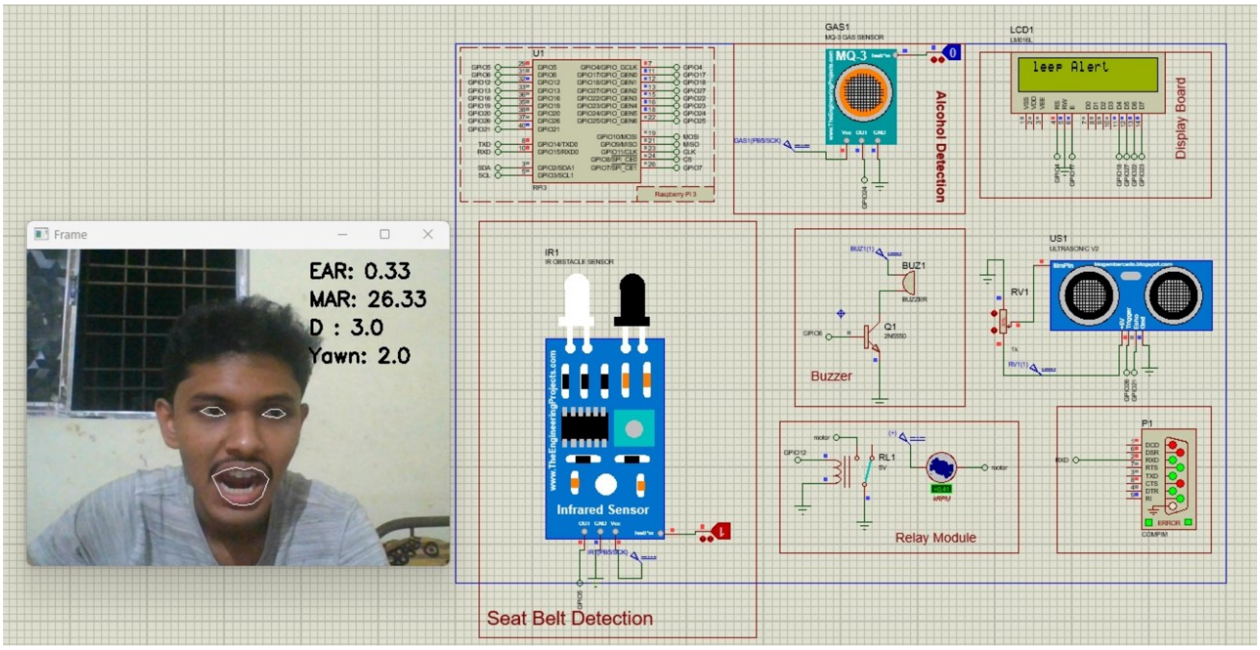
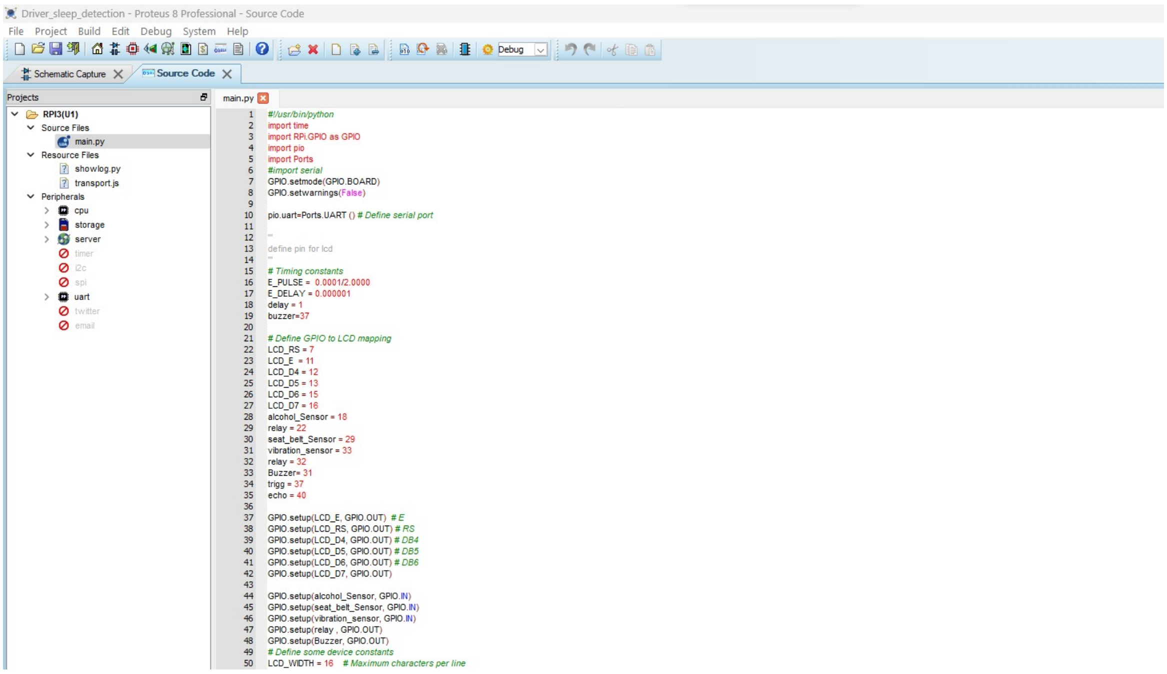
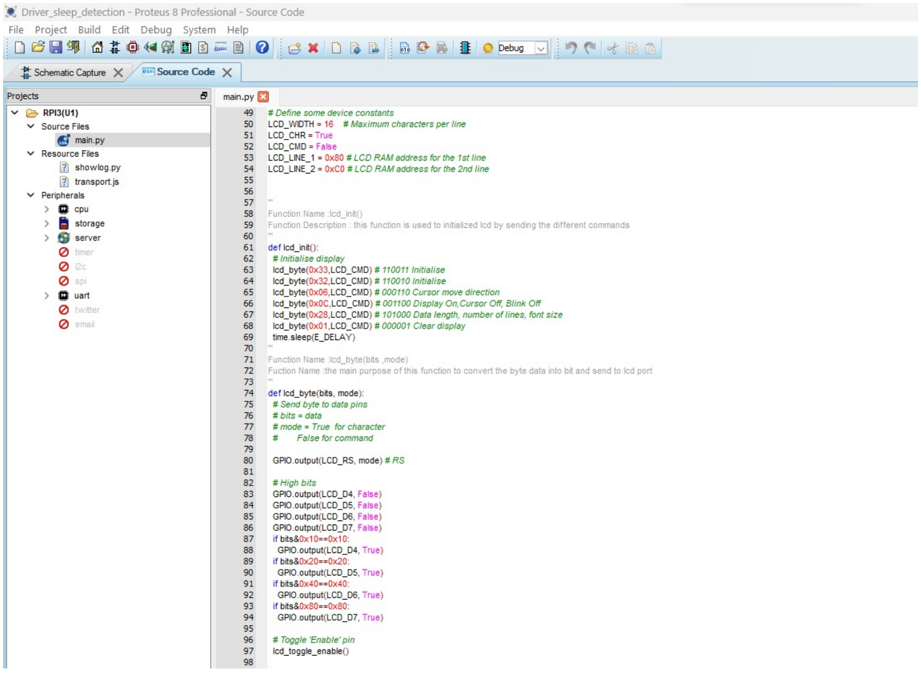
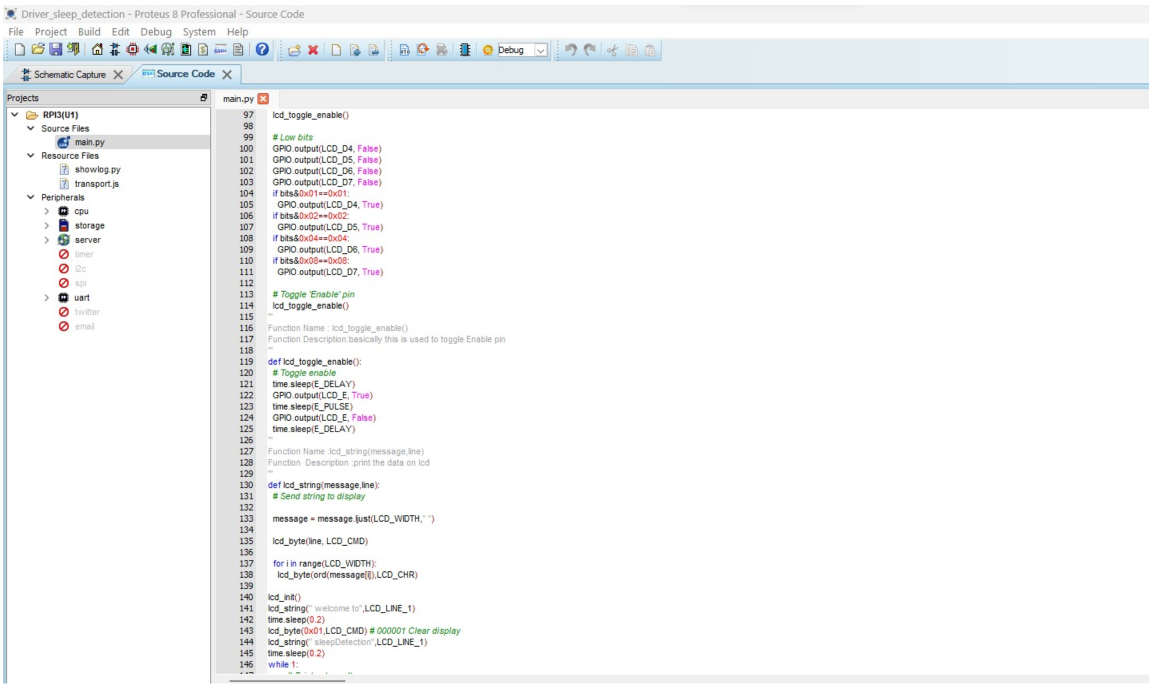


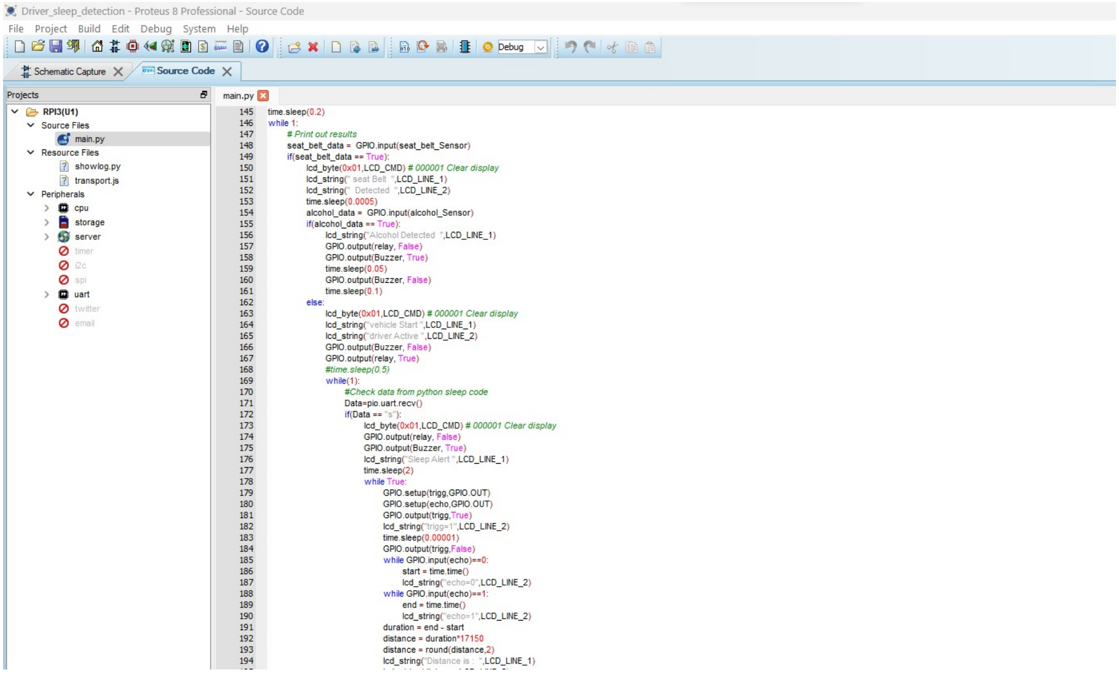
Figure 6.12: Drowsiness Detection (Sleep Alert)



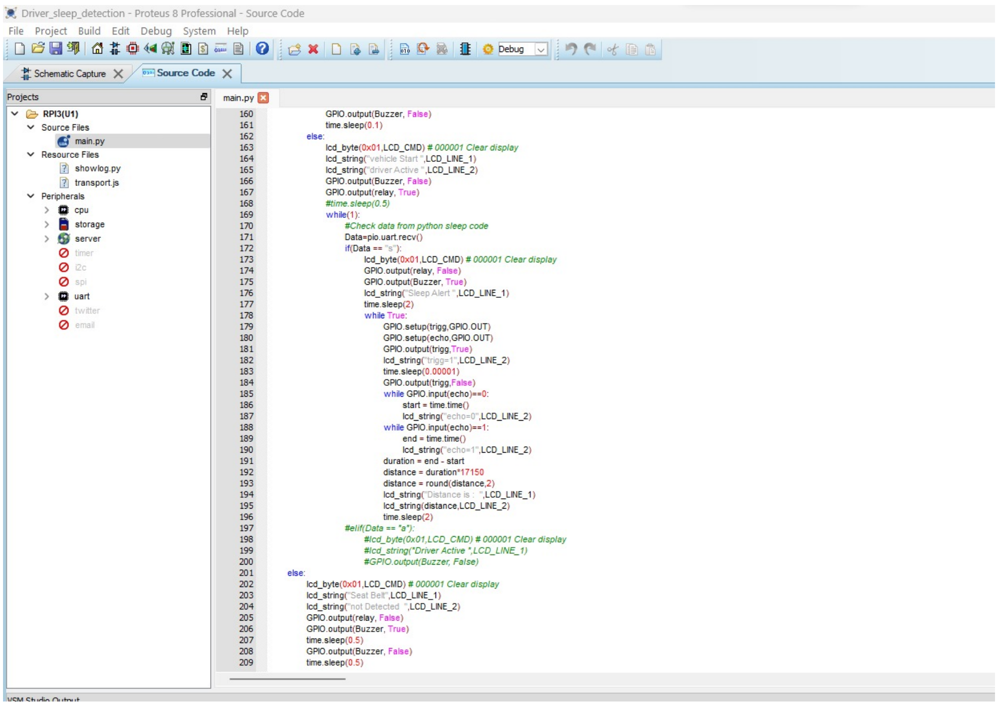

```python
#!/usr/bin/python
import time
import RPi.GPIO as GPIO
import pio
import Ports
#import serial
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

pio.uart=Ports.UART () # Define serial port

'''
define pin for lcd
'''
# Timing constants
E_PULSE =  0.0001/2.0000
E_DELAY = 0.000001
delay = 1
buzzer=37

# Define GPIO to LCD mapping
LCD_RS = 7
LCD_E  = 11
LCD_D4 = 12
LCD_D5 = 13
LCD_D6 = 15
LCD_D7 = 16
alcohol_Sensor = 18
relay = 22
seat_belt_Sensor = 29
vibration_sensor = 33
relay = 32
Buzzer= 31
trigg = 37
echo = 40

GPIO.setup(LCD_E, GPIO.OUT)  # E
GPIO.setup(LCD_RS, GPIO.OUT) # RS
GPIO.setup(LCD_D4, GPIO.OUT) # DB4
GPIO.setup(LCD_D5, GPIO.OUT) # DB5
GPIO.setup(LCD_D6, GPIO.OUT) # DB6
GPIO.setup(LCD_D7, GPIO.OUT)

GPIO.setup(alcohol_Sensor, GPIO.IN)
GPIO.setup(seat_belt_Sensor, GPIO.IN)
GPIO.setup(vibration_sensor, GPIO.IN)
GPIO.setup(relay , GPIO.OUT)
GPIO.setup(Buzzer, GPIO.OUT)
# Define some device constants
LCD_WIDTH = 16    # Maximum characters per line
```


Driver_sleep_detection - Proteus 8 Professional - Source Code
File Project Build Edit Debug System Help
Debug
Schematic Capture
Source Code
Projects
RPI3(U1)
Source Files
main.py
Resource Files
showlog.py
transport.js
Peripherals
cpu
storage
server
timer
i2c
spi
uart
twitter
email
main.py
# Define some device constants
LCD_WIDTH = 16 # Maximum characters per line
LCD_CHR = True
LCD_CMD = False
LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line
'''
Function Name :lcd_init()
Function Description : this function is used to initialized lcd by sending the different commands
'''
def lcd_init():
# Initialise display
lcd_byte(0x33,LCD_CMD) # 110011 Initialise
lcd_byte(0x32,LCD_CMD) # 110010 Initialise
lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
lcd_byte(0x0C,LCD_CMD) # 001100 Display On,Cursor Off, Blink Off
lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines, font size
lcd_byte(0x01,LCD_CMD) # 000001 Clear display
time.sleep(E_DELAY)
'''
Function Name :lcd_byte(bits ,mode)
Fuction Name :the main purpose of this function to convert the byte data into bit and send to lcd port
'''
def lcd_byte(bits, mode):
# Send byte to data pins
# bits = data
# mode = True for character
# False for command
GPIO.output(LCD_RS, mode) # RS
# High bits
GPIO.output(LCD_D4, False)
GPIO.output(LCD_D5, False)
GPIO.output(LCD_D6, False)
GPIO.output(LCD_D7, False)
if bits&0x10==0x10:
GPIO.output(LCD_D4, True)
if bits&0x20==0x20:
GPIO.output(LCD_D5, True)
if bits&0x40==0x40:
GPIO.output(LCD_D6, True)
if bits&0x80==0x80:
GPIO.output(LCD_D7, True)
# Toggle 'Enable' pin
lcd_toggle_enable()


Driver_sleep_detection - Proteus 8 Professional - Source Code
File Project Build Edit Debug System Help
Debug
Schematic Capture
Source Code
Projects
RPI3(U1)
Source Files
main.py
Resource Files
showlog.py
transport.js
Peripherals
cpu
storage
server
timer
i2c
spi
uart
twitter
email
main.py
lcd_toggle_enable()
# Low bits
GPIO.output(LCD_D4, False)
GPIO.output(LCD_D5, False)
GPIO.output(LCD_D6, False)
GPIO.output(LCD_D7, False)
if bits&0x01==0x01:
GPIO.output(LCD_D4, True)
if bits&0x02==0x02:
GPIO.output(LCD_D5, True)
if bits&0x04==0x04:
GPIO.output(LCD_D6, True)
if bits&0x08==0x08:
GPIO.output(LCD_D7, True)
# Toggle 'Enable' pin
lcd_toggle_enable()
'''
Function Name : lcd_toggle_enable()
Function Description:basically this is used to toggle Enable pin
'''
def lcd_toggle_enable():
# Toggle enable
time.sleep(E_DELAY)
GPIO.output(LCD_E, True)
time.sleep(E_PULSE)
GPIO.output(LCD_E, False)
time.sleep(E_DELAY)
'''
Function Name :lcd_string(message,line)
Function Description :print the data on lcd
'''
def lcd_string(message,line):
# Send string to display
message = message.ljust(LCD_WIDTH," ")
lcd_byte(line, LCD_CMD)
for i in range(LCD_WIDTH):
lcd_byte(ord(message[i]),LCD_CHR)
lcd_init()
lcd_string(" welcome to",LCD_LINE_1)
time.sleep(0.2)
lcd_byte(0x01,LCD_CMD) # 000001 Clear display
lcd_string(" sleepDetection",LCD_LINE_1)
time.sleep(0.2)
while 1:




```python
time.sleep(0.2)
while 1:
    # Print out results
    seat_belt_data =  GPIO.input(seat_belt_Sensor)
    if(seat_belt_data == True):
        lcd_byte(0x01,LCD_CMD) # 000001 Clear display
        lcd_string(" seat Belt  ",LCD_LINE_1)
        lcd_string("  Detected  ",LCD_LINE_2)
        time.sleep(0.0005)
        alcohol_data =  GPIO.input(alcohol_Sensor)
        if(alcohol_data == True):
            lcd_string("Alcohol Detected  ",LCD_LINE_1)
            GPIO.output(relay, False)
            GPIO.output(Buzzer, True)
            time.sleep(0.05)
            GPIO.output(Buzzer, False)
            time.sleep(0.1)
        else:
            lcd_byte(0x01,LCD_CMD) # 000001 Clear display
            lcd_string("vehicle Start ",LCD_LINE_1)
            lcd_string("driver Active ",LCD_LINE_2)
            GPIO.output(Buzzer, False)
            GPIO.output(relay, True)
            #time.sleep(0.5)
            while(1):
                #Check data from python sleep code
                Data=pio.uart.recv()
                if(Data == "s"):
                    lcd_byte(0x01,LCD_CMD) # 000001 Clear display
                    GPIO.output(relay, False)
                    GPIO.output(Buzzer, True)
                    lcd_string("Sleep Alert ",LCD_LINE_1)
                    time.sleep(2)
                    while True:
                        GPIO.setup(trigg,GPIO.OUT)
                        GPIO.setup(echo,GPIO.OUT)
                        GPIO.output(trigg,True)
                        lcd_string("trigg=1",LCD_LINE_2)
                        time.sleep(0.00001)
                        GPIO.output(trigg,False)
                        while GPIO.input(echo)==0:
                            start = time.time()
                            lcd_string("echo=0",LCD_LINE_2)
                        while GPIO.input(echo)==1:
                            end = time.time()
                            lcd_string("echo=1",LCD_LINE_2)
                        duration = end - start
                        distance = duration*17150
                        distance = round(distance,2)
                        lcd_string("Distance is :  ",LCD_LINE_1)
```




```python
            GPIO.output(Buzzer, False)
            time.sleep(0.1)
        else:
            lcd_byte(0x01,LCD_CMD) # 000001 Clear display
            lcd_string("vehicle Start ",LCD_LINE_1)
            lcd_string("driver Active ",LCD_LINE_2)
            GPIO.output(Buzzer, False)
            GPIO.output(relay, True)
            #time.sleep(0.5)
            while(1):
                #Check data from python sleep code
                Data=pio.uart.recv()
                if(Data == "s"):
                    lcd_byte(0x01,LCD_CMD) # 000001 Clear display
                    GPIO.output(relay, False)
                    GPIO.output(Buzzer, True)
                    lcd_string("Sleep Alert ",LCD_LINE_1)
                    time.sleep(2)
                    while True:
                        GPIO.setup(trigg,GPIO.OUT)
                        GPIO.setup(echo,GPIO.OUT)
                        GPIO.output(trigg,True)
                        lcd_string("trigg=1",LCD_LINE_2)
                        time.sleep(0.00001)
                        GPIO.output(trigg,False)
                        while GPIO.input(echo)==0:
                            start = time.time()
                            lcd_string("echo=0",LCD_LINE_2)
                        while GPIO.input(echo)==1:
                            end = time.time()
                            lcd_string("echo=1",LCD_LINE_2)
                        duration = end - start
                        distance = duration*17150
                        distance = round(distance,2)
                        lcd_string("Distance is :  ",LCD_LINE_1)
                        lcd_string(distance,LCD_LINE_2)
                        time.sleep(2)
                #elif(Data == "a"):
                    #lcd_byte(0x01,LCD_CMD) # 000001 Clear display
                    #lcd_string("Driver Active ",LCD_LINE_1)
                    #GPIO.output(Buzzer, False)
    else:
        lcd_byte(0x01,LCD_CMD) # 000001 Clear display
        lcd_string("Seat Belt",LCD_LINE_1)
        lcd_string("not Detected  ",LCD_LINE_2)
        GPIO.output(relay, False)
        GPIO.output(Buzzer, True)
        time.sleep(0.5)
        GPIO.output(Buzzer, False)
        time.sleep(0.5)
```

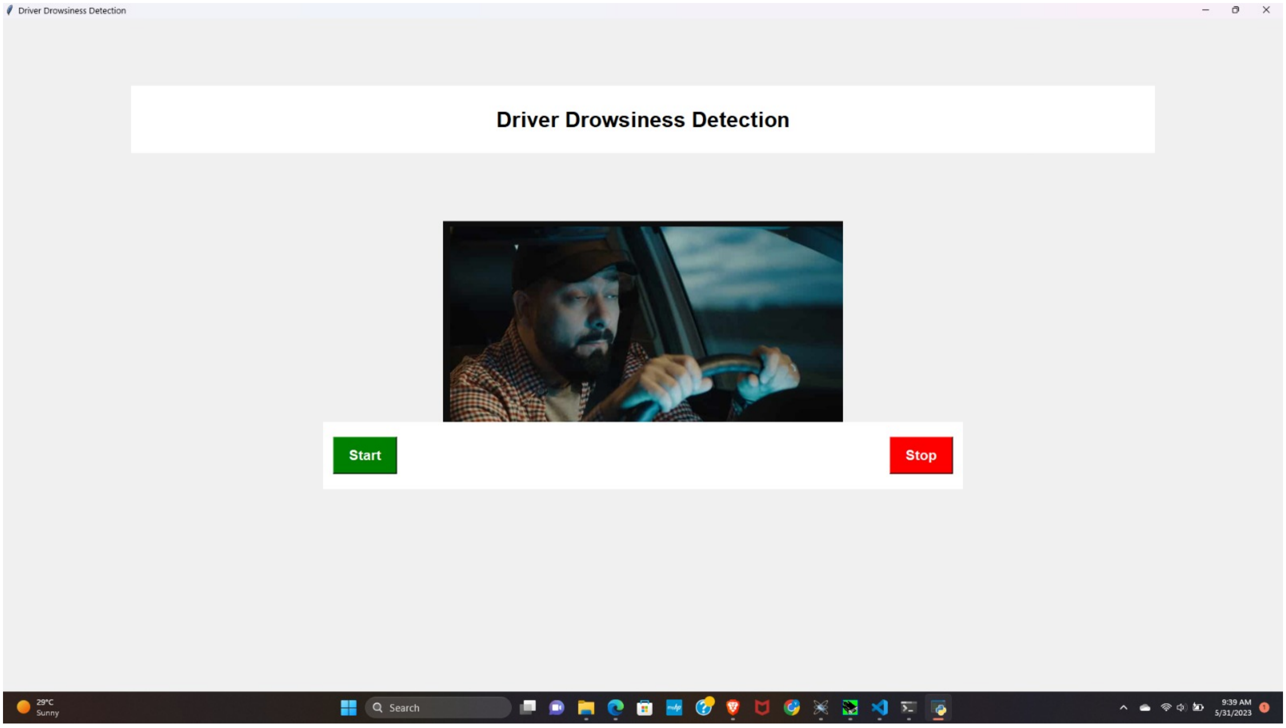


Figure 6.13: Final Output of the Drowsiness Detection System

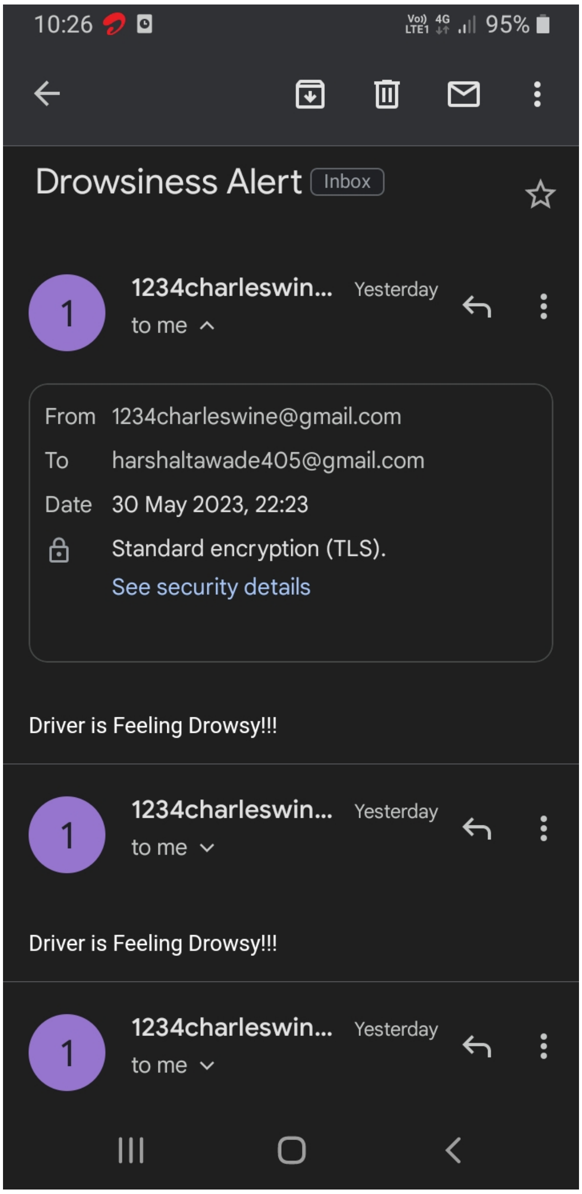


Figure 6.14: Email Received For Drowsy Driver

Chapter 7

Results and Discussion

The graph in 7.1 depicts the total time taken to stop the vehicle, as soon as the drowsiness is detected, assuming that the vehicle has been driving at a certain initial speed.

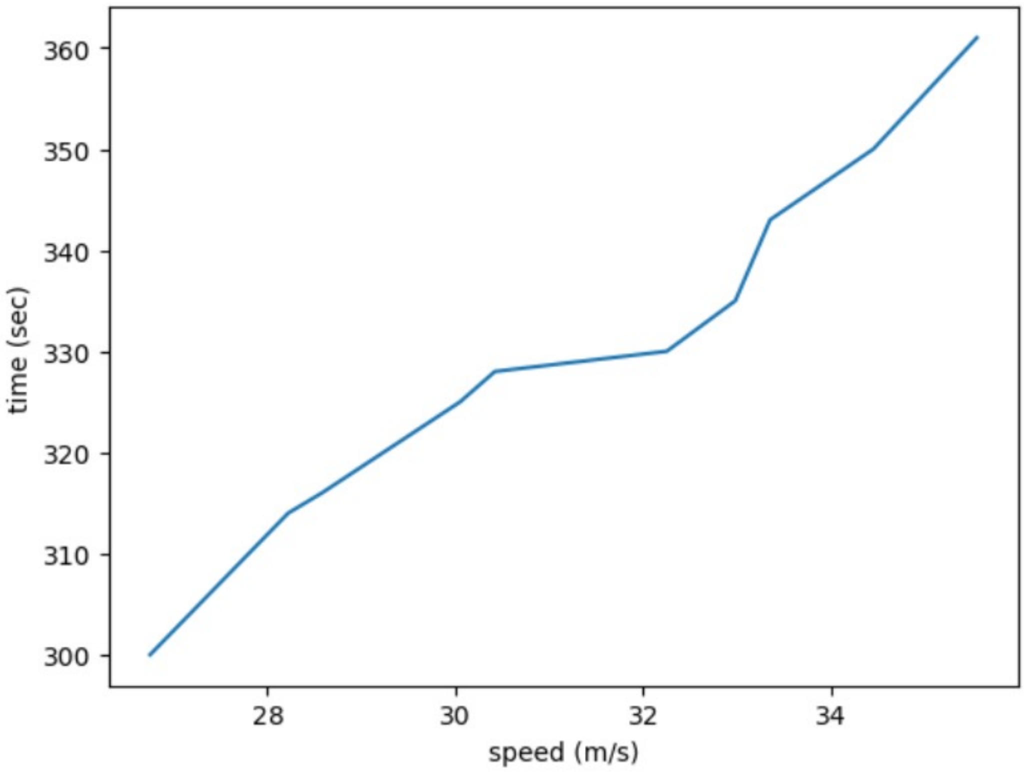


Figure 7.1: Drowsiness Detection (Sleep Alert)

The graph in 7.2 depicts the total possible distance covered by a vehicle once drowsiness is detected. It's the distance covered until the speed becomes zero and vehicle stops.

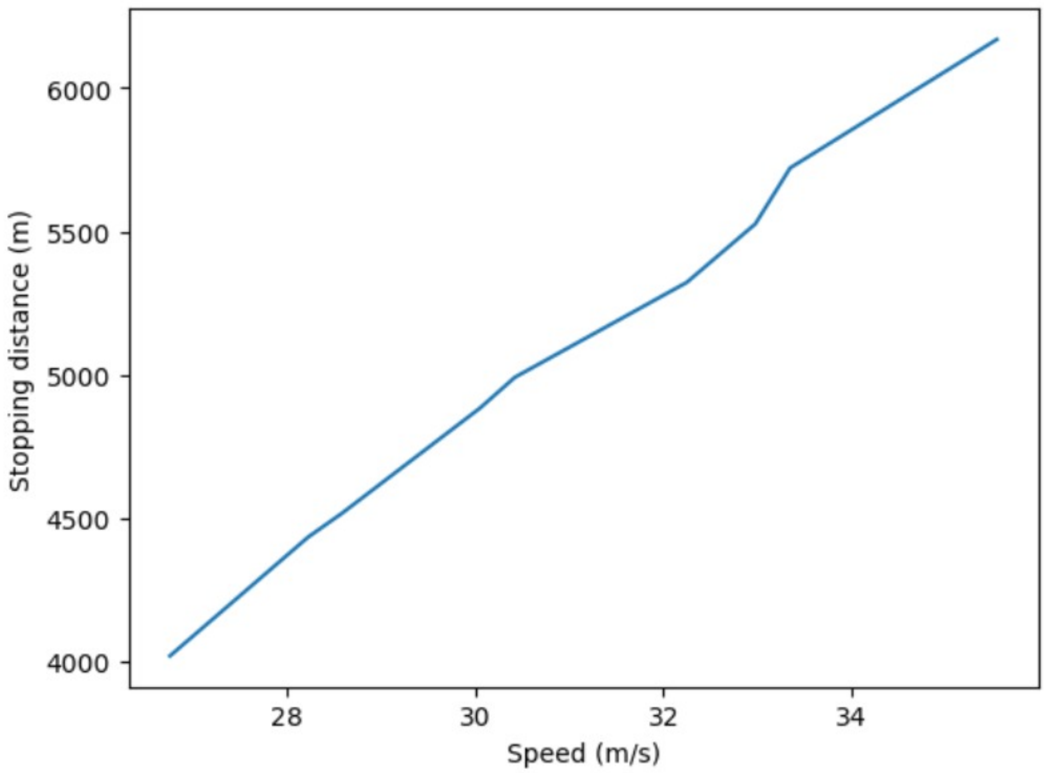


Figure 7.2: Drowsiness Detection (Sleep Alert)

As shown in Fig.7.3,the MAR (Mouth Aspect Ratio) threshold value of 30 and for EAR (Eye Aspect Ratio) threshold value is 0.20 This test can determine whether the driver is drowsy or not, where the EAR value is high and the driver is not drowsy. Similarly, a low EAR value (less than 0.20) indicates that the driver is drowsy. However, it's important to note that the EAR values can frequently change due to movements of the eyelids. Maintaining the count of eye closure and yawning can determine whether the driver is drowsy or not.

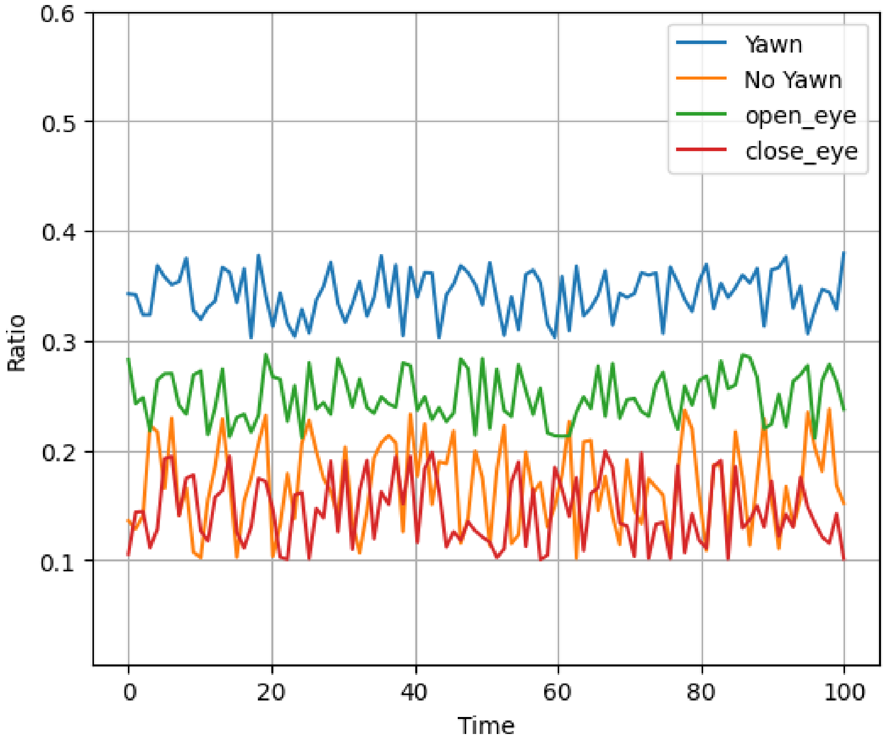


Figure 7.3: Ratios vs Time

The results presented in Table 1, provide a comprehensive overview of the performance of our proposed driver drowsiness detection system. The table categorizes the system's detections into various states, including open eyes, closed eyes, yawning, and no yawning. The system demonstrated a high level of accuracy of 96% in determining these states, with a notable correct detection rate .The few instances of incorrect detections indicate the system's robustness in distinguishing between different driver states, highlighting its reliability in real-world scenarios.

Table 7.1: Eye State and Yawning Detection Results

State	**Correct**	**Incorrect**
Open Eye	398	19
Close Eye	402	15
Yawn	397	20
No Yawn	405	12

In Fig.7.4, the focus is on close eyes detection, specifically when the Eye Aspect Ratio (EAR) falls below the threshold of 0.20, alert is generated.

In Fig.7.5,yawn detection is illustrated.When the Mouth Aspect Ratio (MAR) is above the threshold of 30, yawn is detected and alert is generated.

In Fig. 7.6, the integration of our drowsiness detection system with an alert mechanism is showcased. The figure illustrates the initiation of an alert mail notification in response to the system's detection of driver drowsiness.

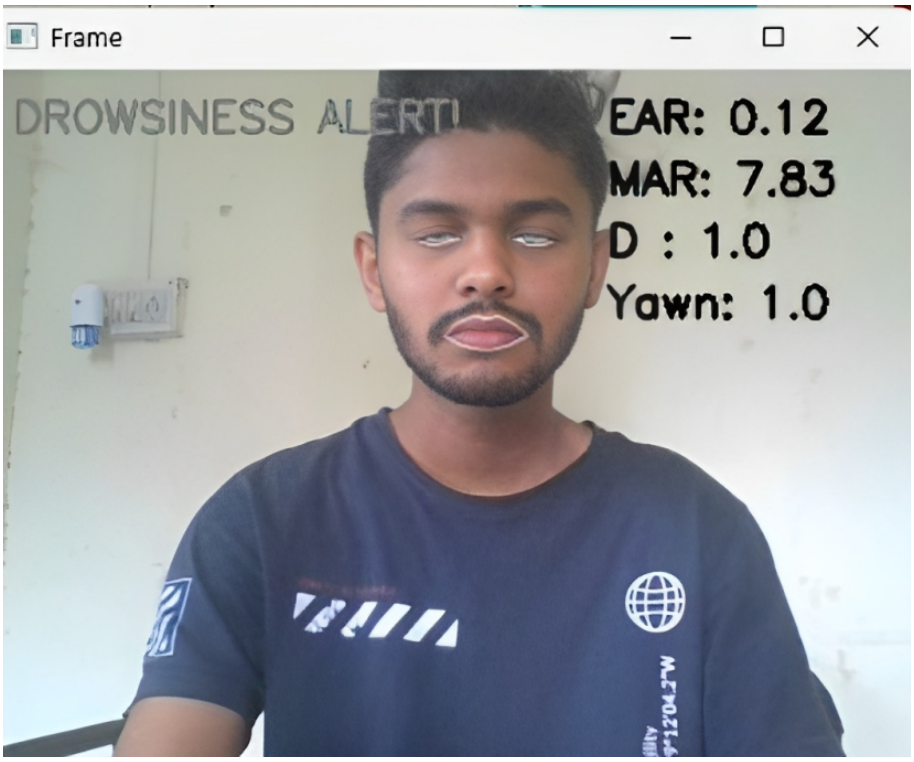


Figure 7.4: Close Eyes Detection

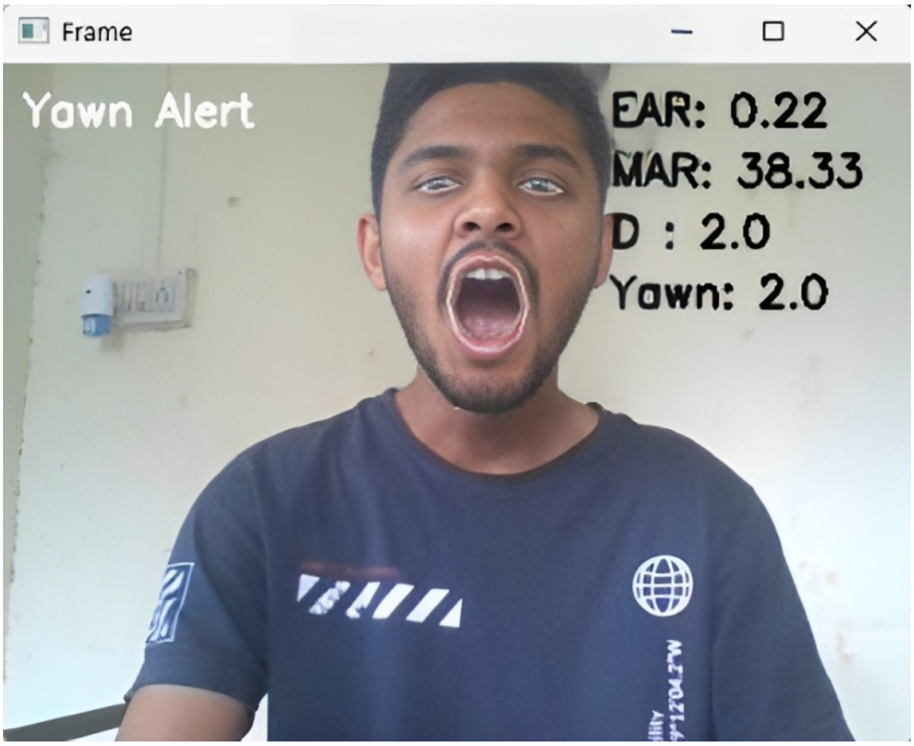


Figure 7.5: Yawn Detection

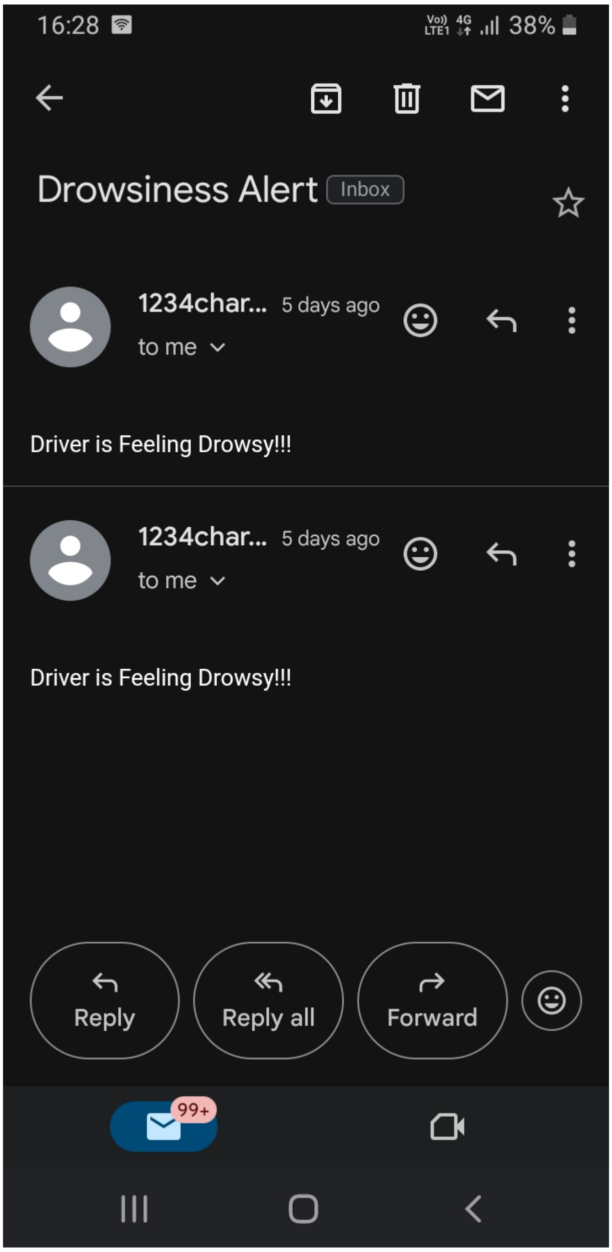


Figure 7.6: Mail Alert

Chapter 8

Conclusion

8.1 Conclusion

The advanced drowsiness detection system represents a remarkable technological leap in ensuring road safety by swiftly identifying signs of drowsiness in drivers. Employing sophisticated algorithms and cutting-edge technology, this system demonstrates an unparalleled ability to discern between regular eye blinks and the telltale signs of drowsiness. This demarcation is crucial in thwarting the onset of driver fatigue, a major factor contributing to road accidents.

The system operates seamlessly, employing intricate sensors and real-time data analysis to monitor the driver's eye movements. By accurately distinguishing between ordinary blinks and the extended closures indicative of drowsiness, it provides a prompt alert to the driver. This timely intervention acts as a vital safeguard, preventing the driver from succumbing to a state of sleepiness while behind the wheel.

In essence, the drowsiness detection system serves as a vigilant co-pilot, offering a preemptive defense against the potential hazards associated with driver fatigue. Its proactive nature contributes significantly to averting road accidents caused by lapses in attentiveness due to drowsiness. By addressing this critical aspect of driver safety, the system plays a pivotal role in promoting responsible and secure driving practices.

8.2 Future Scope

The future trajectory of research in drowsiness detection systems appears promising, with a potential focus on incorporating external factors to enhance accuracy and broaden the scope of detection. Integrating information such as vehicle statuses, sleeping patterns, weather conditions, and mechanical data into the existing system could provide a more comprehensive and nuanced understanding of a driver's state. This holistic approach may significantly improve the system's ability to preemptively identify drowsiness by considering a broader array of contextual factors.

Appendices

References

Altameem, A., Kumar, A., Poonia, R. C., Kumar, S., & Saudagar, A. K. J. (2021). Early identification and detection of driver drowsiness by hybrid machine learning. *IEEE Access*, *9*, 162805-162819. doi: 10.1109/ACCESS.2021.3131601

AlZu'bi, H. S., Al-Nuaimy, W., & Al-Zubi, N. S. (2013). Eeg-based driver fatigue detection. In *2013 sixth international conference on developments in esystems engineering* (p. 111-114). doi: 10.1109/DeSE.2013.28

Arefnezhad, S., Samiee, S., Eichberger, A., & Nahvi, A. (2019, 02). Driver drowsiness detection based on steering wheel data applying adaptive neuro-fuzzy feature selection. *Sensors*, *19*, 943. doi: 10.3390/s19040943

Automatic detection of alertness/drowsiness from physiological signals using wavelet-based nonlinear features and machine learning. (2015). *Expert Systems with Applications*, *42*(21), 7344-7355. doi: https://doi.org/10.1016/j.eswa.2015.05.028

Awais, M., Badruddin, N., & Drieberg, M. (2017, 08). A hybrid approach to detect driver drowsiness utilizing physiological signals to improve system performance and wearability. *Sensors*, *17*, 1991. doi: 10.3390/s17091991

Doudou, M., & Bouabdallah, A. (2018, 01). Performance specifications of market physiological sensors for efficient driver drowsiness detection system. In *Proceedings of the 7th international conference on sensor networks* (p. 99-106). doi: 10.5220/0006607800990106

Feng, R., Zhang, G., & Cheng, B. (2009). An on-board system for detecting driver drowsiness based on multi-sensor data fusion using dempster-shafer theory. In *2009 international conference on networking, sensing and control* (p. 897-902). doi: 10.1109/ICNSC.2009.4919399

Forsman, P., Vila, B., Short, R., Mott, C., & Dongen, H. (2012, 05). Efficient driver drowsiness detection at moderate levels of drowsiness. *Accident; analysis and prevention*, *50*. doi: 10.1016/j.aap.2012.05.005

He, J., Choi, W., Wu, X., & Yang, Y. (2015, 09). Detection of operator drowsiness using google glass. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 1607-1611. doi: 10.1177/1541931215591348

Jacobé de Naurois, C., Bourdin, C., Stratulat, A., Diaz, E., & Vercher, J.-L. (2017, 12). Detection and prediction of driver drowsiness using artificial neural network models. *Accident; analysis and prevention*, *126*. doi: 10.1016/j.aap.2017.11.038

Katsuki, T., Zhao, K., & Yoshizumi, T. (2020). Learning to estimate driver drowsiness from car acceleration sensors using weakly labeled data. In *Icassp 2020 - 2020 ieee international conference on acoustics, speech and signal processing (icassp)* (p. 3002-3006). doi: 10.1109/ICASSP40776.2020.9053100

Kumar, A., & Patra, R. (2018). Driver drowsiness monitoring system using visual behaviour and machine learning. In *2018 ieee symposium on computer applications industrial electronics (iscaie)* (p. 339-344). doi: 10.1109/ISCAIE.2018.8405495

Liu, C. C., Hosking, S. G., & Lenné, M. G. (2009). Predicting driver drowsiness using vehicle measures: Recent insights and future challenges. *Journal of Safety Research*, *40*(4), 239-245. doi: https://doi.org/10.1016/j.jsr.2009.04.005

Miranda, M., Villanueva, A., Buo, M. J., Merabite, R., Perez, S. P., & Rodriguez, J. M. (2018). Portable prevention and monitoring of driver's drowsiness focuses to eyelid movement using internet of things. In *2018 ieee 10th international conference on humanoid, nanotechnology, information technology,communication and control, environment and management (hnicem)* (p. 1-5). doi: 10.1109/HNICEM.2018

.8666334

Omidyeganeh, M., Javadtalab, A., & Shirmohammadi, S. (2011). Intelligent driver drowsiness detection through fusion of yawning and eye closure. In *2011 ieee international conference on virtual environments, human-computer interfaces and measurement systems proceedings* (p. 1-6). doi: 10.1109/VECIMS.2011.6053857

Picot, A., Charbonnier, S., & Caplier, A. (2012, 05). On-line detection of drowsiness using brain and visual information. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *42*, 1 - 12. doi: 10.1109/TSMCA.2011.2164242

Pranowo, D., Artanto, D., & Sulistyanto, M. (2019, 04). Non-intrusive vehicle-based measurement system for drowsiness detection. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *17*, 956. doi: 10.12928/telkomnika.v17i2.11759

Pratama, B. G., Ardiyanto, I., & Adji, T. B. (2017). A review on driver drowsiness based on image, bio-signal, and driver behavior. In *2017 3rd international conference on science and technology - computer (icst)* (p. 70-75). doi: 10.1109/ICSTC.2017.8011855

Pritchett, S., Zilberg, E., Xu, Z., Karrar, M., Burton, D., & Lal, S. (2011, 11). Comparing accuracy of two algorithms for detecting driver drowsiness — single source (eeg) and hybrid (eeg and body movement). *IB2COM 2011 - 6th International Conference on Broadband Communications and Biomedical Applications, Program*. doi: 10.1109/IB2Com.2011.6217916

Valsan A, V., Mathai, P. P., & Babu, I. (2021). Monitoring driver's drowsiness status at night based on computer vision. In *2021 international conference on computing, communication, and intelligent systems (icccis)* (p. 989-993). doi: 10.1109/ICCCIS51004.2021.9397180

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 ieee computer society conference on computer vision and pattern recognition. cvpr 2001* (Vol. 1, p. I-I). doi: 10.1109/CVPR.2001.990517

report

ORIGINALITY REPORT

10%	5%	15%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to MIT Academy of Engineering Student Paper	4%
2	www.ijert.org Internet Source	2%
3	www.coursehero.com Internet Source	1%
4	J S Femilda Josephin, C Lakshmi, S Joseph James. "A review on the measures and techniques adapted for the detection of driver drowsiness", IOP Conference Series: Materials Science and Engineering, 2020 Publication	1%
5	ijcspub.org Internet Source	1%
6	V Uma Maheswari, Rajanikanth Aluvalu, MVV Prasad Kantipudi, Krishna Keerthi Chennam, Ketan Kotecha, Jatinderkumar R. Saini. "Driver Drowsiness Prediction Based on Multiple Aspects Using Image Processing Techniques", IEEE Access, 2022 Publication	1%

7	Md. Ebrahim Shaik. "A systematic review on detection and prediction of driver drowsiness", Transportation Research Interdisciplinary Perspectives, 2023 Publication	1 %
8	medium.com Internet Source	1 %
9	Submitted to Coventry University Student Paper	1 %
10	dergipark.org.tr Internet Source	1 %
11	etrr.springeropen.com Internet Source	1 %
12	Liu Yang, Qianxi Zhao. "An aggressive driving state recognition model using EEG based on stacking ensemble learning", Journal of Transportation Safety & Security, 2023 Publication	1 %
13	hal.archives-ouvertes.fr Internet Source	1 %
14	Submitted to University of Sheffield Student Paper	1 %
15	www.scribd.com Internet Source	<1 %
16	wrap.warwick.ac.uk	

	Internet Source	<1%
17	Javier O. Garcia, Justin Brooks, Scott Kerick, Tony Johnson, Tim R. Mullen, Jean M. Vettel. "Estimating direction in brain-behavior interactions: Proactive and reactive brain states in driving", NeuroImage, 2017 Publication	<1%
18	Submitted to Leeds Beckett University Student Paper	<1%
19	Submitted to Study Group Australia Student Paper	<1%
20	diglib.tugraz.at Internet Source	<1%
21	E. Aidman, K. Johnson, G.M. Paech, C. Della Vedova et al. "Caffeine reduces the impact of drowsiness on driving errors", Transportation Research Part F: Traffic Psychology and Behaviour, 2018 Publication	<1%
22	R. N. Ashlin Deepa, DontiReddy Sai Rakesh Reddy, K. Milind, Y. Vijayalata, Kamishetty Rahul. "Chapter 61 Drowsiness Detection Using IoT and Facial Expression", Springer Science and Business Media LLC, 2023 Publication	<1%

23	Submitted to INTI Universal Holdings SDM BHD Student Paper	<1%
24	pdfslide.net Internet Source	<1%
25	Submitted to Bournemouth University Student Paper	<1%
26	iopscience.iop.org Internet Source	<1%
27	www.elkraft.chalmers.se Internet Source	<1%
28	Submitted to Higher Education Commission Pakistan Student Paper	<1%
29	Submitted to University of Nottingham Student Paper	<1%
30	Submitted to An-Najah National University Student Paper	<1%
31	link.springer.com Internet Source	<1%
32	www.mdpi.com Internet Source	<1%
33	www.researchgate.net Internet Source	<1%

34	Submitted to Engineers Australia Student Paper	<1 %
35	dokumen.pub Internet Source	<1 %
36	www.science.gov Internet Source	<1 %
37	Submitted to Sogang University Student Paper	<1 %
38	T. Srilakshmi, Harshavardhan Reddy, Yaswanthi Potluri, Lakshmi Ramani Burra, Mohana Vamsi Thota, Rishita Gundimeda. "Automated Driver Drowsiness Detection System using Computer Vision and Machine Learning", 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2023 Publication	<1 %
39	iplab.sakura.ne.jp Internet Source	<1 %
40	penerbit.uthm.edu.my Internet Source	<1 %
41	odr.chalmers.se Internet Source	<1 %
42	pure.tue.nl Internet Source	<1 %

43	Submitted to Flinders University Student Paper	<1 %
44	Mohammad Ehsanul Alim, Sarosh Ahmad, Marzieh Naghdi Dorabati, Ihab Hassoun. "Design & Implementation of IoT Based Smart Helmet for Road Accident Detection", 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2020 Publication	<1 %
45	Submitted to University of Hertfordshire Student Paper	<1 %
46	www.ijraset.com Internet Source	<1 %
47	Submitted to University at Buffalo Student Paper	<1 %
48	archives.univ-biskra.dz Internet Source	<1 %
49	sachinshetty.webs.com Internet Source	<1 %
50	ijircce.com Internet Source	<1 %
51	mecs-press.net Internet Source	<1 %
52	www.hindawi.com	

IOT-Integrated System for Drowsiness detection using Facial Landmarks

1st Prasad Chavan
Dept. of Computer Engineering
MIT Academy of Engineering
Pune, India
prasad.chavan@mitaoe.ac.in

3rd Aditya Gaikwad
Dept. of Computer Engineering
MIT Academy of Engineering
Pune, India
aditya.gaikwad@mitaoe.ac.in

2nd Harshal Tawade
Dept. of Computer Engineering
MIT Academy of Engineering
Pune, India
harshal.tawade@mitaoe.ac.in

4th Yash Gaikwad
Dept. of Computer Engineering
MIT Academy of Engineering
Pune, India
yash.gaikwad@mitaoe.ac.in

5th Vinodini Gupta
Dept. of Computer Engineering
MIT Academy of Engineering
Pune, India
vinodini.gupta@mitaoe.ac.in

***Abstract*—Driver drowsiness is a significant cause of road accidents, resulting in injuries and fatalities around the world. Traditional methods of detecting driver drowsiness such as monitoring the steering wheel or detecting lane deviation have limitations, and may not be reliable in all situations. Therefore, there is a need for an efficient and accurate driver drowsiness detection system that can help prevent accidents caused by driver fatigue. With the advent of Computer Vision and Internet of Things (IoT), it is now possible to develop such a system that utilizes various sensors and devices to monitor the driver's state and take appropriate actions. In this paper, we propose an IoT integrated facial landmarks based driver drowsiness detection system that uses facial landmark points to detect and monitor the driver's state. The system employs the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) to determine if the driver's eyes are open, if they are yawning, or if they are showing signs of sleepiness. In cases where the driver is deemed drowsy, the system triggers an alarm and alerts, thereby ensuring the safety of the driver and other road users. The proposed system is cost-effective and can be integrated into existing vehicle systems, making it a practical solution for detecting driver drowsiness. This paper aims to provide a detailed description of the proposed system, its implementation, and evaluation, and to demonstrate its effectiveness in detecting driver drowsiness in real-world scenarios.**

***Keywords*—Drowsiness Detection, Iot, OpenCV, Eye Aspect Ratio, Mouth Aspect Ratio, Raspberry pi, Driver Fatigue,**

I. INTRODUCTION

Driver drowsiness is a significant cause of road accidents, resulting in injuries and fatalities around the world.Driver drowsiness refers to a state of being sleepy or fatigued while driving. It can happen due to various reasons, such as lack of sleep, long hours of driving, medications, and medical conditions that cause fatigue.Drowsiness raises the risk of collision/near-collision by a factor of four, plays a significant role in accidents (12%) and near-collisions (10%), and is a contributing factor in more than 22% to 24% of crashes/near-collisions [1].When a driver is drowsy, their reaction time is slower, and their ability to make good decisions can be impaired. This can cause them to miss important cues on the road, such as traffic signals, road signs, and other vehicles, and make errors while driving.
Drowsiness is a common problem that affects people from all walks of life, and can have serious consequences in certain situations, such as driving. The inability to detect drowsiness in time can result in accidents, injuries, and even fatalities. Therefore,

the development of an effective and reliable drowsiness detection system is essential to prevent such incidents. Traditional methods of detecting driver drowsiness such as monitoring the steering wheel or detecting lane deviation have limitations, and may not be reliable in all situations. Therefore, there is a need for an efficient and accurate driver drowsiness detection system that can help prevent accidents caused by driver fatigue.In recent years, various approaches have been proposed to detect drowsiness, ranging from physiological monitoring to behavioral analysis. Among these, computer vision-based approaches have gained popularity due to their non-invasive nature and ease of deployment. With the advent of the Internet of Things (IoT), it is now possible to develop such a system that utilizes various sensors and devices to monitor the driver's state and take appropriate actions.

In this paper, we propose an IoT integrated driver drowsiness detection system that uses facial landmark points to detect and monitor the driver's state. The system employs the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) to determine if the driver's eyes are open, if they are yawning, or if they are showing signs of sleepiness. In cases where the driver is deemed drowsy, the system triggers an alarm and reduces the speed of the vehicle, thereby ensuring the safety of the driver and other road users. The proposed system is cost-effective and can be integrated into existing vehicle systems, making it a practical solution for detecting driver drowsiness.

This paper aims to provide a detailed description of the proposed system, its implementation, and evaluation, and to demonstrate its effectiveness in detecting driver drowsiness in real-world scenarios.The rest of the paper is organized as follows: Section 2 presents a literature review. Section 3 describes the components used to design the proposed system. Section 4 presents a block diagram of the proposed system and Section 5 provides a brief explanation of the methodology with a flowchart. Section 6 discusses the result analysis. Finally, Section 7 provides a conclusion of the paper, along with future work.

II. Background Study

In this section, we delve into a comprehensive background study of driver drowsiness detection methods, exploring a diverse array of techniques and research findings that have paved the way for our proposed IoT-based system.

A. Physiological Signals and EEG-based Detection

Antoine Picot et al. [] introduced a drowsiness detection system that relies on physiological signals, specifically EEG and EOG. Their approach involved tracking brain activity using EEG and employing fuzzy logic for sleepiness detection. They also utilized blinking detection through EOG channels, extracting blinking characteristics. This method provided insights into the neurological aspects of drowsiness, offering a deeper understanding of the driver's cognitive state.

B. Wearable Devices and Eye Blink Monitoring

In a study by William Choi et al. [], the focus was on wearable technology, specifically Google Glass, equipped with a proximity sensor. The sensor tracked eye blink frequency as an indicator of drowsiness. The research involved a group of drivers, both male and female, in real-world driving scenarios, demonstrating the feasibility of using wearable devices for early drowsiness detection. This approach illustrates the potential of integrating everyday technology into safety systems.

C. Computer Vision and Facial Landmark Detection

Valsan A. et al. [] presented a real-time video-based system that capitalizes on computer vision techniques. Their method involved capturing the driver's face through a digital camera and applying image processing to locate facial landmarks. By analyzing ratios related to eye and mouth movements, the system detected signs of drowsiness. This approach showcased the power of computer vision in non-invasive drowsiness detection, offering real-time monitoring capabilities.

D. Machine Learning and Image Processing

A. Kumar et al. [] proposed a drowsiness detection system based on image processing and machine learning. Their system utilized a camera to record driver footage, with image processing techniques identifying facial landmarks. The system employed machine learning algorithms to determine drowsiness based on eye aspect ratio and mouth opening ratio values. This combination of image processing and machine learning highlighted the potential for automation in drowsiness detection, allowing for adaptability and accuracy.

E. IoT Integration and Alert Systems

Menchie Manda et al. [] introduced a technology that continually monitored eyelid movements to detect drowsiness. Once detected, an alert was issued to the driver through a sounding siren, and the report was transmitted to the car owner via internet connectivity. This approach emphasized the importance of real-time alert systems and connectivity, showcasing the potential for integrated IoT solutions in enhancing driver safety.

F. Multi-Sensor Fusion and Data Analysis

Several studies [] have explored the concept of multi-sensor fusion for drowsiness detection. These approaches combine various sensor inputs, including physiological signals, image data, and vehicle measures, to create a holistic picture of the driver's state. Machine learning algorithms and data analysis techniques are often employed to process and interpret the integrated data. These studies demonstrate the potential for comprehensive and robust drowsiness detection systems that consider multiple aspects of driver behavior and physiology.

The field of driver drowsiness detection encompasses a wide range of approaches, from physiological signal analysis to computer vision, machine learning, wearables, and IoT integration. Each approach contributes unique insights and methods to address the critical issue of driver drowsiness, ultimately aiming to enhance road safety. The proposed IoT-based system in this paper builds upon these concepts to provide an efficient and practical solution for detecting driver drowsiness in real-world scenarios.

III. Methodology

The suggested approach aims to reduce the number of accidents brought on by sleepy drivers. Measurements are made of the driver's face and eye movements. Without any sign of depletion, an EAR's threshold value is over 0.25. The quantity of video frames with the driver's eyes closed is represented by the threshold value of a sleepy eye blink sample. The driver's tiredness is identified if the number of consecutive counting frames rises beyond the threshold value range.

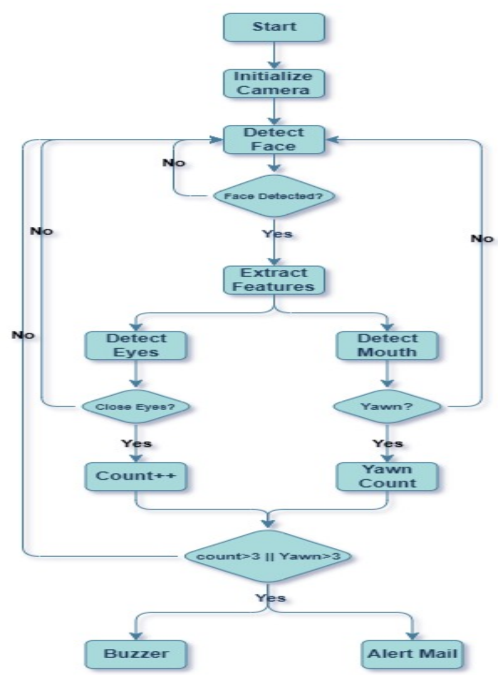


Fig. 1. Flow Diagram

The proposed methodology includes following steps:

Real Time Video Capturing: In the proposed method, first step is to capture video of the driver. For this purpose a camera is used which will capture the real time video of driver. This camera is connected to raspberry pi where further processing will be execute.

Preprocessing: Frame extraction and face detection are two key preprocessing steps in driver drowsiness systems. These steps are crucial for ensuring that the input data is properly prepared for subsequent analysis. Frame extraction involves taking a video stream and extracting individual frames that can be analyzed separately. This process typically involves a video processing library, such as OpenCV, that can read and manipulate video files. Each frame is essentially a static image that captures a moment in time from the video stream. By analyzing each frame individually, the system can identify patterns and changes in the driver's behavior over time, such as changes in eyes and mouth movements. Then the next step is Grayscale conversion. Grayscale conversion is a common preprocessing step in drowsiness detection systems that involve analyzing images or video frames. The purpose of this step is to convert a color image into a grayscale representation that can be more easily analyzed by computer vision algorithms.

In a color image, each pixel is represented by a combination of red, green, and blue (RGB) values. This can make it difficult to analyze the image, as there are many different combinations of colors that can occur. Grayscale conversion simplifies the image by reducing each pixel to a single value that represents its brightness or intensity.

In a grayscale image, each pixel is represented by a single value between 0 and 255 that indicates its brightness. A value of 0 represents black, while a value of 255 represents white. Values in between represent shades of gray. By converting a color image to grayscale, the system can more easily analyze features such as edges and shapes, which can be important for drowsiness detection. Overall, grayscale conversion is an important step in the preprocessing pipeline for drowsiness detection systems, as it simplifies the image and makes it easier to extract relevant features for subsequent analysis.

Next step is Face detection. It is a critical preprocessing step in many computer vision applications, including driver drowsiness systems. One popular algorithm for face detection is the Haar Cascade classifier, which uses a set of pre-trained cascading classifiers to detect faces in an image or video frame.

The Haar Cascade classifier works by breaking down an image into a series of sub-regions, or windows, and analyzing each window to determine whether it contains a face. Each window is analyzed using a series of classifiers, which are essentially decision trees that evaluate different features of the image to determine whether it contains a face.

The features used by the Haar Cascade classifier are derived from the differences in intensity between adjacent regions of the image. These features might include the contrast between the eyes and the surrounding area, or the difference in color between the skin and the background. By analyzing these features across multiple cascading classifiers, the algorithm is able to detect faces with high accuracy.

The Haar Cascade classifier is particularly effective at detecting faces because it is able to take into account the variability in face size, orientation, and lighting conditions that can occur in real-world images. This makes it well-suited for driver drowsiness detection, where lighting conditions and other factors can vary widely. Overall, the Haar Cascade classifier is a powerful tool for face detection in computer vision applications. By using a set of pre-trained cascading classifiers to analyze image features, it is able to detect faces with high accuracy and robustness, even in challenging conditions.

Facial Landmarks Extraction:

Facial landmark detection is a technique used to identify and locate specific points. These points, also known as landmarks points, represent key features of the face,as depicted in Fig.2, such as the eyes, nose, mouth, and jawline. Facial landmark detection is an essential task in many computer vision applications, including face recognition, emotion detection, and facial expression analysis.

Dlib is a popular open-source library for machine learning, which includes a pre-trained facial landmark detector that uses a deep learning-based method to identify and locate the facial landmarks accurately. The dlib face landmark detector is based on a pre-trained convolutional neural network (CNN) that is capable of detecting up to 68 different facial landmarks. The landmark detector is capable of detecting both frontal and profile faces with high accuracy.

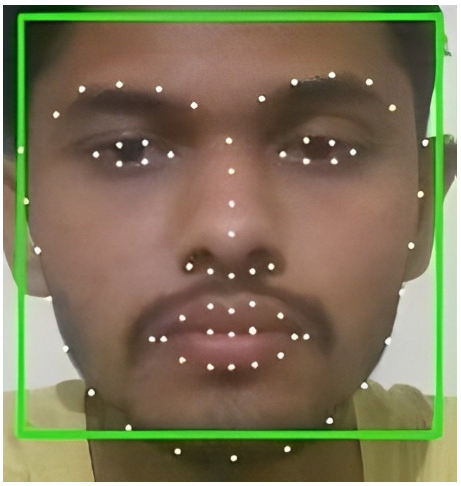


Fig. 2. Facial Landmarks

EAR and MAR Calculation:

EAR (Eye Aspect Ratio) and MAR (Mouth Aspect Ratio) are two commonly used measures in facial landmark detection and analysis for various applications such as fatigue detection, eye closure detection, and emotion recognition.

EAR measures the eye openness and blink rate, while MAR measures the mouth openness. The calculation of these ratios involves the detection of specific facial landmarks, such as the corners of the eyes and mouth.When a person is drowsy, their eyelids tend to drop, which can cause the EAR to decrease. In contrast, when a person is alert, their eyes tend to be more open, and the EAR will increase.

The EAR calculation involves measuring the ratio of the distance between the vertical landmarks of the eye (the outer corner of the eye and the inner corner of the eye) as depicted in Fig.3, to the distance between the horizontal landmarks of the eye (the upper eyelid and the lower eyelid). A threshold value is usually set to determine when the eyes are closed or when the person is blinking. Right eye using [37,42] and Left eye using [43,45]. The following formula is used to calculate the eye aspect ratio. There are a total of 6 spots on the eyes: 2 on the top eyelid, 2 on the lower eyelid, and 2 in the centre of the eye. In order to compute the vertical distance for eye aspect ratio, multiply the horizontal distance from the centre of the eye by two.

Fig. 3. Eye Landmarks

The formula for calculating EAR is:

$$\text{EAR} = \frac{\|P5 - P1\| - \|P4 - P2\|}{2\|P0 - P3\|} \qquad (1)$$

When an eye is open, the EAR is above 0.20, and when it is closed, reduces below 0.20 Based on the EAR computed in the preceding stage, the choice on the eye state is taken. The eye condition is categorized as "closed" if the distance is zero or nearly zero; otherwise, it is categorized as "open."

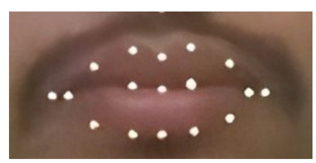


Fig. 4. Mouth Landmarks

We can detect mouth by using landmarks [61,68] as depicted in Fig.4 A formula that is presented below is used to determine the mouth aspect ratio. To do this, we need eight locations on the lips as references: two on the upper, two on the lower, and two in the centre. The vertical distance in the mouth aspect ratio is determined by dividing by two the horizontal distance from the centre of the mouth. When a person is alert, their mouth tends to be less open, which can cause the MAR to decrease. In contrast, when a person is yawning, their mouth tends to be more open, and the MAR will increase.

The formula for calculating MAR is:

$$MAR = \frac{||P2 - P8|| - ||P3 - P7|| - ||P4 - P6||}{3||P1 - P5||} \tag{2}$$

Drowsiness Detection:

Now Eye aspect ratio and mouth aspect ratio are the two important parameters that will help us determine whether a person is feeling sleepy or not. Eye aspect ratio and Mouth aspect ratio have their threshold values. We have taken the threshold value for EAR to be 0.20 and for MAR, it is 0.30. In this system, we are calculating EAR and MAR for each frame of video. If the EAR decreases below the threshold for a continuous 30 frames, then the system will give a speaker alert to the driver, thereby alerting the driver to their drowsy state. Additionally, if the MAR increases above the threshold, a speaker alert will be given to the driver.

The system also maintains a count of eye closures and yawns. If either of these counts goes greater than 3, the system will conclude that the person is feeling drowsy and will play a buzzer to alert both the driver and other passengers. Notably, as part of our innovative approach, we propose integrating a mechanism to reduce the speed of the vehicle when drowsiness is detected. While reducing the vehicle's speed, it is possible that other vehicles may be trailing behind. To address this concern, an ultrasonic sensor will detect the presence of other vehicles and issue an alert to them as well.

IV. Experimental Setup

A. Simulation Scenario

In order to validate the effectiveness of the proposed driver drowsiness detection system,as shown in Fig.5,a simulation was conducted using Proteus, a comprehensive simulation and microcontroller design software. The simulation focused on two critical aspects of the system: alarming the buzzer to alert the driver and passengers, and if other vehicles are around then notify them also. The simulation scenario begins with the assumption that the driver is experiencing drowsiness, as detected by the EAR (Eye Aspect Ratio) and MAR (Mouth Aspect Ratio) calculations, falling below the predetermined threshold values. When the system determines that the driver is drowsy, it triggers an alert in the form of a buzzer. This alert is crucial in awakening the driver and alerting other passengers of the potential danger. To simulate this scenario, we integrated a virtual buzzer component in Proteus. When the drowsiness detection algorithm reached the threshold criteria, a signal was sent to the virtual buzzer, initiating an audible alarm. This simulation allowed us to verify that the alert mechanism operates as intended and that the buzzer effectively serves its purpose in notifying the driver and passengers of the driver's drowsy state.

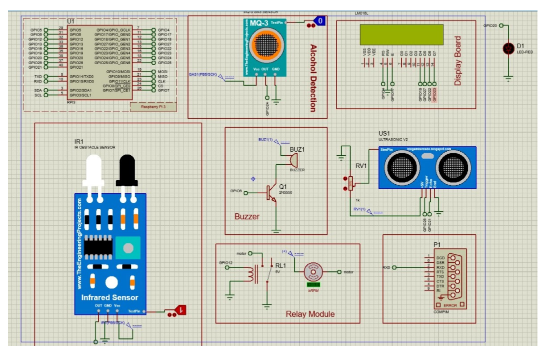


Fig. 5. Circuit Simulation

In addition to alarming the buzzer, the proposed methodology includes an innovative approach to enhancing driver safety by reducing the speed of the vehicle when drowsiness is detected. This novel feature was simulated in Proteus using a simulated motor to represent the vehicle. Upon detecting drowsiness, the system initiated a signal to reduce the speed of the vehicle. The virtual motor in the simulation was programmed to gradually decelerate, simulating the reduction in speed. We analyzed how this feature operates and observed the effects on the virtual vehicle in terms of slowing down and eventually coming to a stop. We also considered various parameters such as the rate of deceleration and the time taken to bring the vehicle to a complete halt.

Another critical aspect of our simulation involved the use of an ultrasonic sensor for detecting the presence of other vehicles trailing behind the drowsy driver. This safety feature is essential to prevent potential accidents caused by sudden speed reduction. We integrated a virtual ultrasonic sensor in Proteus to mimic the detection of other vehicles. When a trailing vehicle is detected, the system initiates an alert signal, which was visually represented in the simulation. This enabled us to assess the capability of the system to detect other vehicles, ensuring that it provides timely alerts to them, enhancing overall road safety.

The comprehensive simulation of these critical elements of the proposed driver drowsiness detection system in Proteus provides valuable insights into the system's functionality and effectiveness. It allows us to verify that the alarming mechanisms, speed reduction, and vehicle detection features work in a coordinated manner to enhance driver safety and prevent accidents due to drowsy driving.

B. Software components

Python is required for the implementation of the drowsiness detection system. Specifically, Python 3 libraries such as NumPy, Pygame, Dlib, Imutils, and OpenCV are needed for the system to function properly.

NumPy is a library used for scientific computing and is used in this system for its numerical processing capabilities. Pygame is a set of Python modules used for game development and is used in this system for its ability to display images and video.

Dlib is a modern C++ toolkit used for machine learning and computer vision applications, which includes face detection and facial landmark detection. Imutils is a set of convenience functions to make basic image processing functions such as translation, rotation, resizing, and displaying images easier with OpenCV.

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library used for real-time computer vision applications. In the drowsiness detection system, OpenCV is used for image and video processing.

The system can be implemented on either Windows or Ubuntu operating systems, as long as the required software and libraries are installed properly.

V. Results

This section presents a successful experimental result that was attained using the suggested approach while driving. The driver's EAR and MAR are observed in order to determine their drowsiness. This method indicates if the eyes are open or closed.

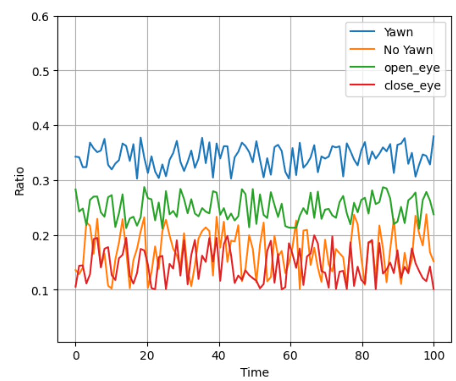


Fig. 6. Ratios vs Time

As shown in Fig.6,the MAR (Mouth Aspect Ratio) threshold value of 30 and for EAR (Eye Aspect Ratio) threshold value is 0.20 This test can determine whether the driver is drowsy or not, where the EAR value is high and the driver is not drowsy. Similarly, a low EAR value (less than 0.20) indicates that the driver is drowsy. However, it's important to note that the EAR values can frequently change due to movements of the eyelids. Maintaining the count of eye closure and yawning can determine whether the driver is drowsy or not.

The results presented in Table 1, provide a comprehensive overview of the performance of our proposed driver drowsiness detection system. The

TABLE I
EYE STATE AND YAWNING DETECTION RESULTS

State	Correct	Incorrect
Open Eye	398	19
Close Eye	402	15
Yawn	397	20
No Yawn	405	12

table categorizes the system's detections into various states, including open eyes, closed eyes, yawning, and no yawning. The system demonstrated a high level of accuracy of 96% in determining these states, with a notable correct detection rate .The few instances of incorrect detections indicate the system's robustness in distinguishing between different driver states, highlighting its reliability in real-world scenarios.

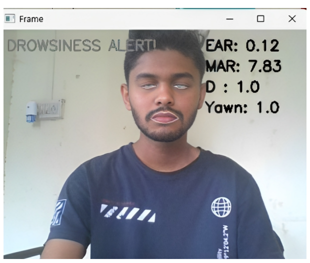


Fig. 7. Close Eyes Detection

In Fig.7, the focus is on close eyes detection, specifically when the Eye Aspect Ratio (EAR) falls below the threshold of 0.20, alert is generated.

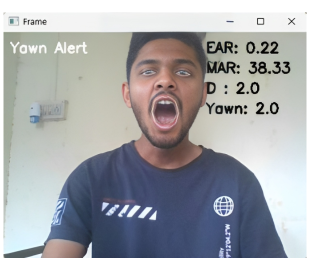


Fig. 8. Yawn Detection

In Fig.8,yawn detection is illustrated.When the Mouth Aspect Ratio (MAR) is above the threshold of 30, yawn is detected and alert is generated.

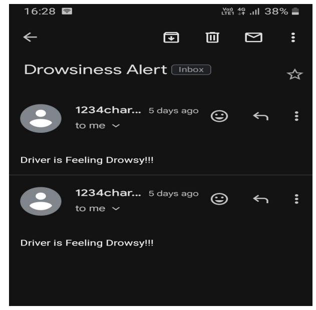


Fig. 9. Mail Alert

In Fig. 9, the integration of our drowsiness detection system with an alert mechanism is showcased. The figure illustrates the initiation of an alert mail notification in response to the system's detection of driver drowsiness.

VI. CONCLUSION AND FUTURE SCOPE

Based on the results and analysis presented in this paper, it can be concluded that the proposed driver drowsiness detection system using facial landmark points is an effective and reliable solution for detecting driver drowsiness. The system uses the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) to accurately determine the driver's state and take appropriate actions to ensure their safety and that of other road users.

Looking forward, there is scope for further improvement and enhancement of the proposed system. For example, additional sensors could be integrated to monitor other physiological signals such as heart rate and respiration to provide more accurate and reliable detection of driver drowsiness. Furthermore, machine learning algorithms could be employed to improve the accuracy of the system over time.

In conclusion, the proposed system has the potential to significantly reduce the number of accidents caused by driver drowsiness, and can be integrated into existing vehicle systems to provide a cost-effective and practical solution for detecting driver drowsiness in real-world scenarios. Further research and development in this area could lead to even more advanced and effective driver drowsiness detection systems.

References

[1] A. Picot, S. Charbonnier, and A. Caplier, ”On-Line Detection of Drowsiness Using Brain and Visual Information,” in *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 3, pp. 764-775, May 2012, doi: 10.1109/TSMCA.2011.2164242.

[2] C. C. Liu, S. G. Hosking, and M. G. Lenné, ”Predicting driver drowsiness using vehicle measures: Recent insights and future challenges,” in *Journal of Safety Research*, vol. 40, no. 4, pp. 239-245, 2009, doi: 10.1016/j.jsr.2009.04.005.

[3] V. A. Vidhu, P. P. Mathai, and I. Babu, ”Monitoring Driver’s Drowsiness Status at Night Based on Computer Vision,” in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, 2021, pp. 989-993, doi: 10.1109/ICCCIS51004.2021.9397180.

[4] A. Kumar and R. Patra, ”Driver drowsiness monitoring system using visual behaviour and machine learning,” in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018, pp. 339-344, doi: 10.1109/ISCAIE.2018.8405495.

[5] M. Omidyeganeh, A. Javadtalab, and S. Shirmohammadi, ”Intelligent driver drowsiness detection through fusion of yawning and eye closure,” in *2011 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems Proceedings*, 2011, pp. 1-6, doi: 10.1109/VECIMS.2011.6053857.

[6] M. Miranda, A. Villanueva, M. J. Buo, R. Merabite, S. P. Perez, and J. M. Rodriguez, ”Portable Prevention and Monitoring of Driver’s Drowsiness Focuses to Eyelid Movement Using Internet of Things,” in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 2018, pp. 1-5, doi: 10.1109/HNICEM.2018.8666334.

[7] A. Altameem, A. Kumar, R. C. Poonia, S. Kumar, and A. K. J. Saudagar, ”Early Identification and Detection of Driver Drowsiness by Hybrid Machine Learning,” in *IEEE Access*, vol. 9, pp. 162805-162819, 2021, doi: 10.1109/ACCESS.2021.3131601.

[8] B. G. Pratama, I. Ardiyanto, and T. B. Adji, ”A review on driver drowsiness based on image, bio-signal, and driver behavior,” in *2017 3rd International Conference on Science and Technology - Computer (ICST)*, 2017, pp. 70-75, doi: 10.1109/ICSTC.2017.8011855.

[9] M. Awais, N. Badruddin, and M. Drieberg, ”A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve System Performance and Wearability,” in *Sensors*, vol. 17, no. 9, p. 1991, 2017, doi: 10.3390/s17091991.

[10] P. Forsman, B. Vila, R. Short, C. Mott, and H. Dongen, ”Efficient driver drowsiness detection at moderate levels of drowsiness,” in *Accident; analysis and prevention*, vol. 50, 2012, doi: 10.1016/j.aap.2012.05.005.

[11] C. Jacobé de Naurois, C. Bourdin, A. Stratulat, E. Diaz, and J. Vercher, ”Detection and prediction of driver drowsiness using artificial neural network models,” in *Accident; analysis and prevention*, vol. 126, 2019, doi: 10.1016/j.aap.2017.11.038.

[12] S. Arefnezhad, S. Samiee, A. Eichberger, and A. Nahvi, ”Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection,” in *Sensors*, vol. 19, no. 4, p. 943, 2019, doi: 10.3390/s19040943.

[13] R. Feng, G. Zhang, and B. Cheng, ”An on-board system for detecting driver drowsiness based on multi-sensor data fusion using Dempster-Shafer theory,” in *2009 International Conference on Networking, Sensing and Control*, 2009, pp. 897-902, doi: 10.1109/ICNSC.2009.4919399.

[14] D. Pranowo, D. Artanto, and M. Sulistyanto, ”Non-intrusive vehicle-based measurement system for drowsiness detection,” in *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 2, p. 956, 2019, doi: 10.12928/telkomnika.v17i2.11759.

[15] T. Katsuki, K. Zhao, and T. Yoshizumi, ”Learning to Estimate Driver Drowsiness from Car Acceleration Sensors Using Weakly Labeled Data,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3002-3006, doi: 10.1109/ICASSP40776.2020.9053100.

[16] P. Viola and M. Jones, ”Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, doi: 10.1109/CVPR.2001.990517.

[17] J. He, W. Choi, X. Wu, and Y. Yang, ”Detection of Operator Drowsiness Using Google Glass,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 59, no. 1, pp. 1607-1611, 2015, doi: 10.1177/1541931215591348.

[18] M. Pratama, I. Ardiyanto, and T. B. Adji, ”Performance Specifications of Market Physiological Sensors for Efficient Driver Drowsiness Detection System,” in *Proceedings of the 7th International Conference on Sensor Networks*, 2018, pp. 99-106, doi: 10.5220/0006607800990106.