Amrita Vishwa Vidyapeetham, Bengaluru

Department of Electronics and Communication Engineering

19ECE384: Open lab

DRIVER DROWSINESS DETECTION AND WARNING SYSTEM

Submitted by

BATCH: 20ECE27

Reg. Number	Name
BL.EN.U4ECE20088	KATTAMURU KRISHNA KOUSHIK
BL.EN.U4ECE20012	ALLURI ADHARSH
BL.EN.U4ECE20096	KUSHMA GURUSWAMY

Faculty in charge

Ms. Priya. B. K

Assistant Professor

Examiner 1 **Examiner 2**

TABLE OF CONTENTS

ABSTRACT

Driver drowsiness poses a significant risk on roadways, contributing to a substantial number of accidents and fatalities. This abstract presents a driver drowsiness detection system aimed at addressing this pressing issue. By leveraging advanced technologies and statistical insights, our system aims to detect and prevent accidents caused by drowsy drivers. The statistics surrounding accidents attributed to drowsy driving are alarming. Studies indicate that drowsiness is a factor in approximately 20% of fatal road accidents. Moreover, it is estimated that drowsy driving-related accidents result in thousands of injuries and fatalities annually. These statistics underscore the urgent need for effective drowsiness detection systems to enhance driver safety and prevent tragic outcomes. To tackle this challenge, we have developed a comprehensive driver drowsiness detection system. The system utilizes a three-stage architecture to accurately identify and mitigate drowsiness episodes. The first stage employs computer vision techniques, leveraging OpenCV, to detect closed eyes in a sequence of images captured by an in-vehicle camera. Closed eyes are a reliable indicator of drowsiness and driver fatigue. Once a predetermined threshold of closed-eye instances is exceeded, the system proceeds to the next stage. In the second stage, the system incorporates a galvanic skin response (GSR) sensor, measuring the conductance value of the driver's skin. Reduced skin conductance is closely associated with drowsiness and stress levels. By continuously monitoring the driver's conductance values, the system can effectively detect physiological signs of drowsiness. If the conductance level falls below a specified threshold, it triggers the activation of the third stage. The third stage introduces an interactive response mechanism to ensure driver engagement and attentiveness. The driver is prompted to press a designated button within a defined time frame. Failure to respond within this period activates an alarm, alerting the driver and mitigating the risk of an accident. This stage serves as a critical safeguard, requiring the driver's active participation and responsiveness during instances of drowsiness.

CHAPTER 1 – INTRODUCTION

Driver drowsiness detection systems are crucial technologies aimed at mitigating the risks associated with driver fatigue. These systems utilize various sensors and algorithms to monitor the driver's physiological and behavioral indicators, such as eye movements, facial expressions, and physiological responses. By analyzing these signals in real-time, the system can detect signs of drowsiness and issue timely alerts or warnings to the driver. The primary objective of these systems is to enhance road safety by preventing accidents caused by drowsy driving, thereby reducing injuries and fatalities. Implementing effective drowsiness detection systems is a vital step towards creating safer driving environments and protecting both drivers and other road users.

1.1 Overall Background and System

1.1.1 What is Driver Drowsiness Detection System.

A driver drowsiness detection system is a technology designed to monitor and identify signs of drowsiness in drivers, aiming to prevent accidents caused by fatigue. It typically utilizes various sensors and algorithms to analyze the driver's behavior and physiological indicators. These systems often employ computer vision techniques to detect eye closure or changes in facial expressions, along with sensors such as infrared or GSR sensors to measure physiological responses like skin conductance. By continuously monitoring these parameters, the system can identify drowsiness-related patterns and issue warnings or alerts to the driver, prompting them to take appropriate actions to prevent accidents. The goal is to enhance driver safety and reduce the risk of accidents caused by driver fatigue, ultimately saving lives on the road.

1.1.2 Importance of Driver Drowsiness Detection System

Driver drowsiness detection systems are of utmost importance due to the significant impact of drowsy driving on road safety. These systems help prevent accidents by alerting fatigued drivers, reducing the risk of collisions caused by decreased alertness. By monitoring physiological and behavioral indicators, they provide timely warnings, allowing drivers to take necessary actions or rest. Driver drowsiness detection systems play a crucial role in preventing injuries, fatalities, and property damage, making roads safer for all users.

1.1.3 Applications of Driver Drowsiness Detection System

1. **Automotive Safety**: These systems are widely used in automobiles to ensure driver safety by detecting and alerting drowsiness, reducing the risk of accidents caused by fatigue.
2. **Transportation Industry**: Driver drowsiness detection systems are employed in various transportation sectors, including commercial trucks, buses, trains, and aviation, to prevent fatigue-related incidents and enhance passenger safety.
3. **Fleet Management**: Fleet operators utilize drowsiness detection systems to monitor driver fatigue levels and take proactive measures to prevent accidents, improve productivity, and reduce operational costs.
4. **Personal Safety Devices**: Drowsiness detection systems are integrated into wearable devices and smartwatches, providing individuals with real-time alerts and reminders to take breaks or rest when fatigue is detected.
5. **Medical Applications**: Driver drowsiness detection technology is also utilized in healthcare settings to monitor the drowsiness levels of patients under sedation, ensuring their safety during medical procedures and preventing adverse events.

1.2 Objective

The objective of the project is to develop a reliable and accurate system that can detect and alert drivers when they exhibit signs of drowsiness or fatigue.

1.3 Methodology

1.3.1 Hardware Tools Used:

1. **Raspberry pi 3b**
2. **Raspberry pi camera rev 1.3**
3. **ESP 8266**
4. **GSR Sensor**

1.3.2 Software tools used:

1. **VS Code**
2. **Arduino IDE**

1.4 Outcome

Successfully developed a driver drowsiness detection system based on the PERCLOS value of the driver by following this methodology and applying the specified hardware and software tools.

CHAPTER 2

2.1 SUMMARY OF STATE OF ART AND DESCRIPTION OF TOOLS

Title	Problem Addressed	Methodology	Outcome	Gap identified
System and Method for Driver Drowsiness Detection Using Behavioral and Sensor-Based Physiological Measures	To detect driver drowsiness more accurately using a combination of methods.	The proposed model combines intrusive and non-intrusive approach to increase the accuracy.	Better accuracy using the combination of methods.	Usage of physiological sensors is a tedious task.
A Smartphone-Based Drowsiness Detection and Warning System for Automotive Drivers	To detect driver drowsiness more accurately using a three-stage architecture	Three stage architecture comprises of PERCLOS, VUR, Touch response.	Better accuracy with the use of a 3-stage architecture.	Overall process is time taking.
Driver drowsiness detection and smart alerting using deep learning and IoT	To detect driver drowsiness using deep learning and IOT instead of normal image classifiers	Training a model with a set of images to detect eyes more accurately	Using deep learning increases the accuracy as the model will learn how to classify eyes as open or closed	Requires high performance systems.
A Survey on State-of-the-Art Drowsiness Detection Techniques	Survey on the present advancements and projects on driver drowsiness detection.	Gathering and organizing data from primary studies, including research papers, case studies, etc…	covers behavioral, vehicular, and physiological parameter-based techniques, highlighting their advantages and disadvantages	HMM shows lower error rates but is slower and more expensive than SVM and CNN classifiers

Deep Learning for Eye Blink Detection	threshold-based algorithm used in connected glasses has limitations in performance	Training a model with a set of images to detect eyes more accurately	Using deep learning increases the accuracy as the model will learn how to classify eyes as open or closed	Requires high performance systems.

2.2 Description of Tools

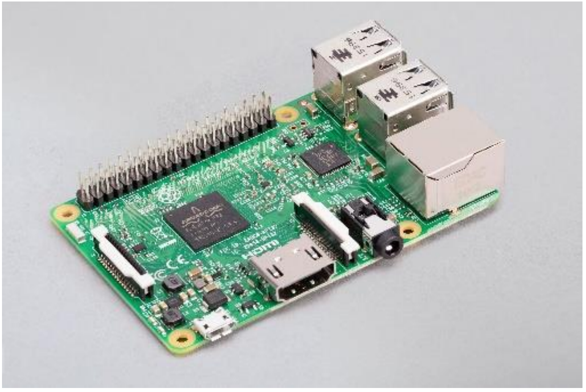
Raspberry Pi 3B: The Raspberry Pi 3B is a single-board computer that serves as the main computing platform in this project. It features a quad-core ARM Cortex-A53 processor, clocked at 1.2GHz, providing sufficient processing power for various tasks. The board is equipped with 1GB RAM, allowing for smooth multitasking and data processing.

In this project, the Raspberry Pi 3B is responsible for multiple functions. Firstly, it interfaces with the Raspberry Pi Camera Rev 1.3 to capture a 5-second video of the driver's face. The camera module connects to the Raspberry Pi's CSI (Camera Serial Interface) port, enabling seamless integration

Once the video is recorded, the Raspberry Pi 3B utilizes the pscp command to transfer the video file to a connected PC via SSH. This transfer enables further processing and analysis on the PC, leveraging its computational capabilities.

On the PC, a Python code is executed to convert the video into a set of images, which are then individually analyzed to detect open or closed eyes. The Raspberry Pi 3B communicates with the PC using SSH and enables the transfer of video data for processing.

Overall, the Raspberry Pi 3B acts as the central control unit, coordinating the video recording, data transfer, and communication with external modules. It utilizes its processing power, connectivity options, and interface compatibility to enable the various stages of the driver drowsiness detection system, ensuring efficient and accurate analysis of the driver's condition.



RASPBERRYPI 3B

<u>Raspberry Pi Camera Rev 1.3</u>: The Raspberry Pi Camera Rev 1.3 is a compact camera module specifically designed for Raspberry Pi boards, including the Raspberry Pi 3B used in this project. It features a 5-megapixel fixed-focus camera capable of capturing still images and videos.

In this project, the Raspberry Pi Camera Rev 1.3 serves as the primary visual input device. It connects directly to the Raspberry Pi's CSI (Camera Serial Interface) port, ensuring a straightforward and convenient integration process.

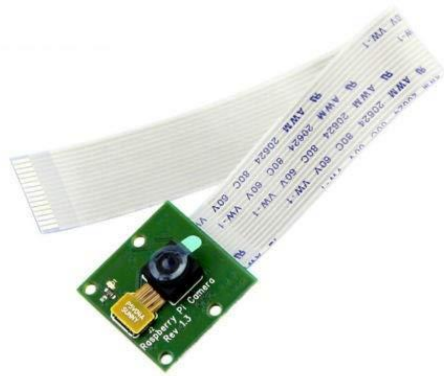
The camera module is responsible for recording a 5-second video of the driver's face, which is essential for detecting drowsiness. It captures high-quality visuals and ensures that the recorded footage contains sufficient details for subsequent analysis.

With its small form factor and lightweight design, the Raspberry Pi Camera Rev 1.3 is ideal for embedding in compact systems such as the driver drowsiness detection system.

The camera module supports various image resolutions and frame rates, providing flexibility to adjust the recording settings according to project requirements.

By connecting to the Raspberry Pi 3B, the camera module takes advantage of the board's processing power and capabilities to handle the captured video data effectively.

Overall, the Raspberry Pi Camera Rev 1.3 plays a crucial role in this project by providing a reliable and easily integrated visual input source, enabling the system to detect and analyze the driver's eye movements and identify signs of drowsiness accurately.



RASPBERRYPI CAMERA

<u>GSR (Galvanic Skin Response) Sensor</u>: The GSR (Galvanic Skin Response) sensor used in this project is a device that measures the electrical conductance of the skin. It detects changes in the skin's conductance due to variations in sweat gland activity, which is closely linked to emotional or arousal states.

In this project, the GSR sensor is integrated with the ESP8266 module. The ESP8266 module, with its built-in Wi-Fi capabilities, acts as a bridge between the GSR sensor and the Raspberry Pi 3B.

The GSR sensor connects to the ESP8266 module to measure the driver's skin conductance levels. The sensor typically consists of two electrodes that make contact with the driver's skin, typically on the fingers or palms.

The GSR sensor sends the conductance measurements to the ESP8266 module, which then communicates this data wirelessly to the Raspberry Pi 3B over Wi-Fi. On the Raspberry Pi, the received GSR data is processed along with other sensor inputs and video analysis to determine the driver's drowsiness levels accurately.

Integrating the GSR sensor with the ESP8266 module allows for real-time monitoring of the driver's physiological response, providing valuable data for the driver drowsiness detection system.

The ESP8266 module's role in this project is to enable seamless wireless communication between the GSR sensor and the Raspberry Pi 3B, ensuring the timely transmission of conductance data for drowsiness analysis.

Together, the GSR sensor and the ESP8266 module enhance the accuracy and effectiveness of the driver drowsiness detection system by capturing physiological responses and facilitating their integration into the overall analysis.

GSR SENSOR

ESP8266 Module: The ESP8266 module is a versatile and widely used Wi-Fi module that plays a crucial role in this project. It serves as a bridge between various components, enabling wireless communication and data transmission.
In this project, the ESP8266 module facilitates the wireless connectivity between the GSR sensor and the Raspberry Pi 3B. It acts as a medium for transmitting the GSR sensor's conductance data to the Raspberry Pi for analysis.
The module integrates seamlessly with the GSR sensor, receiving the conductance measurements from the sensor via its analog or digital interface.
Equipped with built-in Wi-Fi capabilities, the ESP8266 module establishes a wireless connection with the Raspberry Pi, allowing for reliable and real-time data transmission.
Using the Wi-Fi connection, the ESP8266 module communicates with the Raspberry Pi, sending the received GSR data to the main computing platform for further processing and analysis.
The ESP8266 module's compact size and low-power consumption make it suitable for embedded applications, such as integrating it with the GSR sensor in the driver drowsiness detection system.
Its compatibility with various programming languages and development environments makes it flexible and accessible for developers.
The module's firmware can be programmed to establish secure Wi-Fi connections and handle data transmission protocols effectively.
In this project, the ESP8266 module acts as a crucial link, enabling wireless

communication between the GSR sensor and the Raspberry Pi 3B, ensuring the seamless integration of physiological data into the driver drowsiness detection system.

Overall, the ESP8266 module enhances the system's functionality by providing wireless connectivity, enabling efficient data transmission, and facilitating the real-time monitoring and analysis of the driver's physiological response.

ESP8266

VS Code: VS Code, short for Visual Studio Code, is a widely used source code editor developed by Microsoft. It provides a powerful and versatile environment for software development and programming tasks. In this project, VS Code serves as the primary development tool for writing, editing, and managing the Python code used in the driver drowsiness detection system.

With its intuitive user interface and extensive features, VS Code offers a convenient and efficient coding experience. It supports various programming languages, including Python, making it suitable for this project's requirements.

In VS Code, developers can leverage its built-in functionalities such as code highlighting, auto-completion, and debugging tools to streamline the development process. These features contribute to writing high-quality and error-free code.

Additionally, VS Code provides seamless integration with Git, enabling version control and collaboration with other developers. This feature facilitates code management and allows for efficient tracking of project changes.

The extensive marketplace of VS Code offers a wide range of extensions and plugins that can further enhance its capabilities and tailor it to specific project needs.

Throughout the project, developers can utilize VS Code's integrated terminal to execute commands, run scripts, and monitor the system's output, providing a comprehensive development and testing environment.

Overall, VS Code is an essential tool in this project as it provides a robust and user-friendly coding environment. Its features and flexibility contribute to efficient code development, debugging, and project management, enabling the successful implementation of the driver drowsiness detection system.

ARDUINO IDE: The Arduino IDE (Integrated Development Environment) is a software platform used for programming and developing applications for Arduino boards. It provides a user-friendly interface that simplifies the process of writing, compiling, and uploading code to Arduino microcontrollers.

In this project, the Arduino IDE is utilized for programming the ESP8266 module, which acts as a bridge between the GSR sensor and the Raspberry Pi 3B.

The Arduino IDE supports the ESP8266 module, allowing developers to write code in C++ or Arduino programming language to control and communicate with the module.

Using the Arduino IDE, developers can access a wide range of libraries and functions specifically designed for the ESP8266 module, making it easier to implement the required functionalities for wireless communication and data transmission.

The IDE provides a code editor with syntax highlighting, automatic code completion, and error checking, ensuring that the code is written accurately and efficiently.

Once the code is written, the Arduino IDE compiles it into a binary file that can be uploaded to the ESP8266 module via a USB connection.

During the uploading process, the Arduino IDE communicates with the ESP8266 module, programming it with the compiled code and configuring it to perform the desired tasks.

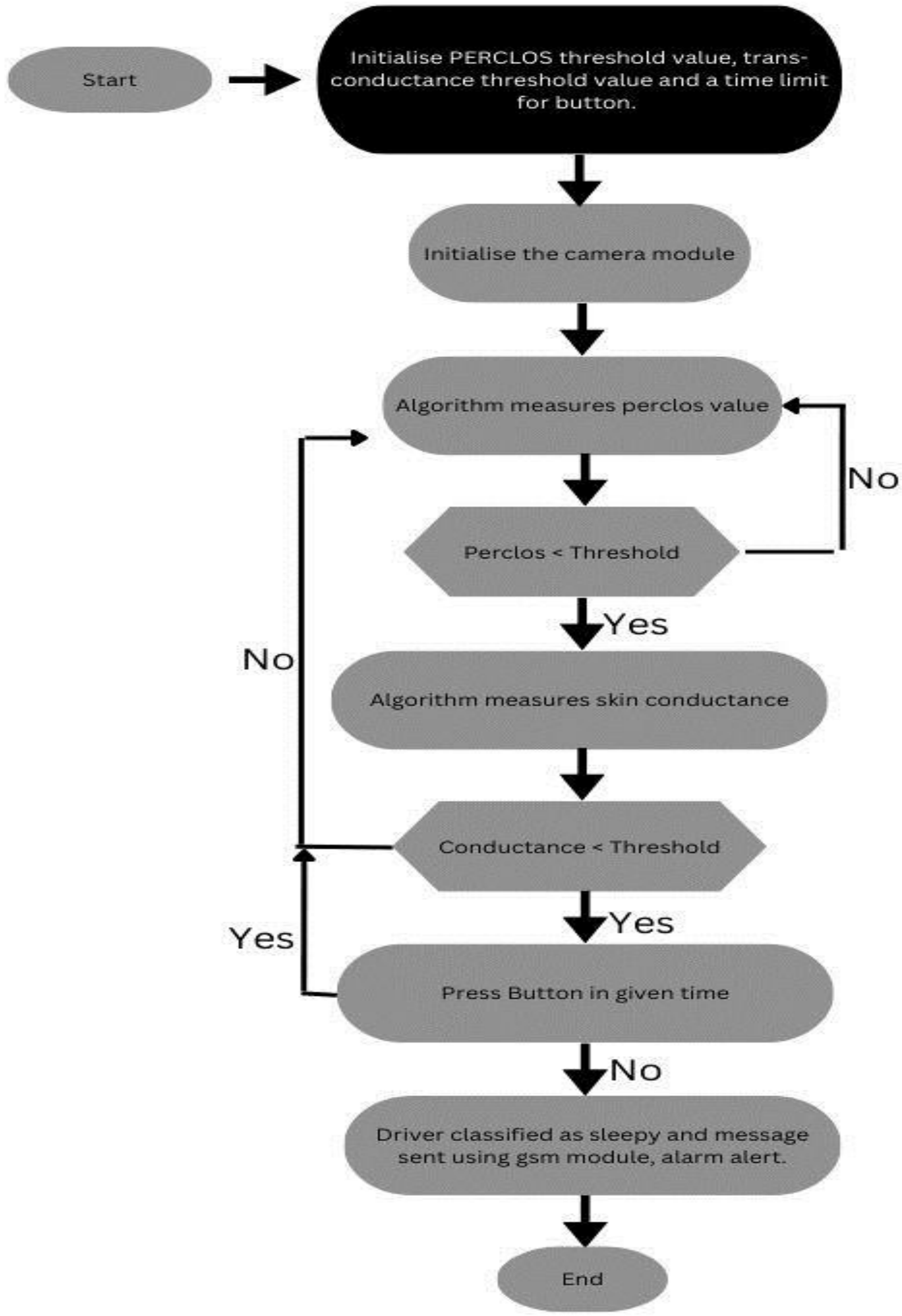
The Arduino IDE also provides a serial monitor, which allows for debugging and monitoring the data exchanged between the ESP8266 module and other components in real-time.

In this project, the Arduino IDE serves as the development environment for programming the ESP8266 module, enabling wireless communication between the GSR sensor and the Raspberry Pi 3B, and facilitating the seamless integration of the physiological data into the driver drowsiness detection system.

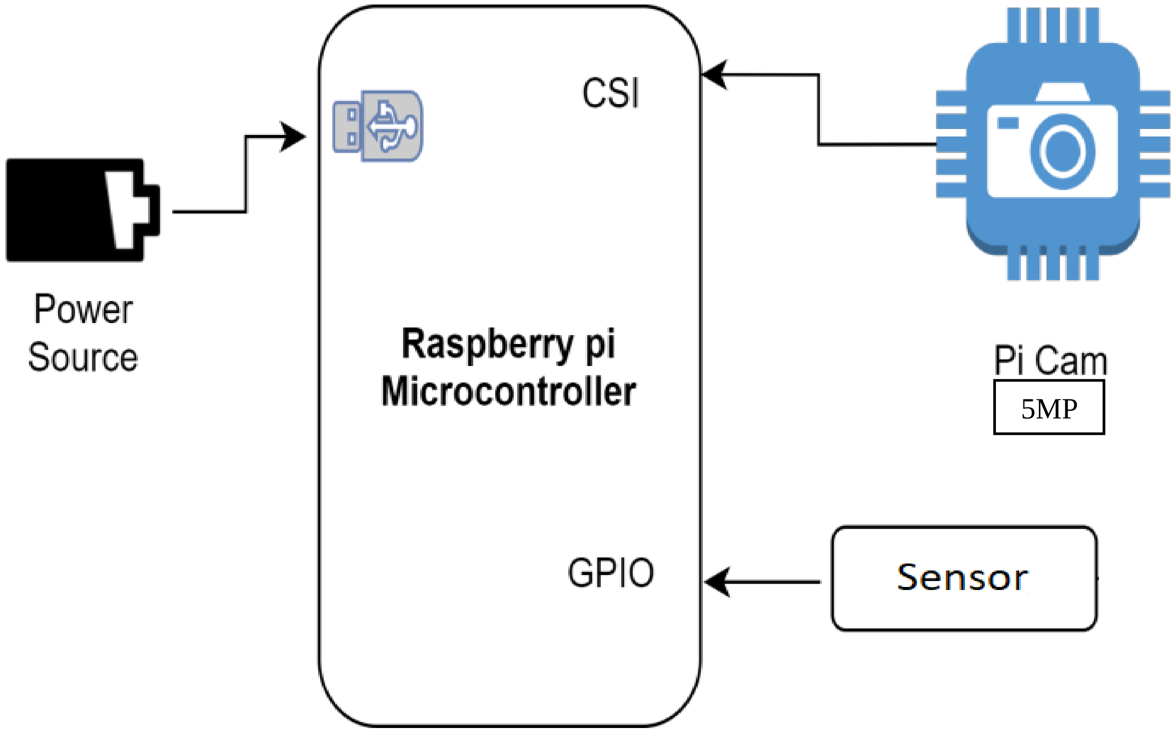
Overall, the Arduino IDE simplifies the programming process, enhances development efficiency, and enables the implementation of complex functionalities in the project by providing a comprehensive set of tools and libraries for the ESP8266 module.

CHAPTER 3 – DESIGN AND DEVELOPMENT

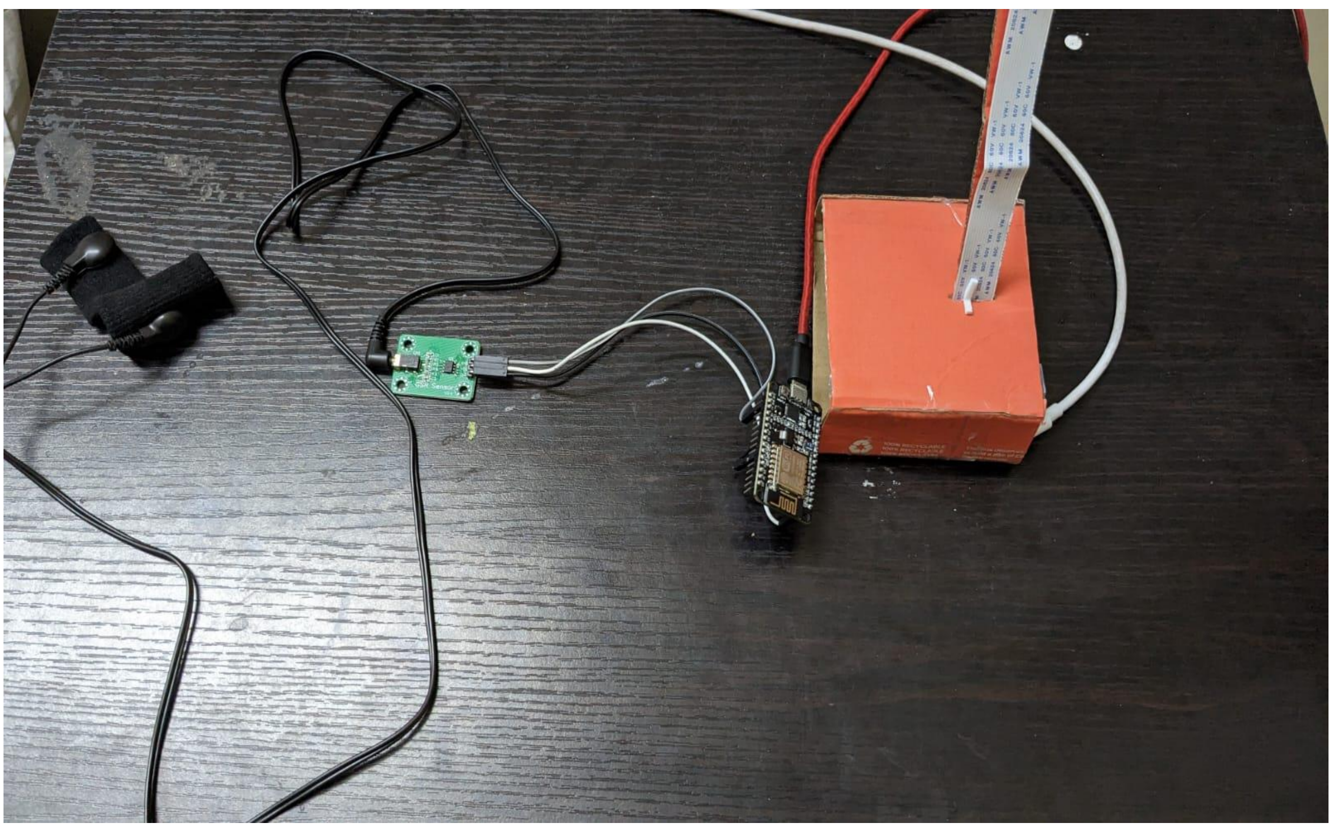
3.1 BLOCK DIAGRAM


Start
Initialise PERCLOS threshold value, trans-conductance threshold value and a time limit for button.
Initialise the camera module
Algorithm measures perclos value
No
Perclos < Threshold
Yes
No
Algorithm measures skin conductance
Conductance < Threshold
Yes
Yes
Press Button in given time
No
Driver classified as sleepy and message sent using gsm module, alarm alert.
End

3.2 CIRCUIT:



3.3 CIRCUIT CONNECTIONS:



CHAPTER 4 – DEMONSTRATION/EVALUATION AND RESULTS

4.1 Demonstration:

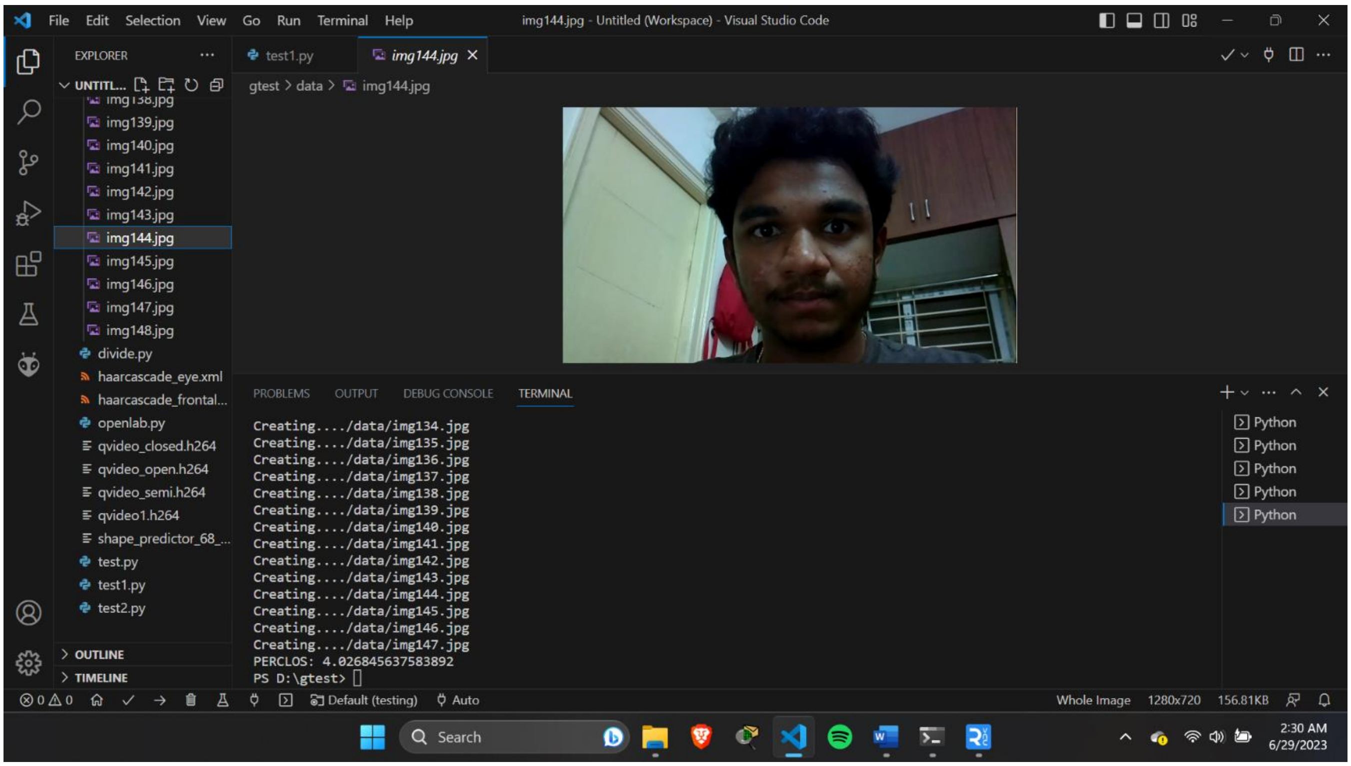
1. **Initialization**: The Raspberry Pi 3B, equipped with the required modules (Raspberry Pi Camera Rev 1.3, GSR sensor, and ESP8266 module), is powered on, and the necessary software components are loaded.
2. **Video Recording**: The Raspberry Pi Camera Rev 1.3 captures a 5-second video of the driver's face.
3. **Video Transfer**: Using the SSH protocol and the pscp command, the recorded video is transferred from the Raspberry Pi 3B to a PC connected via the network.
4. **Image Extraction**: On the PC, a Python code executes and converts the video into a set of images. Each image represents a frame from the video.
5. **Eye State Detection**: The Python code analyzes each image in the folder using Haar cascade classifiers, specifically haarcascade frontal face and haarcascade eye XML files. It detects whether the eyes are open or closed in each image.
6. **Closed Eye Percentage Calculation**: The code keeps count of the number of closed-eye images in the set. It then calculates the percentage of closed-eye images relative to the total number of images.
7. **Drowsiness Check**: If the calculated percentage of closed-eye images surpasses a predefined threshold, the system proceeds to the next stage, indicating a potential drowsiness condition
8. **Conductance Measurement**: Simultaneously, the GSR sensor, connected to the ESP8266 module, measures the driver's skin conductance level. The GSR sensor captures the variations in the driver's sweat gland activity, which indicates changes in stress or arousal levels.
9. **Conductance Threshold Check**: The conductance value obtained from the GSR sensor is compared against a predefined threshold. If the conductance level falls below the threshold, it signifies decreased alertness.
10. **Button Press Challenge**: To further assess the driver's responsiveness, a time-limited challenge is presented. The driver must press a button connected to the Raspberry Pi 3B within the stipulated time.
11. **Drowsiness Classification**: If the driver fails to press the button within the time limit, or if any of the previous stages indicate drowsiness (high closed-eye percentage or low

conductance value), the system classifies the driver as drowsy.

12. **Alarm Trigger**: Upon classification of drowsiness, an alarm is triggered to alert the driver and notify others of the potential hazard.

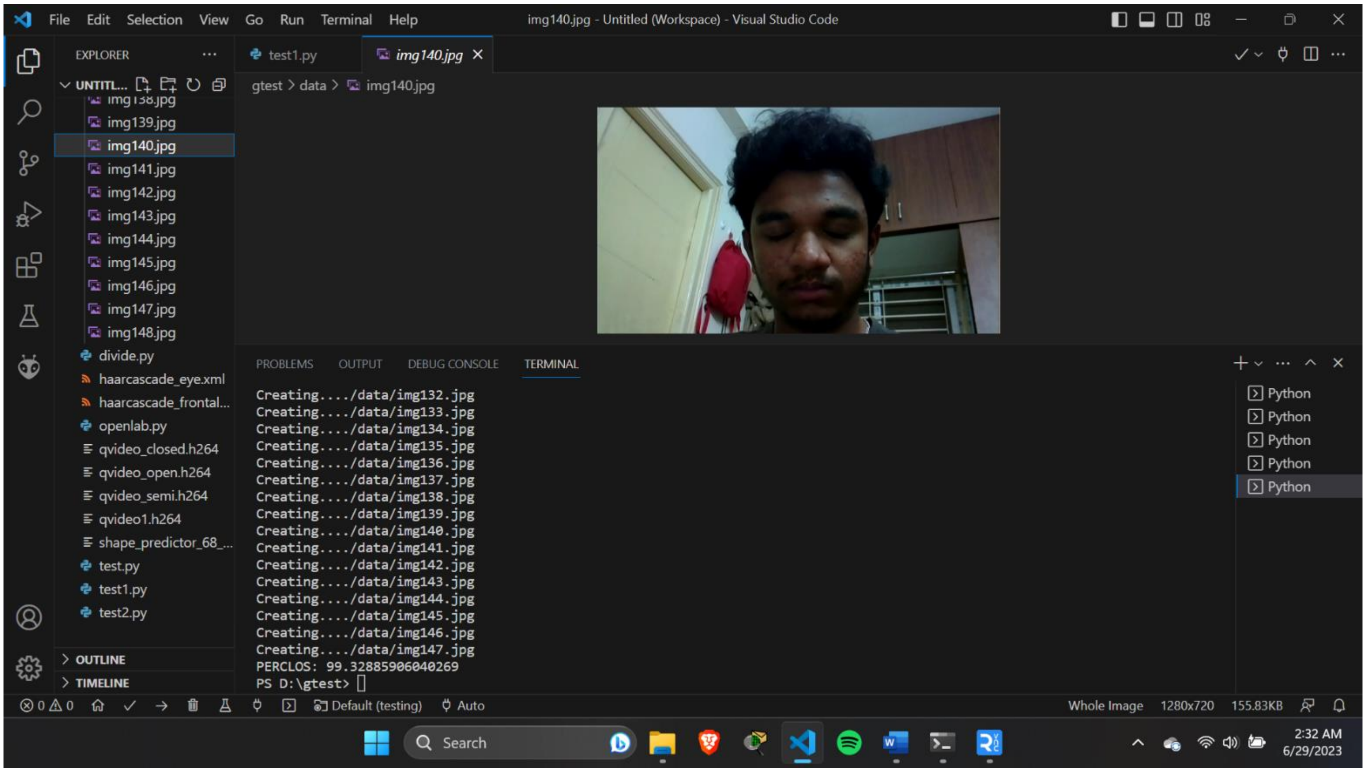
4.2 RESULTS:

1) When only open eyes are present:



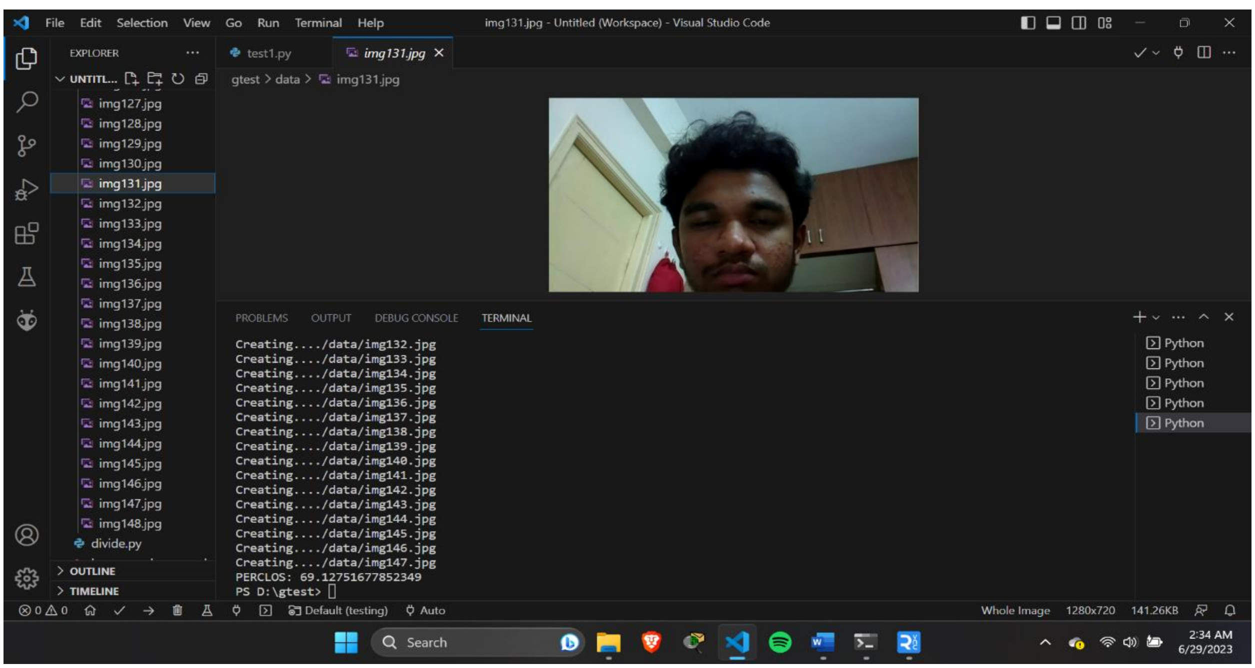
PERCLOS: 4.02%

2) When only closed eyes are present:



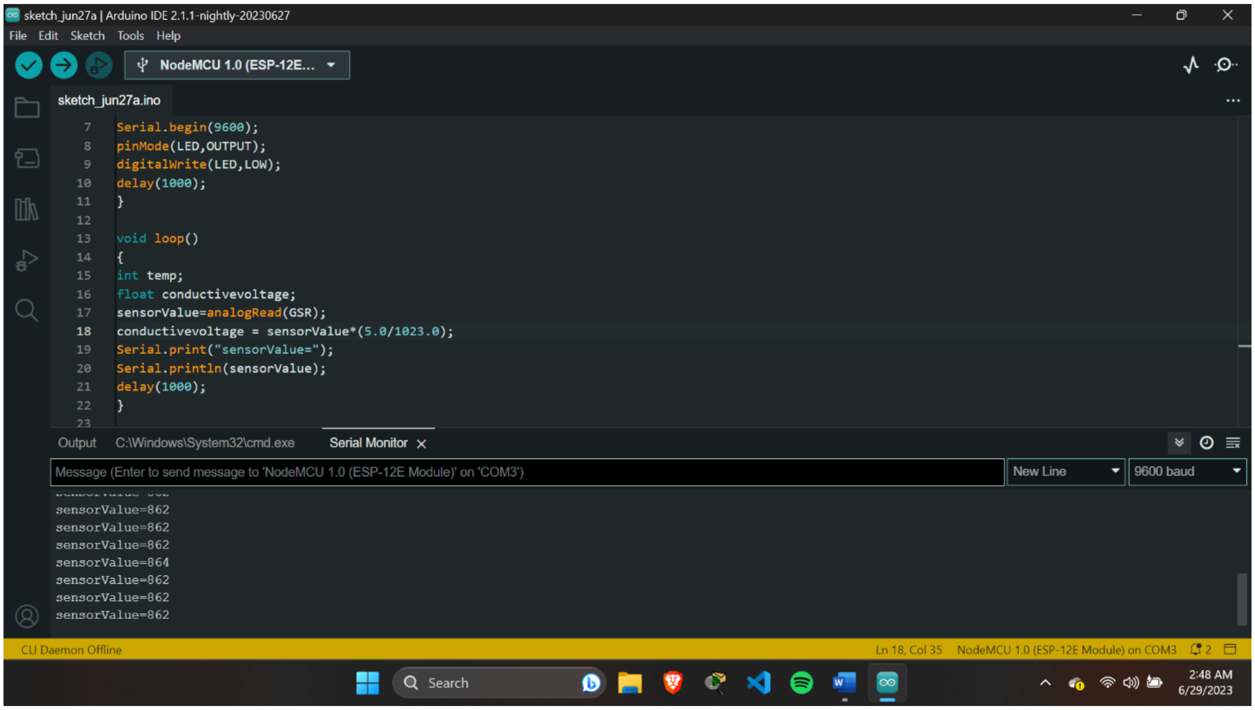
PERCLOS: 99.32%

3) When user is sleepy. Some open and Some closed eye images are present:



PERCLOS: 69.17

GSR Sensor:


sketch_jun27a | Arduino IDE 2.1.1-nightly-20230627
File Edit Sketch Tools Help
NodeMCU 1.0 (ESP-12E...
sketch_jun27a.ino
Serial.begin(9600);
pinMode(LED,OUTPUT);
digitalWrite(LED,LOW);
delay(1000);
}
void loop()
{
int temp;
float conductivevoltage;
sensorValue=analogRead(GSR);
conductivevoltage = sensorValue*(5.0/1023.0);
Serial.print("sensorValue=");
Serial.println(sensorValue);
delay(1000);
}
Output
C:\Windows\System32\cmd.exe
Serial Monitor
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM3')
New Line
9600 baud
sensorValue=862
sensorValue=862
sensorValue=862
sensorValue=864
sensorValue=862
sensorValue=862
sensorValue=862
CLI Daemon Offline
Ln 18, Col 35
NodeMCU 1.0 (ESP-12E Module) on COM3
Search
2:48 AM
6/29/2023

CHAPTER 5 – CONCLUSION AND FUTURE WORK

Conclusion:

In conclusion, the driver drowsiness detection system, utilizing the Raspberry Pi 3B, Raspberry Pi Camera Rev 1.3, and ESP8266 module, demonstrates a functional solution for monitoring and detecting driver drowsiness. The system effectively captures and analyzes eye movements through video processing, measures conductance levels using the GSR sensor, and prompts the driver for a button press challenge. These components work in tandem to assess the driver's alertness and trigger an alarm if drowsiness is detected.

Future Work:

While the current implementation showcases promising results, there are a few identified gaps that can be addressed in future work. One significant gap is the inability to interface the GSR sensor directly with the Raspberry Pi 3B. To overcome this limitation, future work could focus on finding alternative methods or modules that enable seamless integration between the GSR sensor and the Raspberry Pi. Exploring additional communication protocols or hardware interfaces may facilitate direct data acquisition from the GSR sensor, allowing for more robust analysis and assessment of driver drowsiness levels.

Additionally, we could involve refining and optimizing the detection algorithms and thresholds used in the system. This would improve the accuracy of eye state detection and conductance level analysis, resulting in more reliable drowsiness classification.

Furthermore, considering the dynamic nature of driver drowsiness, future work could explore the incorporation of additional physiological and behavioral measures to enhance the system's effectiveness. These measures could include heart rate monitoring, head movement tracking, or steering wheel grip analysis, among others.

By addressing these gaps and considering future improvements, the driver drowsiness detection system can evolve into a more robust and accurate solution for ensuring driver safety and reducing the risks associated with drowsy driving.

REFERENCES

1) Bajaj, Jaspreet & Kumar, Naveen & Kaushal, Rajesh Kumar & H L, Gururaj & Flammini, Francesco & Natarajan, Rajesh. (2023). System and Method for Driver Drowsiness Detection Using Behavioral and Sensor-Based Physiological Measures. Sensors. 23. 1292. 10.3390/s23031292.
2) Dasgupta, Anirban & Rahman, Daleef & Routray, Aurobinda. (2018). A Smartphone-Based Drowsiness Detection and Warning System for Automotive Drivers. IEEE Transactions on Intelligent Transportation Systems. PP. 1-10. 10.1109/TITS.2018.2879609.
3) Anh-Cang Phan, Thanh-Ngoan Trieu, Thuong-Cang Phan,Driver drowsiness detection and smart alerting using deep learning and IoT,Internet of Things,Volume 22,2023,100705,ISSN 2542-6605.
4) M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas and A. Mahmood, "A Survey on State-of-the-Art Drowsiness Detection Techniques," in IEEE Access, vol. 7, pp. 61904-61919, 2019, doi: 10.1109/ACCESS.2019.2914373.
5) Arcaya, Alexis & Pegatoquet, Alain & Castagnetti, Andréa & Raybaut, Julien & Coz, Pierre. (2020). Deep Learning for Eye Blink DetectioN. IEEE Embedded Systems Letters. PP. 10.1109/LES.2020.3029313.