DRIVER SLEEP DETECTION AND ALARMING SYSTEM

By

ASHWIN M (71382104008)
YOGIRAJ G (71382104049)
SRI HARI S (71382104303)

Report submitted in partial fulfilment of
the requirements for the Degree of
Bachelor of Engineering in Electronics
And Communication Engineering

Sri Ramakrishna Institute of Technology
Coimbatore – 641010
May,2024

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who actuated it, without them it would never have into existence. To them we lay the word of gratitude imprinted within us.

We would like to express our gratitude to the SNR management Trust and the Principal **Dr. M. Paulraj**, for giving us access to all the necessary technological resources.

We express our gratitude to **Dr. S. Anila**, Professor &Head, Department of Electronics and Communication Engineering for her conscientious guidance and encouragement.

We would like to extend our sincere gratitude to our Project Supervisor, **Ms.A.Poornima**, Assistant Professor, Department of Electronics and Communication Engineering for his cordial support and guidance.

We would like to extend our sincere gratitude to our Design Project coordinator, **Ms.A.K.Kavitha**, Assistant professor(Sr,Gr.), Department of Electronics and Communication Engineering for her cordial support and guidance.

We sincerely thank all the teaching and non-teaching staff of Electronics and Communication Engineering department for all what they have been for their enquiry about the project.

BONAFIDE CERTIFICATE

Certified that this Design Project report titled "**DRIVER SLEEP DETECTION AND ALARMING SYSTEM "** is the bonafide work of **ASHWIN M (71382104008) , YOGIRAJ G (71382104049) ,SRI HARI S (71382104303)** who carried out the project work under my supervision.

SIGNATURE

Dr. S. ANILA

Professor and Head

Department of Electronics and
Communication Engineering,
Sri Ramakrishna Institute of
Technology, Coimbatore – 10.

SIGNATURE

Ms. A. POORNIMA

Assistant Professor
Project Supervisor

Department of Electronics and
Communication Engineering,
Sri Ramakrishna Institute of
Technology, Coimbatore – 10

Submitted for the viva-voce examination held on ________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

APPROVAL AND DECLARATION

This project report titled "DRIVER SLEEP DETECTION AND ALARMING SYSTEM" was prepared and submitted by Ashwin M (71382104008),Yogiraj G (71382104049) and Sri Hari S (71382104303) has been found satisfactory in terms of scope, quality and presentation as partial fulfilment of the requirement for the Bachelor of Engineering (Electronics and Communication Engineering) in Sri Ramakrishna Institute of Technology,Coimbatore(SRIT).

Checked and Approved by

Ms. A. Poornima
Assistant Supervisor
Project Supervisor

Department of Electronics and Communication Engineering
Sri Ramakrishna Institute of Technology,
Coimbatore-10
May, 2024

DRIVER SLEEP DETECTION AND ALARMING SYSTEM

ABSTRACT

The objective of the project to detect the sleepiness of driver while driving in order to reduce the risks associated with drowsy driving. This project provides the design and implementation of a system for detecting driver fatigue and alerting the driver to avoid potential accidents. The system includes camera, raspberry pi that monitor the driver's behavior in real time. Machine learning algorithms are used to analyze the collected data and identify patterns indicating drowsiness or fatigue. When the system detects drowsiness, it sounds an alarm to alert the driver and prompt them to take the necessary actions, such as taking a break or stopping. The system's effectiveness is assessed through simulations and real-world testing scenarios, demonstrating its potential to improve road safety by addressing the dangers associated with driver drowsiness

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Driver fatigue and drowsiness are significant contributors to road accidents worldwide, posing a serious threat to public safety. According to statistics from road safety authorities, a substantial portion of accidents can be attributed to drivers who are either fatigued or drowsy. Recognizing the critical need to address this issue, researchers and automotive engineers have developed advanced driver sleep detection and alarming systems. These systems aim to detect signs of driver fatigue in real-time and alert the driver to prevent potential accidents. The introduction of such systems marks a significant advancement in vehicle safety technology, leveraging a combination of sensors, algorithms, and human-machine interaction principles. By continuously monitoring the driver's behavior, physiological indicators, and vehicle dynamics, these systems can accurately identify patterns associated with drowsiness or fatigue. Once detected, timely alerts are issued to the driver through various means, such as auditory alarms, visual cues, or haptic feedback, prompting them to take corrective action.

The issue of driver fatigue is a pressing one, given the increasing number of vehicles on the road and the growing demands placed on drivers. Long hours behind the wheel, irregular sleep patterns, and monotonous driving conditions all contribute to the prevalence of drowsy driving. Studies have shown that fatigued drivers exhibit impaired reaction times, reduced vigilance, and compromised decision-making abilities, all of which significantly increase the likelihood of an accident. Addressing driver fatigue is therefore crucial in enhancing road safety and reducing the incidence of accidents. The development of driver sleep detection and alarming systems represents a proactive approach to this problem, utilizing cutting-edge technology to mitigate the risks associated with drowsy driving.

One of the key components of these systems is the use of various sensors to monitor the driver’s state. Cameras can track eye movements and facial expressions, detecting signs of drowsiness such as frequent blinking, yawning, or head nodding. Other sensors can monitor physiological indicators such as heart rate and skin conductivity, which can also provide clues about the driver’s level of alertness. Additionally, vehicle dynamics sensors can detect erratic driving patterns, such as lane drifting or sudden steering corrections, which may indicate a loss of attention. By integrating data from these different sources, advanced algorithms can analyze the information in real-time and accurately determine whether the driver is fatigued.

When signs of drowsiness are detected, the system activates an alert to prompt the driver to take corrective action. The alerts can vary in form, including auditory alarms, visual cues on the dashboard, or haptic feedback through the steering wheel or seat. The goal is to capture the driver’s attention and encourage them to take a break, switch drivers, or employ other measures to restore their alertness. Some systems are also integrated with vehicle control functions, such as adaptive cruise control or lane-keeping assist, which can provide additional support in maintaining safe driving conditions.

1.1 Applications

- Commercial Vehicles: Implementation of driver sleep detection and alarming systems in long-haul trucks, buses, and delivery vehicles to mitigate the risks of drowsy driving among professional drivers. Long hours on the road can lead to significant fatigue, and these systems can alert drivers before they reach a critical point of tiredness. The systems can be especially beneficial for trucking companies, where drivers often face tight schedules and pressure to drive extended distances without sufficient rest.
- Personal Vehicles: Integration of these systems in passenger cars and recreational vehicles to enhance safety for individual drivers and their passengers during long journeys. As the technology becomes more affordable, it is expected that more personal vehicles will come equipped with driver sleep detection systems, providing an added layer of safety for families and individuals on road trips or daily commutes.
- Public Transportation: Deployment of driver sleep detection systems in taxis, ride-sharing vehicles, and public transit buses to ensure the safety of passengers and other road users. In urban environments, where drivers often face heavy traffic and long hours, these systems can play a crucial role in preventing accidents caused by driver fatigue.
- Fleet Management: Utilization of these systems by fleet operators to monitor driver behavior, reduce accident rates, and improve overall operational efficiency. By tracking fatigue levels, fleet

managers can better schedule breaks and shift changes, ensuring that drivers remain alert and reducing the likelihood of accidents. Additionally, these systems can provide valuable data for training and improving driver safety protocols.

- Transportation Services: Adoption of driver sleep detection technology by transportation companies offering shuttle services, airport transfers, and other passenger transportation options. These companies can market their commitment to safety, attracting customers who prioritize secure and reliable transportation. The technology can also help in maintaining schedules and reducing liability in the event of an accident.
- Emergency Response Vehicles: Installation of these systems in ambulances, fire trucks, and police cars to prevent accidents and ensure the safety of emergency responders while en route to incidents. Given the high-stress nature of their work and the critical need for timely responses, ensuring that drivers are fully alert is paramount. These systems can provide alerts during prolonged shifts or when responders are called out in the middle of the night.
- Automotive Industry: Integration of driver sleep detection features into new vehicle models as part of advanced driver assistance systems (ADAS) to offer enhanced safety features for consumers. Automakers are increasingly focusing on safety innovations, and incorporating sleep detection technology can differentiate their products in the competitive market. It can also prepare manufacturers for future regulatory requirements that may mandate such features.
- Aviation Industry: Although primarily designed for road vehicles, driver sleep detection technology can be adapted for use in aviation. Pilots, especially those on long-haul flights, face similar challenges with fatigue. Implementing similar systems in cockpits can enhance flight safety by ensuring that pilots remain alert during critical phases of flight.
- Maritime Industry: Ships and other maritime vessels can benefit from sleep detection technology to monitor the alertness of captains and crew members, especially during long voyages. Fatigue at sea can lead to severe accidents, and timely alerts can help prevent navigation errors and collisions.
- Construction and Heavy Machinery: Operators of heavy machinery and construction vehicles can also experience fatigue, leading to dangerous working conditions. Implementing sleep detection systems in these vehicles can improve site safety and reduce the risk of accidents caused by drowsiness.
  - Railway Systems: Train operators can benefit from sleep detection technology to prevent accidents on long routes. The systems can provide alerts to operators and control centers, allowing for timely interventions if an operator is detected to be fatigued.
  - Military Vehicles: Military operations often involve extended periods of vigilance and operations under stressful conditions. Implementing sleep detection systems in

military vehicles can help ensure that drivers and operators remain alert, reducing the risk of accidents in critical situations.

- Mining Operations: Vehicles used in mining operations often operate around the clock in harsh conditions. Sleep detection systems can enhance safety by monitoring the alertness of operators who manage heavy equipment in these challenging environments.

Personal Vehicles**:**

- Safety Enhancement**:** Reduces the risk of accidents due to driver fatigue by alerting the driver when signs of drowsiness are detected.
- Insurance Benefits: Some insurance companies might offer discounts for vehicles equipped with advanced safety features like driver sleeping detection systems.

Commercial Fleet Management**:**

- Driver Safety: Protects drivers of commercial trucks and buses by monitoring their alertness and preventing accidents caused by drowsiness.
- Operational Efficiency: Helps fleet managers ensure drivers take necessary breaks, thereby maintaining compliance with regulations and improving overall efficiency.
- Reduced Liability: Lowers the risk of accidents and associated liabilities for transport companies.

Taxi and Ride-Sharing Services**:**

- Customer Safety: Provides an added layer of safety for passengers by ensuring the driver remains alert during the ride.
- Driver Welfare: Encourages drivers to take breaks and avoid driving while drowsy, promoting better health and safety practices.

CHAPTER 2

LITERATURE SURVEY

1.Title: Driver Drowsiness Detection

Authors: K. Satish , A. Lalitesh ,K. Bhargavi, M. Sishir Prem ,T Anjali.

Publication: 2020 International Conference on Communication and Signal Processing (ICCSP)

Abstract: All over the world Drowsiness has been the significant cause of horrible accidents which is causing deaths and fatalities injuries. Day by Day fatal injuries numbers are increasing globally. From the past many years, researchers have concluded drivers with a lack of sleep and more tiredness which causes drowsiness of the driver. this paper shows a new experimental model is designed for detecting drowsiness of driver is presented to reduce accidents caused by this problem which increases transport safety. In this work, two ways are used to detect the drowsiness of a person effectively. First Driver face is captured and eye retina detection and facial feature extraction are done and blinking values are calculated then threshold values are set. Secondly, the Arduino module is used which is integrated with elastomeric sensors for real-time calculation of driver hand pressure on the car steering wheel and the threshold value is set. The result from both methods is taken as input for taking the final decision and alerting the driver.

2.Title: Driver drowsiness detection system

Authors: Belal Alshaqaqi ,Abdullah Salem Baquhaizel ,Mohamed El Amine Ouis ,Meriem Boumehed ,Abdelaziz Ouamri

Publication: : 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)

Abstract : Drowsiness and Fatigue of drivers are amongst the significant causes of road accidents. Every year, they increase the amounts of deaths and fatalities injuries globally. In this paper, a module for Advanced Driver Assistance System (ADAS) is presented to reduce the number of accidents due to drivers fatigue and hence increase the transportation safety; this system deals with automatic driver drowsiness detection based on visual information and Artificial Intelligence. We propose an algorithm to locate, track, and analyze both the drivers face and eyes to measure PERCLOS, a scientifically supported measure of drowsiness associated with slow eye closure.

3.Title: Intelligent driver drowsiness detection through fusion of yawning and eye closure

Authors: M. Omidyeganeh ,A. Javadtalab ,S. Shirmohammadi

Publication: 2011 IEEE International Conference on Virtual Environments, Human Computer Interfaces and Measurement Systems Proceedings

Abstract: Driver drowsiness is a major factor in most driving accidents. In this paper we present a robust and intelligent scheme for driver drowsiness detection employing the fusion of eye closure and yawning detection methods. In this approach, the driver's facial appearance is captured via a camera installed in the car. In the first step, the face region is detected and tracked in the captured video sequence utilizing computer vision techniques. Next, the eye and mouth areas are extracted from the face; and they are studied to find signs of driver fatigue. Finally, in a fusion phase the driver state is determined and a warning message is sent to the driver if the drowsiness is detected. Our experiments prove the high efficiency of the proposed idea.

4.Title: Driver Drowsiness Detection and Warning using Facial Features and Hand Gestures

Authors: Arpit S Agarkar , R Gandhiraj ,Manoj Kumar Panda

Publication: 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)

Abstract: According to National Highway Traffic Safety Administration (NHTSA), drowsy driving is one of the primary causes of accidents. Numerous valuable lives can be saved, accidents can be reduced or avoided, and the cost of injury and damage to infrastructure may be reduced with a timely alert or warning to the negligent driver. Advanced Driver Assistance Systems (ADAS) consists of the active safety system which includes the detection of the driver's face to determine their level of drowsiness. This paper provides a camera-based technique which relies on fiducial components, such as lips, eye movement, and hand gestures of the driver which are often natural responses of a human to yawning. A front camera installed on the windscreen is used to continually monitor the driver and Raspberry Pi is utilized for processing the images. The proposed warning system gives an audio warning when the driver is yawning or going into the state of drowsiness. Results illustrate that the proposed technique is effective at detecting signs of driver's drowsiness and yawning. It can differentiate between when the driver's hand is placed over the mouth to infer it as yawning and when it is touching other parts of the face to infer it as not yawning.

5.Title: Driver Drowsiness Detection based on Monitoring of Eye Blink Rate

Authors: P Baby Shamini ,M. Vinodhini ,B Keerthana ,S Lakshna ,K. R Meenatchi

Publication: 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT) .

Abstract : A computer vision-based technology has been implemented for the making of Driver Drowsiness Systems. The small camera called webcam is used as the framework that mainly concentrates towards the activity of the driver and continuously monitors the driver by checking whether the driver is drowsy or not. A notice sign as an alarm is issued to the driver in such circumstances when certain precautions is perceived. The data framework oversees using the information which is directly picked up for the video to find the landmarks which the ratio where the eyes of a certain limited individuals may exist. If driver has his or her eyes closed for a period of time the proposed system declares that the driver is falling asleep and a warning has been given to warn the driver. This project also works when the driver is yawning and works in lightning conditions too.

6.Title: Driver Drowsiness Detection Using Machine Learning Algorithm

Authors: N Prasath ,J Sreemathy , P Vigneshwaran

Publication: 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)

Abstract : It is just a short time until self-driving vehicles become omnipresent; be that as it may, human driving management will stay a need for quite a long time. Driver Drowsiness is one of the significant reasons of roadways accidents these days. Hence fatigue and drowsiness detection play a major role in preventing the road accidents. Every year, because of this there is an increase in the number of deaths and injuries globally. Recently, in this decade, many images processing-based approaches were created and used to detect driver's drowsiness status. To minimize the number of accidents, a method is proposed in this paper. The algorithm focuses on the eye closure and yawning ratios. The driver is alarmed, if he/she is feeling sleepy.

7.Title: Driver Drowsiness Detection by Employing CNN and Dlib

Authors**:** Nawazish Ali ,Imran Hasan ,Tansel Özyer ,Reda Alhajj

Publication: 2021 22nd International Arab Conference on Information Technology (ACT)

Abstract : Every year thousands of people lose their life due to road accidents. One of the main reasons for these accidents is driver drowsiness. In driver drowsiness, the driver slept while driving, which causes the road accident, especially on the long routes. Driver fatigue and micro sleep while driving caused the fatal accident and death of human beings. To overcome this problem, we are implementing a technique in which we capture the image of the driver. After capturing the image of the driver, we process driver images to detect driver drowsiness. For the processing of the driver image, we are using two different techniques with each other. In the first technique, we are using the Dlib for image drowsiness detection by detecting that driver's eyes are closed and the driver is yawning. In the second technique, we used CNN for the detection of yawning and the eyes of the driver are closed or not and predict driver drowsiness. After implementing the two techniques we combine the output of both techniques. After combining both techniques we test the system, and it gives us very good results

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Problem Statement

Drowsy driving poses a significant threat to road safety, leading to a substantial number of accidents, injuries, and fatalities worldwide. Despite awareness campaigns and legal measures, the problem persists, particularly in long-haul transportation and commercial vehicle operations. To address this pressing issue, there is a critical need for the development and implementation of an effective Driver Sleep Detection and Alarming System. The primary objective of this project is to design, develop, and validate a comprehensive driver sleep detection and alarming system capable of reliably detecting signs of driver fatigue and alerting the driver in real-time. The system should integrate advanced sensor technologies, data processing algorithms, and alert mechanisms to accurately monitor the driver's alertness level and mitigate the risks associated with drowsy driving

3.2 Methodology

Creating a driver sleeping detection and alarming system using Raspberry Pi involves several steps and components. Here's a proposed methodology:

- Hardware Setup:

  - Raspberry Pi: Choose a Raspberry Pi board (like Raspberry Pi 5) as the main controller.
  - Camera Module: Connect a camera module to the Raspberry Pi for capturing images or video
  - Display and Alarm: Connect a display and an alarm (such as a buzzer or LED) for real-time alerts.

- Driver Monitoring Algorithm:

  - Use image processing techniques to detect the driver's face and eyes.
  - Implement a machine learning model (like OpenCV) to detect eye closure or drowsiness.

- Data Collection and Training:

  - Collect a dataset of images or video clips containing both awake and drowsy states of the driver.
  - Label the dataset accordingly (awake, drowsy).
  - Train the machine learning model using the labeled dataset to recognize signs of drowsiness.

- Real-time Monitoring:
  - Continuously capture images or video frames using the camera module.
  - Process the captured data in real-time using the trained machine learning model to detect drowsiness.

- Alert System:
  - When drowsiness or erratic behavior is detected, trigger the alarm system.
  - Display visual alerts on the connected display.
  - Sound audible alarms through the buzzer or LED.

- Testing and Optimization:
  - Test the system extensively under various conditions (different lighting, driver positions, etc.).
  - Fine-tune the algorithms and thresholds for detection to minimize false positives and false negatives.
  - Optimize the system for performance and efficiency.

- Integration and Deployment:
  - Integrate all components together into a cohesive system.
  - Ensure proper power supply and physical mounting within the vehicle.
  - Test the system in a real-world driving scenario to verify its effectiveness and reliability.

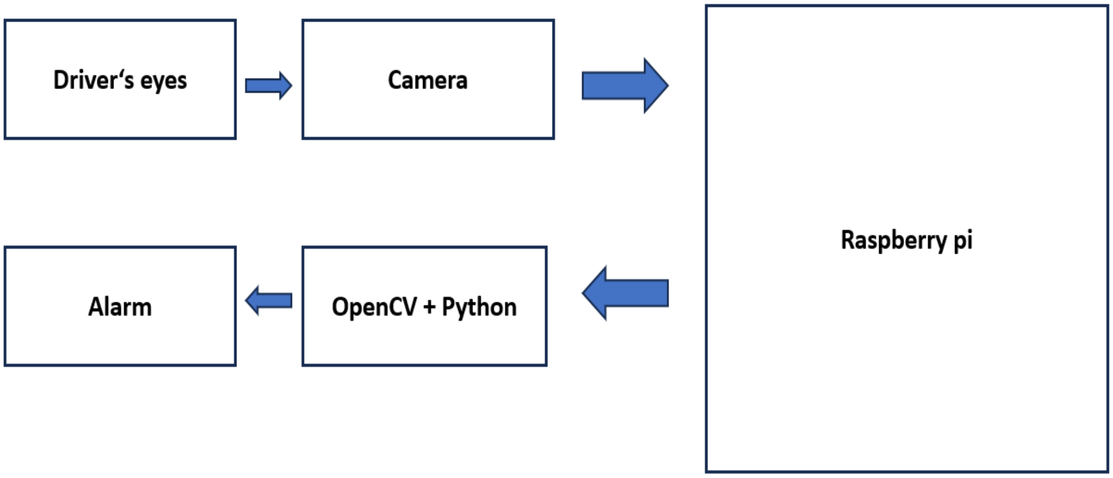


Figure 3.1: Block diagram of the proposed system

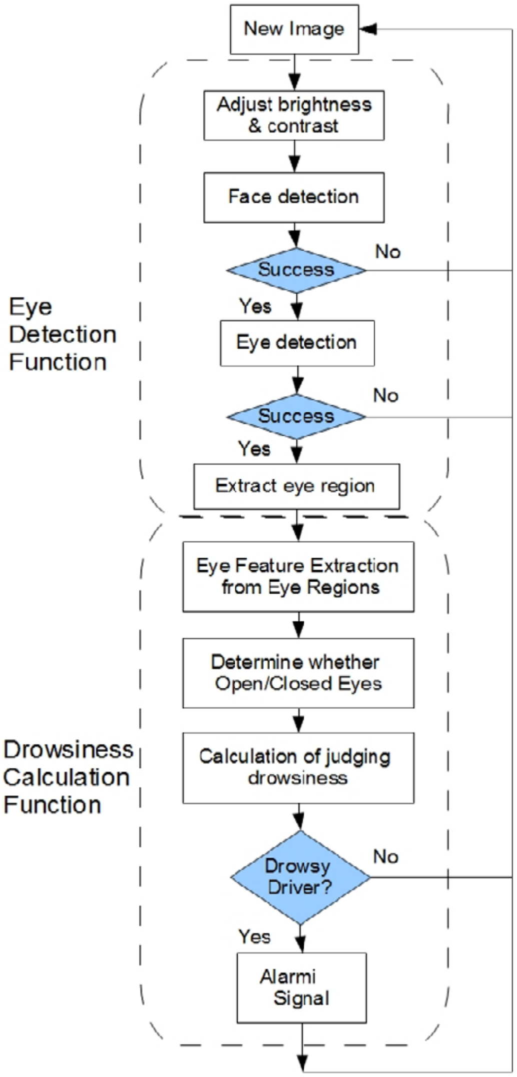

New Image
Adjust brightness & contrast
Face detection
Success
No
Yes
Eye detection
Success
No
Yes
Extract eye region
Eye Detection Function
Eye Feature Extraction from Eye Regions
Determine whether Open/Closed Eyes
Calculation of judging drowsiness
Drowsy Driver?
No
Yes
Alarmi Signal
Drowsiness Calculation Function

Figure 3.2: Flow Chart

3.3 Scope of the Project

- Enhanced Safety: The primary advantage of implementing a driver sleep detection and alarming system is the significant enhancement of road safety. By detecting signs of drowsiness in real-time and alerting the driver promptly, the system helps prevent accidents caused by driver fatigue, reducing the risk of injuries and fatalities on the road.

- Accident Prevention: By alerting drivers when they show signs of drowsiness, the system helps prevent potential accidents before they occur. This proactive approach reduces the likelihood of collisions, especially during long-distance journeys or nighttime driving when fatigue is more prevalent.

- Improved Driver Health and Well-being: Continuous monitoring of driver alertness promotes better driver health and well-being by encouraging regular breaks and rest periods. By preventing drivers from becoming excessively fatigued, the system helps reduce the risk of fatigue-related health issues and improves overall driver fitness.

- Cost Savings: Implementing a driver sleep detection system can lead to significant cost savings for fleet operators and insurance providers. By reducing the number of accidents caused by drowsy driving, organizations can lower expenses associated with vehicle repairs, medical bills, and insurance premiums.

- Regulatory Compliance: In many regions, there are regulations mandating the use of safety systems, including driver sleep detection technology, in commercial vehicles and transportation services. Implementing such systems ensures compliance with these regulations, avoiding potential penalties and legal consequences.

- Enhanced Reputation: Organizations that prioritize safety by implementing driver sleep detection systems demonstrate their commitment to road safety and customer well-being. This can enhance their reputation among customers, employees, and stakeholders, leading to increased trust and loyalty.

- Increased Productivity: By reducing the likelihood of accidents and downtime due to driver fatigue, organizations can improve overall productivity. Vehicles can operate more efficiently, and drivers can maintain higher levels of alertness, leading to smoother operations and faster delivery times.

- Technological Innovation: Implementing driver sleep detection technology showcases an organization's commitment to technological innovation and advancement. It positions them as leaders in the field of vehicle safety and reinforces their competitiveness in the market.

- Customization and Adaptability: Driver sleep detection systems can be customized and adapted to suit the specific needs and preferences of different organizations and vehicle types. This flexibility allows for the integration of additional features and functionalities tailored to unique operational requirements.

3.4 Hardware Description

3.4.1 Raspberry pi 5

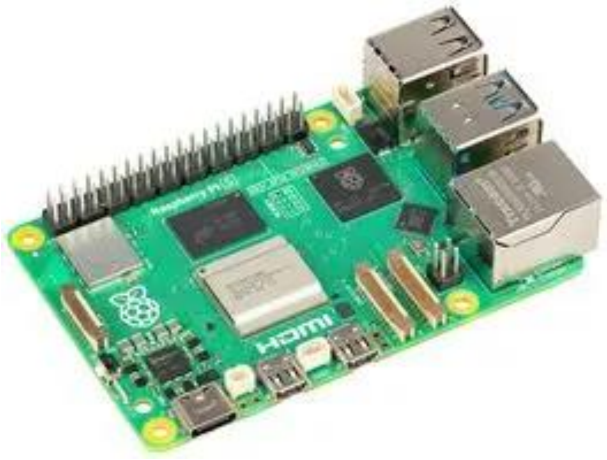


Figure 3.3 : Raspberry pi 5

The Raspberry Pi 5 is the latest iteration of the popular single-board computer series, known for its enhanced performance and versatility. Released with significant upgrades, it features a quad-core ARM Cortex-A76 processor running at 2.4 GHz, providing a substantial boost in processing power compared to its predecessors. This model also includes up to 8GB of LPDDR4X RAM, enabling better multitasking and smoother performance for more demanding applications. Connectivity options have been improved with dual 4K HDMI outputs, USB 3.0 ports, and Gigabit Ethernet, allowing for faster data transfer and more robust multimedia capabilities. Additionally, the Raspberry Pi 5 supports PCIe for expansion, offering new possibilities for developers and hobbyists to integrate high-speed peripherals. Enhanced power management and cooling solutions ensure stable operation, making the Raspberry Pi 5 a powerful, versatile tool for education, DIY projects, and even some professional applications **.**

It is powered by a Broadcom BCM2712 quad-core Arm Cortex-A76 CPU running at 2.4GHz, offering about 2-3 times the performance of the Raspberry Pi 4. It also includes a VideoCore VII GPU which supports OpenGL ES 3.1 and Vulkan 1.2, enhancing graphics performance significantly .The Pi 5 comes in 4GB and 8GB LPDDR4X RAM variants, omitting the lower 2GB option found in earlier models to better meet the demands of more intensive applications. The board features dual-band 802.11ac Wi-Fi, Bluetooth 5.0, Gigabit Ethernet with PoE+ support, and a PCIe 2.0 x1 interface for fast peripherals. It also includes 2 USB 3.0 and 2 USB 2.0 ports, dual 4Kp60 HDMI outputs, and two 4-lane MIPI camera/display interfaces.The Raspberry Pi 5 requires a 5V/5A DC power supply via USB-C and includes new features such as a real-time clock (RTC) powered by an external battery and a power button. Active cooling is recommended for optimal performance under heavy loads.While maintaining a similar form factor to previous models, the Pi 5 introduces new mounting holes for a heatsink and revised FPC connectors for cameras and displays. The Ethernet port has been repositioned to improve board layout. The launch also includes new accessories such as an updated case with an integrated fan for enhanced cooling and an optional active cooler for users requiring robust thermal management

3.4.2 GPIO and the 40-pin header

A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. A 40-pin GPIO header is found on all current Raspberry Pi boards, although it is unpopulated on Raspberry Pi Zero, Raspberry Pi Zero W, and Raspberry Pi Zero 2 W. The GPIO headers on all boards have a 0.1in (2.54mm) pin pitch.

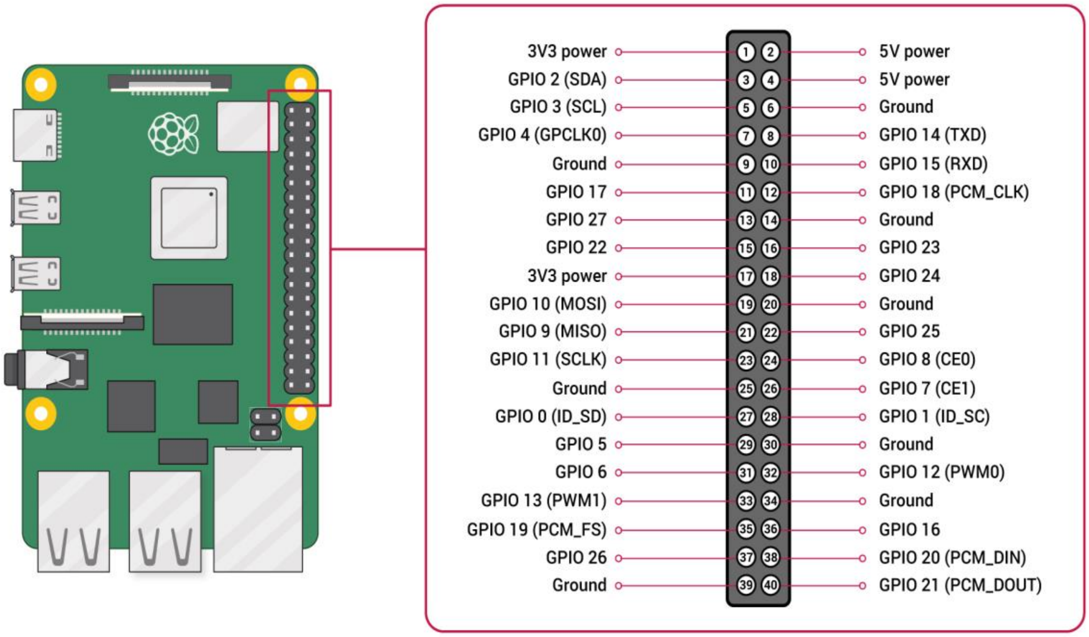


Figure 3.4: Diagram of the GPIO Pins

Voltages

Two 5V pins and two 3.3V pins are present on the board, as well as a number of ground pins (GND), which can not be reconfigured. The remaining pins are all general-purpose 3.3V pins, meaning outputs are set to 3.3V and inputs are 3.3V-tolerant.

Outputs

A GPIO pin designated as an output pin can be set to high (3.3V) or low (0V).

Inputs

A GPIO pin designated as an input pin can be read as high (3.3V) or low (0V). This is made easier with the use of internal pull-up or pull-down resistors. Pins GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins this can be configured in software.

Other functions

As well as simple input and output devices, the GPIO pins can be used with a variety of alternative functions, some are available on all pins, others on specific pins.

- **PWM (pulse-width modulation)**

  - Software PWM available on all pins
  - Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19

- **SPI**

  - SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
  - SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)

- **I2C**

  - Data: (GPIO2); Clock (GPIO3)
  - EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)

- **Serial**

  - TX (GPIO14); RX (GPIO15)

3.4.3 Features

• VideoCore VII GPU, supporting OpenGL ES 3.1, Vulkan 1.2

• Dual 4Kp60 HDMI® display output with HDR support

• 4Kp60 HEVC decoder

• LPDDR4X-4267 SDRAM (4GB and 8GB SKUs available at launch)

• Dual-band 802.11ac Wi-Fi® • Bluetooth 5.0/Bluetooth Low Energy (BLE)

• microSD card slot, with support for high-speed SDR104 mode

• 2 × USB 3.0 ports, supporting simultaneous 5Gbps operation

• 2 × USB 2.0 ports

• Gigabit Ethernet, with PoE+ support (requires separate PoE+ HAT)

• 2 × 4-lane MIPI camera/display transceivers

• PCIe 2.0 x1 interface for fast peripherals (requires separate M.2 HAT or other adapter)

• 5V/5A DC power via USB-C, with Power Delivery support

• Raspberry Pi standard 40-pin header • Real-time clock (RTC), powered from external battery
• Power button

3.4.4 ADVANTAGES OF RASPBERRY PI 5

• **Improved Performance**:

- **Faster Processor**: The Raspberry Pi 5 features a more powerful CPU, the BCM2712, a quad-core ARM Cortex-A76 processor, which provides a significant performance boost compared to the previous models.
- **Enhanced GPU**: With the VideoCore VII GPU, the Pi 5 offers better graphics performance, making it suitable for more demanding multimedia applications and light gaming.

• **Increased Memory Options**:

- The Pi 5 comes with options for 4GB and 8GB of LPDDR4X RAM, providing more memory for multitasking and running more complex applications.

• **Better Connectivity**:

- **Gigabit Ethernet**: The inclusion of true Gigabit Ethernet (unlike the limited performance on Pi 4) allows for faster wired network connections.
- **Dual HDMI Outputs**: Support for dual 4K monitors through the dual micro-HDMI ports, enhancing its usability for media centers and digital signage.
- **USB 3.0 Ports**: Two USB 3.0 ports alongside two USB 2.0 ports offer faster data transfer rates for external drives and peripherals.

• **Enhanced Storage Options**:

- **PCIe Support**: The addition of a PCIe slot enables users to add high-speed storage solutions like NVMe SSDs, significantly improving read/write speeds compared to microSD cards.

- **SD Card Speed**: Improved microSD card performance due to better integration and bus speed.

- **Better Power Management**:

- **Efficient Power Usage**: Despite the increased performance, the Raspberry Pi 5 has optimized power consumption, making it efficient for continuous use in various projects.
- **Power over Ethernet (PoE)**: Native support for PoE with an additional HAT, making it easier to power the device via Ethernet cables, which is especially useful for networked applications.

- **Enhanced I/O Capabilities**:

- **More GPIO Pins**: Additional GPIO pins provide greater flexibility for connecting sensors, LEDs, and other peripherals.
- **Real-Time Clock (RTC)**: An integrated RTC allows the Pi 5 to keep accurate time even when powered off, which is crucial for time-sensitive applications.

- **Software and Ecosystem**:

- **Backward Compatibility**: The Raspberry Pi 5 maintains compatibility with accessories and software from previous models, ensuring a smooth transition for existing projects.
- **Active Community and Support**: The extensive Raspberry Pi community offers a wealth of resources, tutorials, and forums for troubleshooting and project ideas.

- **Improved Multimedia Capabilities**:

- **4K Video Playback**: Enhanced ability to handle 4K video playback smoothly, making it ideal for use as a media center.
- **Advanced Audio Features**: Better audio output quality and support for more audio formats.

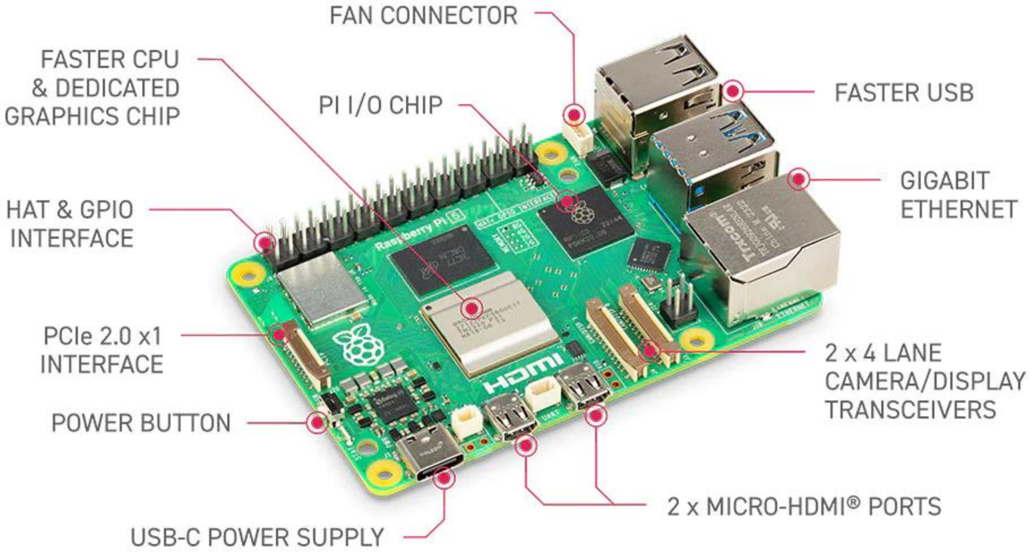


Figure 3.5 : Raspberry pi 5 Specifications

3.4.5 CAMERA

Figure 3.6: Camera

A webcam, short for "web camera," is a digital camera that captures video and sometimes audio, typically in real-time, and is connected to a computer or network. The camera lens captures the visual information, including images and videos, by focusing light onto the camera sensor. The image sensor converts the light from the camera lens into digital signals, creating a digital representation of the captured image or video. Common types of image sensors include CMOS (Complementary Metal-Oxide-Semiconductor) and CCD (Charge-Coupled Device). Many webcams also include a built-in microphone for capturing audio alongside the video. This allows users to have voice conversations or record audio to accompany the video. Webcams are typically connected to a computer or network device via a USB (Universal Serial Bus) cable or a wireless connection (e.g., Wi-Fi or Bluetooth). The interface allows the webcam to transmit the captured video and audio data to the connected device. Webcams often come with a mounting mechanism that allows them to be attached to various surfaces, such as the top of a computer monitor, a tripod, or a desk stand. This provides flexibility in positioning the webcam for optimal framing and stability. Webcams vary in resolution (measured in pixels) and frame rate (measured in frames per second), which determine the quality and smoothness of the captured video. Higher resolutions and frame rates result in clearer, smoother video footage. Some webcams feature auto-focus and digital zoom capabilities, allowing them to adjust focus automatically and zoom in or out on the subject without physically moving the camera lens. Webcams are typically compatible with various operating systems (e.g., Windows, macOS, Linux) and video conferencing software (e.g., Zoom, Skype, Microsoft Teams). They often come with drivers or software that enables configuration and customization of settings. In recent years, privacy features such as physical camera covers or built-in privacy shutters have become common in webcams. These features provide users with added privacy and security by physically blocking the camera lens when not in use. Webcams are used for a wide range of applications, including video conferencing, live streaming, online gaming, remote learning, telemedicine, surveillance, and content creation. They provide a convenient and cost-effective way to capture and transmit video and audio data over the internet.

3.4.6 SPEAKER

Figure 3.7: Speaker

A speaker is a device that converts electrical signals into sound waves, allowing users to listen to audio content such as music, speech, or other audio recordings. The driver is the core component of a speaker responsible for producing sound waves. It consists of one or more transducers, typically a diaphragm (cone) attached to a voice coil, which moves back and forth in response to electrical signals, causing air molecules to vibrate and produce sound waves. The enclosure, also known as the cabinet or housing, is the structure that houses the speaker components and helps to enhance the quality of sound reproduction. Enclosures are designed to minimize unwanted vibrations, resonances, and distortions, while also providing acoustic isolation and protection for the internal components. In multi-driver speaker systems (e.g., two-way or three-way speakers), a crossover network is used to split the audio signal into separate frequency bands and route them to the appropriate driver(s). This ensures that each driver receives only the frequencies it is designed to reproduce, resulting in more accurate sound reproduction across the frequency spectrum. Impedance is the electrical resistance of the speaker, measured in ohms (Ω). It represents the opposition to the flow of electrical current and affects the amount of power required to drive the speaker at a given volume level. Common speaker impedance ratings include 4 ohms, 8 ohms, and 16 ohms. Power handling, measured in watts (W), refers to the maximum amount of electrical power that a speaker can handle without being damaged. It determines the speaker's ability to handle high-volume levels and transient peaks in the audio sign. Frequency response describes the range of frequencies that a speaker can reproduce accurately, typically measured in Hertz (Hz). It indicates the speaker's ability to reproduce bass, midrange, and treble frequencies, with a wider frequency response range generally considered preferable for more accurate sound reproduction. Sensitivity, measured in decibels (dB), indicates how efficiently a speaker converts electrical power into sound output. Higher sensitivity ratings mean that the speaker can produce louder sound levels with less power input, making it more suitable for use with low-power amplifiers or portable audio devices. Speakers may feature various connectivity options, including wired connections (e.g., speaker wire terminals, audio jacks) and wireless connections (e.g., Bluetooth, Wi-Fi). Wireless speakers offer the convenience of streaming audio content from smartphones, tablets, and other devices without the need for physical cables. Speakers are used in a wide range of applications, including home audio systems, car audio systems, professional sound reinforcement systems, public address (PA) systems, multimedia and gaming setups, and portable audio devices. They play a crucial role in delivering high-quality audio experiences for entertainment, communication, and artistic expression.

3.4.7 POWER ADAPTER

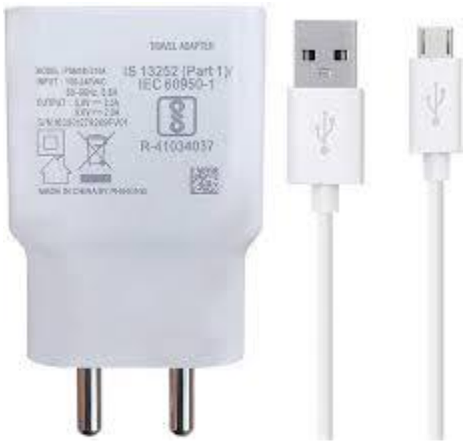


Figure 3.8: Power Adapter

A power adapter is a device that converts alternating current (AC) from a mains power outlet into direct current (DC) power for powering electronic devices. It has an input voltage and plug compatible with the corresponding outlet type and standard. The adapter contains a transformer or converter that converts the AC voltage to the required output voltage and current level. The output voltage and current are specified to match the electronic device's requirements. The adapter is equipped with a connector or cable that interfaces with the device, delivering DC power output. Power adapters are designed for high efficiency and energy-saving operation, minimizing power losses and reducing energy consumption. They are compatible with specific types of electronic devices and are certified to comply with safety standards established by regulatory agencies.

3.5 Software Description

3.5.1 Python

Python is a versatile and widely-used programming language known for its simplicity, readability, and flexibility .It has a simple and intuitive syntax that makes it easy for beginners to learn and use. Its readability and straightforward syntax resemble natural language, making it accessible to programmers of all levels of experience. It is a general-purpose programming language, meaning it can be used for a wide range of applications, including web development, data analysis, machine learning, artificial intelligence, scientific computing, automation, and more. Its versatility makes it a popular choice for developers across various domains. Python comes with a comprehensive standard library that provides a wide range of modules and functions for performing various tasks, such as file I/O, networking, data manipulation, and web development. This extensive library reduces the need for additional third-party packages and simplifies development. Python has a large and active community of developers who contribute to its ecosystem by creating libraries, frameworks, and tools. This vibrant community provides support, documentation, tutorials, and resources, making it easier for developers to find solutions to their problems and collaborate on projects. Python is a high-level programming language, meaning it abstracts away low-level details and provides a more user-friendly interface. This allows developers to focus on solving problems and writing code without worrying about memory management or system-level intricacies .Python is an interpreted language, which means that code is executed line by line by an interpreter rather than compiled into machine code. This makes the development process interactive and facilitates rapid prototyping and testing .Python is platform-independent, meaning it can run on various operating systems, including Windows, macOS, Linux, and Unix. This cross-platform compatibility allows developers to write code once and run it on different platforms without modification. Python is highly extensible, allowing developers to integrate code written in other languages, such as C/C++, Java, and Fortran. This interoperability enables developers to leverage existing libraries and legacy codebases, extending Python's functionality and performance .Python is scalable, meaning it can be used to build small scripts or large-scale applications. Its modular design and support for multiple programming paradigms (such as procedural, object-oriented, and functional programming) make it suitable for projects of any size or complexity. Python is open-source software, meaning its source code is freely available and can be modified, redistributed, and used for commercial purposes. This open-source nature fosters collaboration, innovation, and community-driven development, ensuring Python remains relevant and up-to-date with emerging trends and technologies.

3.5.2 OPEN CV

OpenCV, or Open Source Computer Vision Library, is a popular open-source computer vision and machine learning software library. OpenCV provides a wide range of functions and algorithms for image processing, computer vision tasks, and machine learning. It enables developers to perform tasks such as image filtering, edge detection, object detection and tracking, facial recognition, and more. OpenCV is designed to be cross-platform and can run on various operating systems, including Windows, macOS, Linux, iOS, and Android. This makes it suitable for developing computer vision applications for desktop, mobile, and embedded platforms. OpenCV features an extensive library of optimized algorithms and data structures for computer vision and image processing tasks. It includes functions for manipulating images, performing geometric transformations, extracting features, and analyzing visual data. OpenCV is primarily written in C++ but provides bindings for several programming languages, including Python, Java, and MATLAB. This allows developers to use OpenCV in their preferred programming language and integrate it with existing projects and frameworks. OpenCV can be integrated with other popular libraries and frameworks, such as NumPy, SciPy, TensorFlow, and

PyTorch. This enables developers to leverage the capabilities of these libraries in combination with OpenCV for more advanced computer vision and machine learning tasks. It provides functions for capturing video from cameras, processing frames in real-time, and displaying results with minimal latency. OpenCV has a large and active community of developers, researchers, and enthusiasts who contribute to its development and support. The library is well-documented, with extensive tutorials, guides, and examples available to help users get started and troubleshoot issues. OpenCV is distributed under the open-source BSD license, allowing users to use, modify, and distribute the library freely for both commercial and non-commercial purposes. This makes it accessible to a wide range of users and encourages collaboration and innovation in the field of computer vision. OpenCV is continuously developed and maintained by a dedicated team of developers and contributors. New features, improvements, and bug fixes are regularly released, ensuring that the library remains up-to-date with the latest advancements in computer vision and machine learning. OpenCV is used in a wide range of applications across various industries, including robotics, augmented reality, medical imaging, surveillance, automotive, agriculture, and more. Its versatility and flexibility make it suitable for solving diverse challenges in computer vision and image processing.

3.5.3 DLIB

Dlib is a versatile C++ library that also provides Python bindings, making its powerful functionality accessible to Python developers. Dlib Python library offers a wide range of machine learning algorithms and computer vision tools for Python developers. It includes functionalities for tasks such as object detection, face detection and recognition, image segmentation, shape prediction, and more. The Dlib Python library provides a user-friendly interface for Python developers, allowing them to leverage Dlib's functionality without needing to write C++ code. Python developers can use Dlib's Python API to perform various tasks in machine learning and computer vision with ease.The Dlib Python library is compatible with Python 2 and Python 3, making it accessible to a wide range of Python developers. It can be used on various operating systems, including Windows, macOS, Linux, and Unix-like systems. Dlib Python library can be easily integrated with other popular Python libraries and frameworks, such as NumPy, SciPy, OpenCV, TensorFlow, and PyTorch. This allows Python developers to combine the capabilities of different libraries to build more powerful and versatile software solutions. The Dlib Python library comes with extensive documentation and a rich set of examples to help Python developers get started quickly. The documentation provides detailed explanations of Dlib's functionalities, while the examples demonstrate how to use Dlib in practice for various tasks.Despite being written in C++, the Dlib Python library offers high performance when used in Python applications. It leverages the underlying C++ implementations of machine learning algorithms and computer vision tools, providing efficient execution and low latency.

3.5.4 IMUTILS

A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much easier with OpenCV and both Python 2.7 and Python 3.

Imutils is a widely-used Python library designed to simplify basic image processing tasks and common computer vision operations. Developed by Adrian Rosebrock, the creator of the PyImageSearch blog, imutils provides a collection of convenience functions that make it easier to work with the OpenCV library.

3.5.5 SCIPY

- SciPy is a scientific computation library that uses NumPy underneath.
- SciPy stands for Scientific Python.
- It provides more utility functions for optimization, stats and signal processing.
- Like NumPy, SciPy is open source so we can use it freely.
- SciPy was created by NumPy's creator Travis Olliphant.

3.5.6 NUMPY

NumPy, short for Numerical Python, is a fundamental library for scientific computing in Python. It provides support for arrays, matrices, and a wide range of mathematical functions to operate on these data structures. At its core, NumPy offers a powerful N-dimensional array object, which is a versatile and efficient data structure that allows for fast computation and manipulation of large datasets. This array object forms the basis for almost all numerical operations in the library.

- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.

3.6 WORKFLOW

3.6.1 Detecting the Face of the User

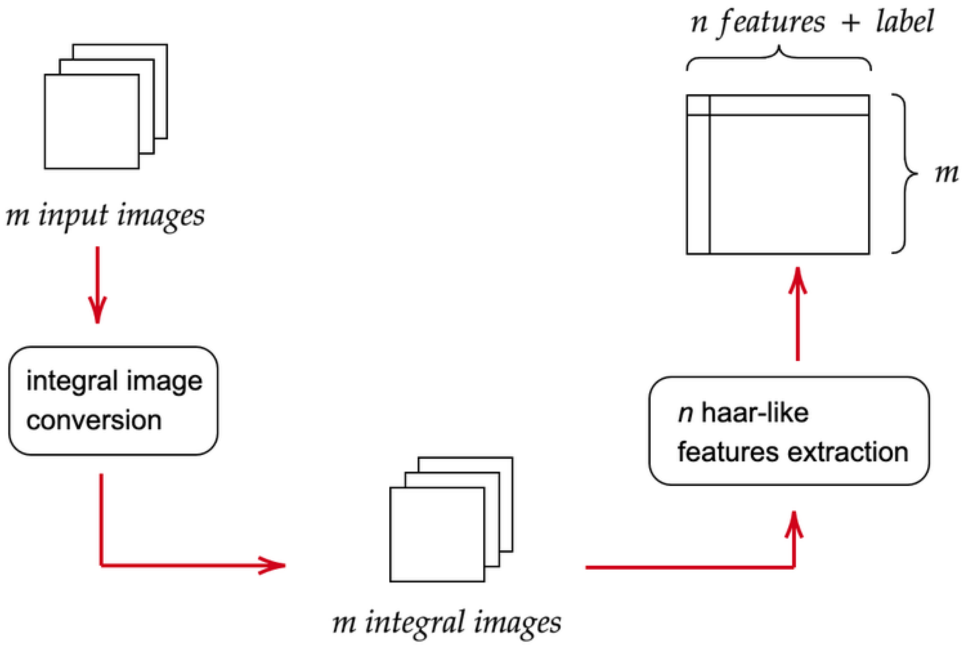


Figure 3.9: Detecting Flow Diagram

So, before we can find if the user is drowsy or not, we need to detect the face of the user. Now there are many algorithms available to use, but here we have to use an algorithm that requires less processing power (so that we can run it in Raspberry Pi), which will be fast, and finally it should be accurate too. Thus We decided to use the famous "Viola-Jones" algorithm for face detection.

3.6.2 Detecting If the Eyes Are Opened or Closed

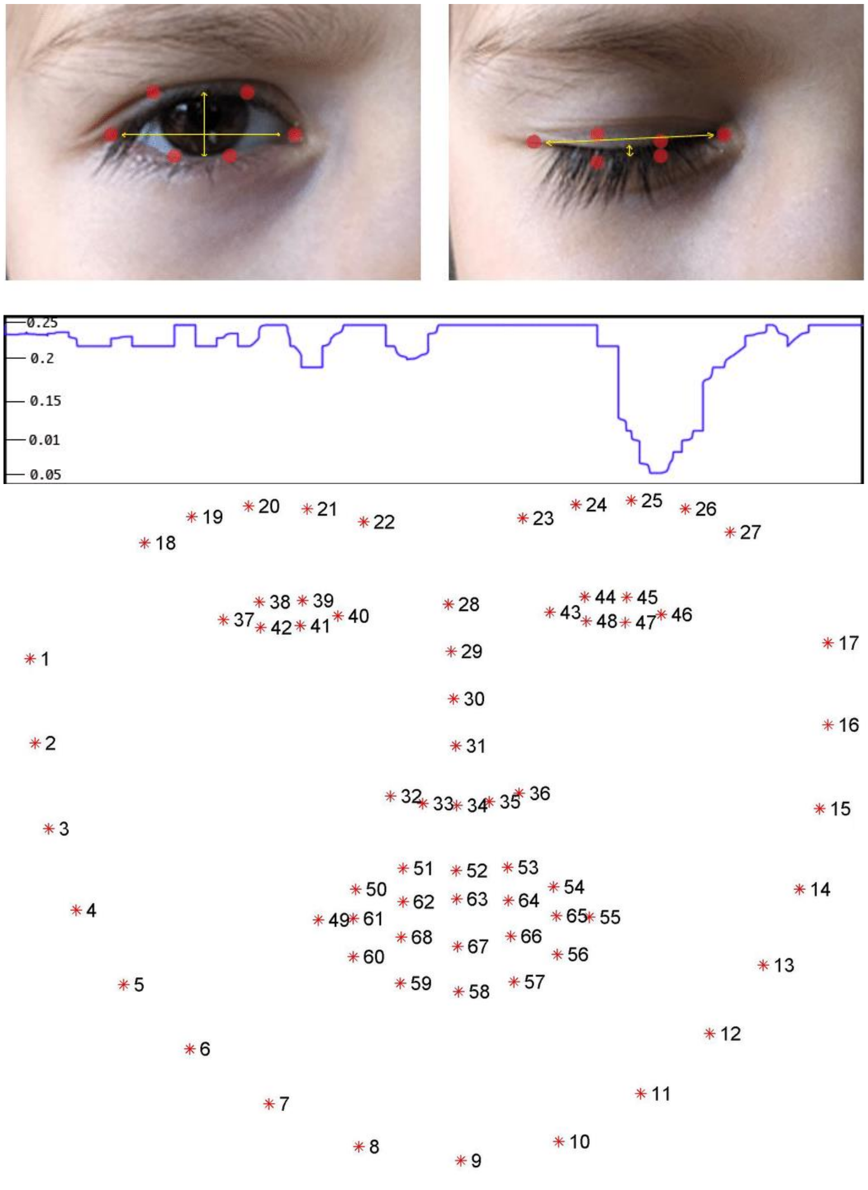


Figure 3.10: Face and Eye Landmarks Diagram

To detect if the user is sleeping or not, we have to find out if the user's eyes are open or not. To find out that, we are going to use the Eye-Aspect-Ratio (EAR). The average eye aspect ratio is 0.339 and 0.141 when the eyes are opened and closed, respectively. So whenever our system will detect a face, it will calculate the EAR and if it's below the threshold (set by the user), then it will alert the user and it will alert the user continuously until the user will open the eyes.

Now to calculate the EAR, we need to find out the eye landmarks in the face (as you can see in the figure). To find out these landmarks, we will use Dlib's 68 facial landmark model, which is a pre-trained model and can be easily used with python. It is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person's face like the above figure.

Finally, after obtaining the points, we can find the EAR, using the formula, EAR = (|(P2-P6)| + |(P3-P5)|) / (2* |(P1 - P4)|) . Next, we will check if this EAR value is within the threshold or not, and based on that the system will alert the user.

3.6.3 Detecting the Yawn

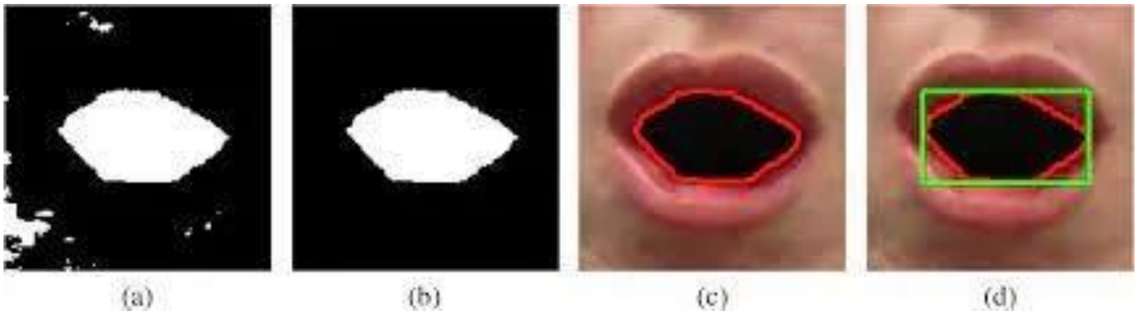
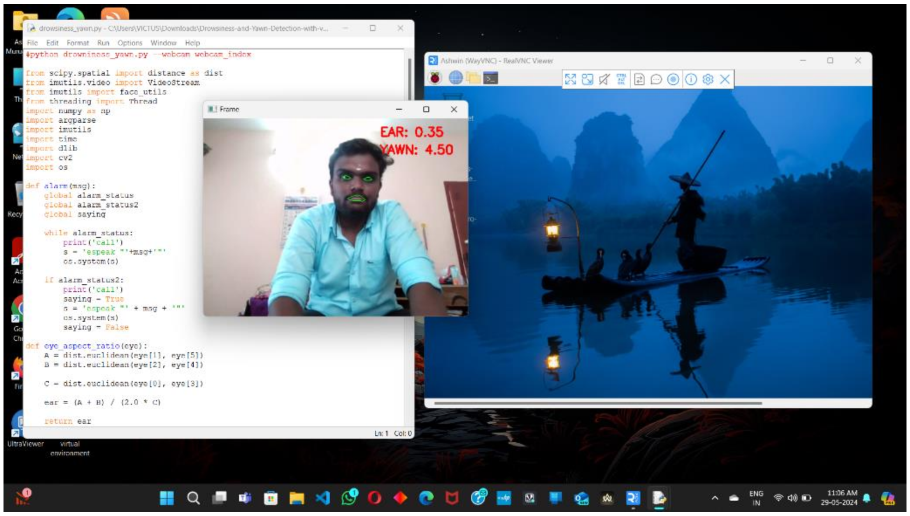


Figure 3.11: Yawn Detection Diagram

To detect the yawn, we need to find out the distance between the user's upper lip and lower lip. So, when a person is talking this distance will be within a limit, but when the person will take a yawn, the distance will be much higher than the limit or threshold. Now to find out the distance between two lips, we need to find out the landmarks of the lips and again we will use the DLIB's facial landmark model here. Then we will simply calculate the distance between the midpoint of the upper lip to the midpoint of the lower lip. And if this distance is more than the threshold, the system will give a yawn alert to the user.

CHAPTER 4

4.1 SIMULATION RESULT


drowsiness_yawn.py - C:\Users\VICTUS\Downloads\Drowsiness-and-Yawn-Detection-with-v...
File Edit Format Run Options Window Help
Frame
EAR: 0.35
YAWN: 4.50
Ashwin (WayVNC) - RealVNC Viewer
Ln: 1 Col: 0
UltraViewer
virtual environment
ENG IN
11:06 AM
29-05-2024

```
#python drowsiness_yawn.py --webcam webcam_index

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

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

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 4.1: Simulation in Raspberry pi 5

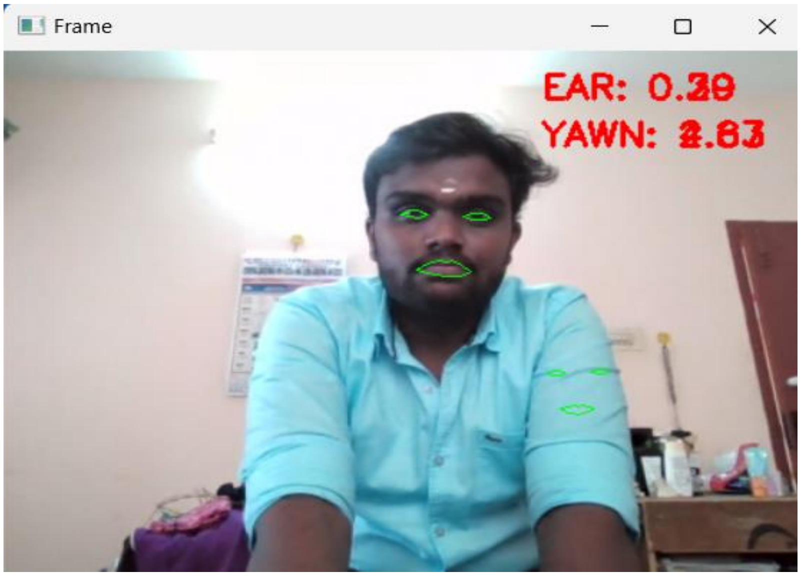
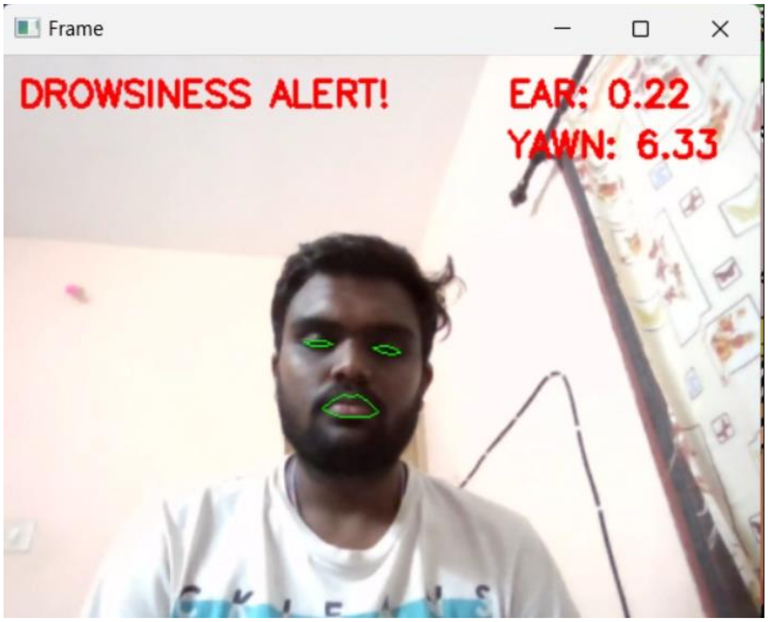
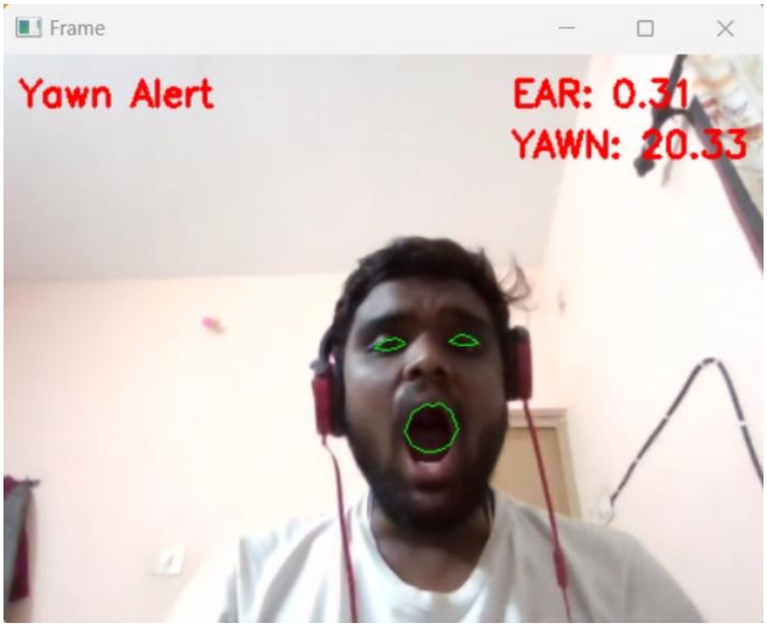


Figure 4.2: Result of Normal Condition



```
DROWSINESS ALERT!          EAR: 0.22
                           YAWN: 6.33
```

Figure 4.3: Result of Drowsiness Alert



```
Yawn Alert                 EAR: 0.31
                           YAWN: 20.33
```

Figure 4.4: Result of Yawn Detection

4.1 HARDWARE RESULT

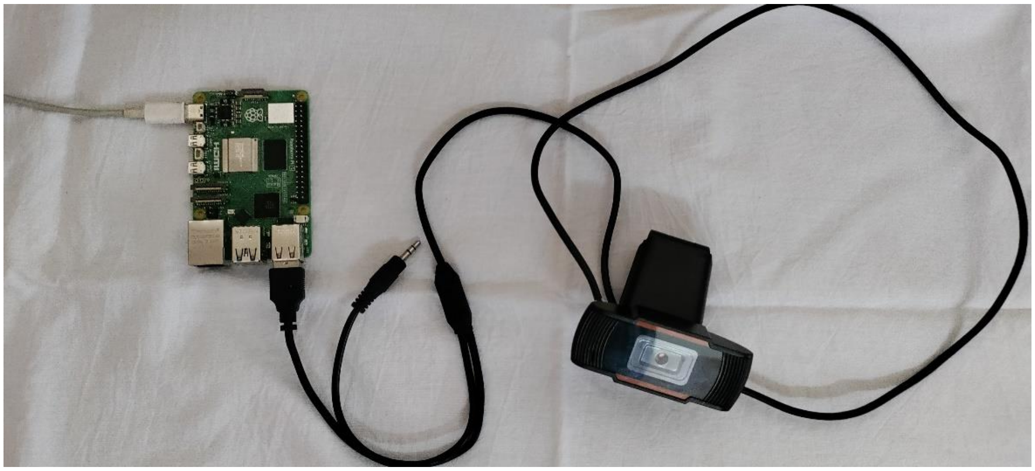


Figure 4.5: Hardware Connection

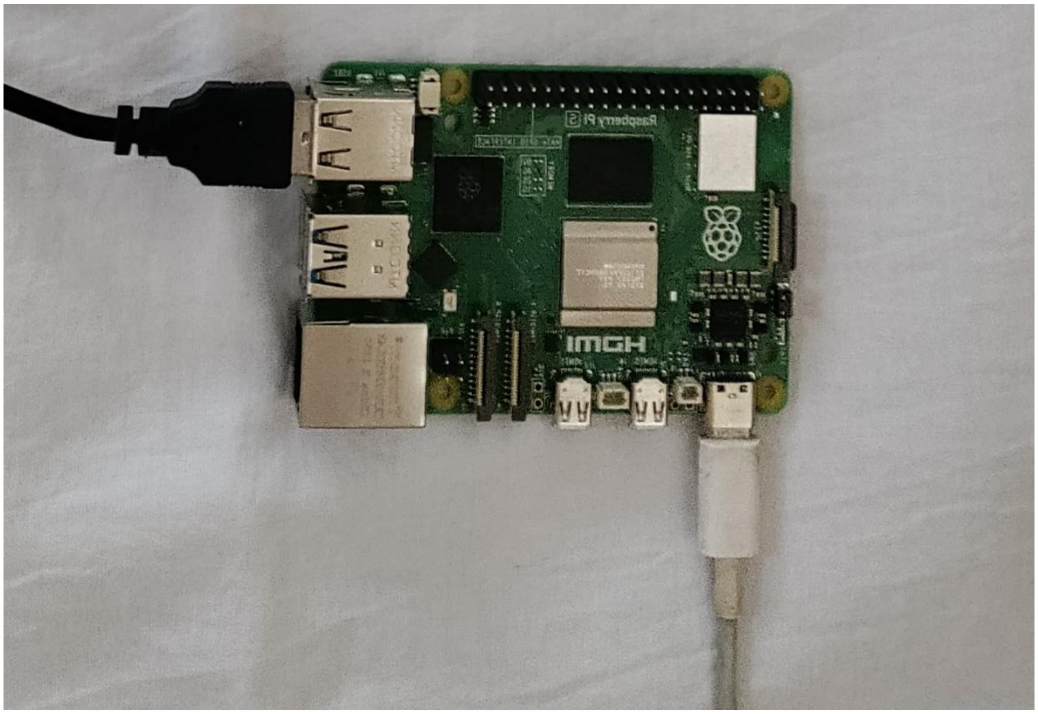


Figure 4.6: Raspberry pi 5

4.2 DISCUSSION:

The successful implementation of the Driver Sleep Detection and Alarming System underscores the importance of proactive measures in addressing the risks associated with drowsy driving. By leveraging technology to monitor driver alertness and intervene when necessary, the system demonstrated its effectiveness in preventing accidents and promoting road safety.

However, several considerations should be taken into account when discussing the project:

- **Accuracy and Reliability**: While the system showed promising results in detecting drowsiness, further research and development are needed to improve its accuracy and reliability. Fine-tuning the algorithms and integrating additional sensor inputs may enhance the system's performance in diverse driving conditions and scenarios.

- **User Acceptance and Adaptation**: The success of the system relies on the acceptance and adaptation of drivers to its alerts and recommendations. Educating drivers about the importance of road safety and the risks of drowsy driving is crucial for fostering acceptance and compliance with the system's recommendations.

- **Integration with Existing Systems**: The Driver Sleep Detection and Alarming System should be seamlessly integrated with existing vehicle safety systems and infrastructure to maximize its effectiveness. Collaborating with automotive manufacturers and transportation companies to standardize the implementation of such systems can facilitate widespread adoption and impact.

Overall, the Driver Sleep Detection and Alarming System represents a significant advancement in vehicle safety technology, with the potential to save lives and reduce accidents caused by drowsy driving. Continued research, development, and collaboration are needed to further refine and deploy the system effectively in real-world driving environments.

CHAPTER 5

CONCLUSION

In conclusion, the Driver Sleep Detection and Alarming System represent a pivotal advancement in automotive safety technology. By leveraging sophisticated sensors, machine learning algorithms, and real-time monitoring capabilities, this system addresses the critical issue of drowsy driving, which poses significant risks to road safety worldwide.

Through continuous monitoring of driver behavior and physiological parameters, the system can accurately detect signs of drowsiness and issue timely alerts to prevent potential accidents. Its ability to integrate seamlessly into various vehicles, including commercial fleets and personal vehicles, makes it a versatile solution with broad applicability.

The benefits of implementing such a system are undeniable. Enhanced safety, reduced accident rates, improved driver well-being, and compliance with regulations are among the key advantages.

REFERENCES

1. K. Satish , A. Lalitesh ,K. Bhargavi, M. Sishir Prem ,T Anjali “ Driver Drowsiness Detection”2020 International Conference on Communication and Signal Processing (ICCSP).

2. Belal Alshaqaqi ,Abdullah Salem Baquhaizel ,Mohamed El Amine Ouis ,Meriem Boumehed ,Abdelaziz Ouamri “Driver drowsiness detection system” 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)

3. M. Omidyeganeh ,A. Javadtalab ,S. Shirmohammadi “Intelligent driver drowsiness detection through fusion of yawning and eye closure” 2011 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems Proceedings

4. Arpit S Agarkar , R Gandhiraj ,Manoj Kumar Panda “ Driver Drowsiness Detection and Warning using Facial Features and Hand Gestures” 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)

5. P Baby Shamini ,M. Vinodhini ,B Keerthana ,S Lakshna ,K. R Meenatchi “ Driver Drowsiness Detection based on Monitoring of Eye Blink Rate” 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)

6. N Prasath ,J Sreemathy , P Vigneshwaran “ Driver Drowsiness Detection Using Machine Learning Algorithm” 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)

7. Nawazish Ali ,Imran Hasan ,Tansel Özyer ,Reda Alhajj “ Driver Drowsiness Detection by Employing CNN and Dlib” 2021 22nd International Arab Conference on Information Technology

8. Meena Siwach , Suman Mann ,Deepa Gupta " A Practical Implementation of Driver Drowsiness Detection Using Facial Landmarks" 2022 10th International Conference on Reliability, Infocom Technologies and Optimization .

9. Dharam Buddhi ,Poonam Negi " Deep Learning based Driver Drowsiness Detection" 2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC).

10. Ali Mansour, Al madani, Ashok T. Gaikwad, Vivek Mahale " Real-time Driver Drowsiness Detection based on Eye Movement and Yawning using Facial Landmark" 2021 International Conference on Computer Communication and Informatics (ICCCI).

11. Jithina Jose,J S Vimali , P Ajitha "Drowsiness Detection System for Drivers Using Image Processing Technique" 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)

12. Shoyab Alam Idrisi ,Tushar Gupta "Driver Drowsiness Detection System with OpenCV & Keras" 2021 International Conference on Simulation, Automation & Smart Manufacturing (SASM).

13. Janki Chandiwala, Shrushti Agarwal "Driver's real-time Drowsiness Detection using Adaptable Eye Aspect Ratio and Smart Alarm System" 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS).

14. R Kannan, Palamakula Jahnavi, M Megha "Driver Drowsiness Detection and Alert System" 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)

15.Meng Zhang, Fumin Zhang " A Driver Fatigue Detection Method Based on Eye Aspect Ratio Compensated by Head Pose Estimation" 2023 8th International Conference on Signal and Image Processing (ICSIP)

APPENDIX

PROGRAM:

```
#python drowniness_yawn.py --webcam webcam_index

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

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

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

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

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

    return ear

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

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

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

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


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

EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 20
alarm_status = False
alarm_status2 = False
saying = False
COUNTER = 0

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


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

```
while True:

    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #rects = detector(gray, 0)
    rects = detector.detectMultiScale(gray, scaleFactor=1.1,
                        minNeighbors=5, minSize=(30, 30),
                        flags=cv2.CASCADE_SCALE_IMAGE)

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

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

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

        distance = lip_distance(shape)

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

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

        if ear < EYE_AR_THRESH:
            COUNTER += 1

            if COUNTER >= EYE_AR_CONSEC_FRAMES:
                if alarm_status == False:
                    alarm_status = True
                    t = Thread(target=alarm, args=('wake up sir',))
                    t.deamon = True
                    t.start()

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

        else:
            COUNTER = 0
            alarm_status = False

        if (distance > YAWN_THRESH):
                cv2.putText(frame, "Yawn Alert", (10, 30),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
                if alarm_status2 == False and saying == False:
                    alarm_status2 = True
                    t = Thread(target=alarm, args=('take some fresh air sir',))
                    t.deamon = True
                    t.start()
        else:
            alarm_status2 = False

        cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        cv2.putText(frame, "YAWN: {:.2f}".format(distance), (300, 60),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


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

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

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