



DROWSINESS DETECTION SYSTEM USING A HARDWARE MODULE

¹Adish P S, ²Anas H, ³Ansiya S, ⁴Vishnupriya Rajan, ⁵Dr Kanchana J

Professor Department of Computer Science Engineering,

Younus College of Engineering and Technology, Pallimukku, Kollam, Kerala, India.

Email id: kanchanasukumar@gmail.com

Abstract—This paper presents a novel drowsiness detection system utilizing a hardware module to improve driver safety by continuously monitoring alertness in real-time. Built around a Raspberry Pi 5 and a REES52 camera module, the system employs advanced computer vision techniques to analyze facial features, particularly eye movements, to detect signs of fatigue. When drowsiness is identified, a multi-modal alert mechanism, including sound and vibration, is activated to prompt immediate driver response. Designed for affordability and ease of integration into vehicles, this hardware-based solution achieves high accuracy and operates efficiently under varying conditions, offering a practical approach to reducing road accidents caused by driver fatigue.

Index Terms—Raspberry Pi 5, REES52 Camera, Ribbon Cable, Drowsiness Detection, Hardware Module, Eye Aspect Ratio, Driver Safety

I. INTRODUCTION

Driver drowsiness is a persistent global challenge, contributing significantly to vehicular accidents and posing risks to both public safety and economic stability. Fatigue impairs reaction times and decision-making abilities, often leading to catastrophic outcomes on the road. While awareness campaigns and regulatory measures have made strides in addressing this issue, the need for automated, reliable, and accessible detection systems remains critical. This paper introduces a drowsiness detection system using a hardware module, aimed at providing an effective solution through real-time monitoring and intervention.

The proposed system leverages a compact hardware setup centered on the Raspberry Pi 5, a versatile microcontroller, paired with a REES52 camera module to capture and process visual data. Unlike software-centric approaches that demand substantial computational resources, this hardware-based design prioritizes efficiency and independence, making it suitable for integration into a variety of vehicles. By focusing on key indicators such as eye closure duration and head position, the system delivers timely alerts to prevent potential accidents, enhancing safety across diverse driving scenarios.

This study outlines the system's development, from hardware selection to algorithmic implementation, emphasizing its practicality and scalability. The integration of a multi-modal alert system ensures that drivers are promptly notified of their drowsy state, encouraging corrective actions like resting. Subsequent sections explore prior research, detail the system's components and methodology, present performance results, and discuss future possibilities for enhancement.

II. LITERATURE REVIEW

Research into drowsiness detection has evolved significantly, with numerous studies exploring both hardware and software-based approaches to mitigate fatigue-related risks. Early efforts focused on physio-logical sensors, such as those monitoring heart rate or brain activity, but these often proved invasive and impractical for widespread use. More recent advancements have shifted toward vision-based systems, leveraging cameras and machine learning to analyze facial behavior, offering non-intrusive and scalable alternatives.

A notable study by Florez et al. (2024)[1] employed a Convolutional Neural Network (CNN) and a near-infrared camera on an NVIDIA Jetson Nano, achieving exceptional accuracy in detecting drowsiness through eye and mouth analysis. Similarly, Safarov et al. (2023)[2] utilized deep learning for eye-blink detection, demonstrating robust performance across diverse conditions. Wearable solutions, such as those proposed by Li et al. (2023)[3], introduced lightweight CNNs integrated into glasses, focusing on eye closure metrics. These works highlight the potential of AI driven systems, though their reliance on specialized hardware can limit affordability and adaptability.

Other research, including Jain et al. (2021)[4] and Biswal (2021)[5], explored simpler hardware setups with real-time computer vision and IoT integration, respectively. These studies emphasize cost-effectiveness and immediate alerting, aligning closely with practical deployment needs. However, challenges such as performance in low-light environments and computational overhead persist across these solutions. This paper builds on these foundations, proposing a hardware module that balances accuracy, cost, and real-world applicability, contributing to the ongoing effort to enhance driver safety through innovative technology. positioning this work as a valuable contribution to automotive safety technology.

III. SYSTEM COMPONENTS

A. Raspberry Pi 5

The Raspberry Pi 5, a powerful single-board computer, forms the backbone of the hardware module. Equipped with a quad-core processor and ample memory, it handles real-time video processing and algorithm execution efficiently. Its versatility and support for peripherals make it an ideal choice for embedded applications like drowsiness detection.

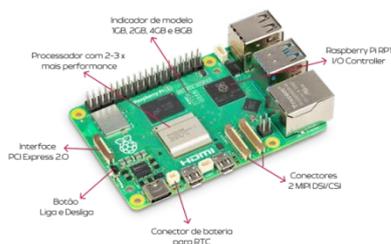


Figure 1:Raspberry Pi 5

Beyond its processing capabilities, the Raspberry Pi 5 offers a cost-effective platform that supports extensive software libraries, facilitating the integration of computer vision and alert systems. Its compact size ensures easy installation within a vehicle's dashboard or console, while its robust design withstands prolonged operation, critical for continuous monitoring during long drives.

B. REES52 Camera Module

The REES52 camera module is a high-resolution imaging device tailored for the Raspberry Pi ecosystem. Capable of capturing detailed video footage, it provides the visual input necessary for analyzing driver behavior. Its adjustable focus and small footprint make it well-suited for discreet placement within a vehicle.



The camera's ability to operate at sufficient frame rates ensures that subtle changes in facial expressions, such as eye blinks, are captured accurately. Its compatibility with the Raspberry Pi simplifies setup, while its low power consumption aligns with the system's efficiency goals. This component is pivotal in delivering the raw data that drives the drowsiness detection process.

C. Buzzer

A buzzer serves as the primary alert mechanism, emitting an audible signal to rouse a drowsy driver. Integrated with the Raspberry Pi, it activates based on predefined drowsiness thresholds, providing an immediate and unmistakable warning.

The simplicity of the buzzer enhances its reliability, requiring minimal power and maintenance. Its role extends beyond mere notification, acting as a critical link between detection and driver response, ensuring that alerts are both timely and effective in preventing accidents.

D. 5V 5A Power Supply

The 5V 5A Power Supply is vital for the Drowsiness Detection System, delivering a stable 5-volt, 5-amp output to power demanding

components like the Raspberry Pi 5 and REES52 Camera Module.



Figure 3: 5V 5A Power Supply

It ensures reliable operation for resource-intensive tasks like real-time image processing and machine learning, preventing power fluctuations that could affect accuracy. With ample current, it supports the Raspberry Pi, camera, and additional peripherals, maintaining performance and video quality for drowsiness detection. Equipped with overcurrent, overvoltage, and short-circuit protection, it enhances safety, efficiency, and durability, making it ideal for continuous monitoring.

E. Raspberry Pi OS

Raspberry Pi OS, built on Debian Linux, is the official operating system for Raspberry Pi devices, optimized for its ARM architecture. It offers a lightweight, stable, and user-friendly environment, ideal for embedded systems like the Drowsiness Detection System.

With pre-installed Python and support for libraries like OpenCV, it's perfect for real-time image processing and machine learning tasks, such as analyzing facial expressions and eye movements for drowsiness detection. Its efficient resource management ensures smooth performance despite the Raspberry Pi's limited power, while built-in tools enhance reliability and optimization, making it a secure and adaptable choice for continuous operation.

F. Python 3

Python 3 is a versatile, high-level programming language prized for its simplicity and extensive libraries, making it ideal for the Drowsiness Detection System. Its clear syntax speeds up development and debugging, supporting tasks like image processing and hardware interfacing on the Raspberry Pi. With libraries like OpenCV, Python 3 enables real-time video analysis for detecting drowsiness indicators such as eye movements, while compatibility with TensorFlow and scikit learn allows future enhancements. Running efficiently on Raspberry Pi OS, Python 3 leverages its cross platform strength and community support to

ensure reliable, adaptable performance for the system.

G. OpenCV

OpenCV, an open-source library initially developed by Intel, is essential for real-time image processing and computer vision, widely used in fields like robotics and medical imaging. In the Drowsiness Detection System, it captures and analyzes video data to detect drowsiness indicators such as blinking patterns and facial expressions using Haar Cascade classifiers for face and eye tracking. Optimized for performance, OpenCV efficiently processes video on resource-limited devices like the Raspberry Pi, integrating seamlessly with Python and Raspberry Pi OS. Its flexibility and robust features make it ideal for accurate, responsive drowsiness detection with potential for further enhancements.

H. Jupyter Notebook

Jupyter Notebook is an open-source web application widely used in research and development for data analysis, machine learning, and scientific computing. In the Drowsiness Detection System, it provides an interactive environment to test and refine code for data preprocessing, model training, and real-time analysis with immediate feedback. Its ability to integrate code and data visualization is key for testing OpenCV functions and machine learning algorithms, allowing developers to monitor and adjust drowsiness detection algorithms efficiently before deployment on a Raspberry Pi. Compatible with Python and libraries like NumPy, Pandas, and Matplotlib, it also enhances documentation and reproducibility, making it essential for the system's iterative development and testing.

I. Webpage Designing

1) Frontend: HTML: HyperText Markup Language (HTML) is the standard language for creating and structuring web pages. It defines elements such as headings, paragraphs, images, and forms using a tag-based system. In this project, HTML structures the user interface for displaying drowsiness detection results, ensuring a clear layout for real-time feedback. Semantic elements like `<header>`, `<section>`, and `<footer>` enhance accessibility, while `<form>` elements enable user interaction, such as adjusting alert settings.

CSS: Cascading Style Sheets (CSS) controls the

presentation and layout of the webpage. It defines styles for colors, fonts, and spacing, ensuring a visually

appealing and responsive design. In this system, CSS uses media queries and flexible units (e.g., percentages, em) to adapt the interface across devices, such as in-vehicle dashboards or mobile screens. Features like animations and Flexbox enhance usability by providing smooth transitions and organized layouts for alert displays.

JavaScript: JavaScript adds interactivity to the webpage, enabling real-time updates and event handling.

It processes data from the Raspberry Pi, dynamically updating the UI with detection results (e.g., eyestate indicators). Using the Document Object Model (DOM), JavaScript modifies HTML elements on-the-fly, while the Fetch API retrieves live data from the backend, ensuring seamless communication between the system and the user interface.

2) Backend: Flask: Flask, a lightweight Python web framework, powers the backend of the drowsiness detection system. It handles HTTP requests and serves dynamic content, such as video snippets and detection logs, to the frontend. Using Jinja2 templates,

Flask renders real-time data into HTML pages, while its routing system manages URL endpoints for alert notifications and system status. Flask's integration with Python allows it to process data from OpenCV and the Raspberry Pi efficiently, making it ideal for this resource-constrained application.

SQLite: SQLite is a lightweight, serverless, embedded relational database used to store and manage drowsiness detection data in this system. It records critical information such as timestamps, Eye Aspect Ratio (EAR) values, alert triggers, and video snippet references in a structured format. SQLite's file based architecture ensures easy integration with the Raspberry Pi, requiring minimal setup and no separate server process, which aligns with the system's goal of resource efficiency. In this project, Flask interfaces with SQLite using Python's sqlite3 module to log drowsiness events and retrieve historical data for analysis or display on the webpage, enhancing system accountability and enabling post-event review.

J. Software Libraries

The drowsiness detection system leverages several Python libraries to enable real-time

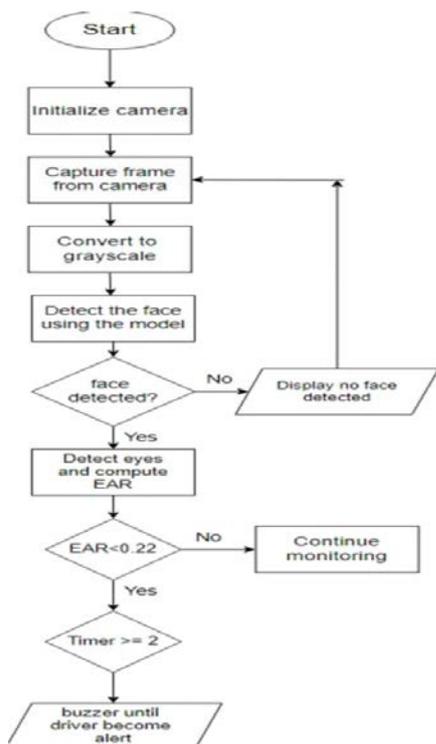
video processing, facial landmark detection, and data management. The following libraries are utilized:

- os: Provides a platform-independent interface to interact with the operating system, used for file and directory operations such as saving video snippets.
- time: Offers time-related functions, employed to manage timing operations like delays and times tamp generation for drowsiness event logs.
- dlib: A machine learning library providing pretrained models for facial landmark detection, critical for identifying eye and facial features using the iBUG 300-W dataset.
- pygame: Facilitates multimedia handling, used here for playing audio alerts (e.g., buzzer sounds) when drowsiness is detected.
- tkinter: A standard Python GUI toolkit, utilized with filedialog to create a basic interface for file selection or system interaction.
- PIL (Python Imaging Library): Imported as Image and ImageTk, it processes and converts images for display within the Tkinter interface.
- numpy: A numerical computing library, used as np for efficient array operations and mathematical computations, such as EAR calculations.
- picamera2: Interfaces with the Raspberry Pi camera module (REES52), enabling real-time video capture and frame acquisition.
- cv2 (OpenCV): An open-source computer vision library, essential for video processing, face detection, and landmark extraction in real-time.
- imutils: A utility library enhancing OpenCV functionality, imported for face_utils to simplify facial landmark handling.
- scipy.spatial.distance: Computes spatial distances, used here to calculate the Eye Aspect Ratio (EAR) based on landmark coordinates.
- datetime: Provides date and time handling, employed to timestamp video snippets and drowsiness events for storage and logging.
- collections.deque: Implements a double-ended queue, used to maintain a rolling buffer of EAR values for smoothing and threshold analysis.
- requests: Facilitates HTTP requests, enabling communication between the Raspberry Pi and the Flask backend for sending video data to the website.
- subprocess: Executes system commands, potentially used to trigger external processes like network operations or hardware controls.

These libraries collectively enable the system's core functionalities, from video capture and analysis to alert generation and data transmission, ensuring efficient performance on the Raspberry Pi platform.

IV. SYSTEM FLOW CHART

The system's operational flow is depicted in a flowchart, outlining the sequence from video acquisition to alert generation. It begins with camera initialization, followed by frame capture, facial analysis, drowsiness assessment, and concludes with alert activation if necessary.



This visual representation clarifies the decision-making process, where each step is optimized for speed and accuracy. The flowchart also aids in identifying potential bottlenecks, guiding future refinements to enhance system responsiveness and reliability in real, world scenarios.

V. METHODOLOGY

The methodology encompasses the development and operation of the drowsiness detection system, structured into distinct phases. Initially, the hardware is configured, followed by continuous video capture and processing to detect facial landmarks. Drowsiness is assessed using the Eye Aspect Ratio (EAR), and alerts are triggered when predefined thresholds are breached. The system employs a pre-trained model from the Dlib library, utilizing the iBUG

300-W dataset to ensure accuracy across diverse facial features and lighting conditions. Video frames are converted to grayscale for efficient processing, with algorithms calculating EAR to monitor eye openness. This hardware-driven approach minimizes latency, enabling real-time performance critical for safety applications. Additionally, the Raspberry Pi captures short video snippets when drowsiness is detected, saving them locally and transmitting them over the same network to the website's backend. The backend, powered by Flask and SQLite, stores these videos with corresponding date and time metadata, facilitating organized archival, also can be download video and post-event analysis.

Testing and calibration refine the system's sensitivity, setting an EAR threshold based on empirical data to differentiate normal blinks from prolonged closures indicative of drowsiness. The methodology emphasizes modularity, allowing for adjustments or upgrades, such as integrating additional sensors, to enhance detection capabilities further. The video-saving feature enhances accountability by providing a record of drowsiness events, accessible via the website for review or integration with safety protocols, leveraging the system's network connectivity for seamless data management.

VI. ADVANTAGES

The drowsiness detection system offers the following key benefits:

- **Low-Cost Implementation:** Its hardware-based design ensures affordability and easy integration into existing vehicles.
- **High Accuracy:** Provides reliable monitoring through precise detection capabilities.
- **Real-Time Processing:** Enables immediate drowsiness detection for timely intervention.
- **Multi-Modal Alerts:** Effectively captures driver attention using diverse alert mechanisms.
- **Adaptability:** Performs well under various conditions, enhancing practical utility.
- **Resource Independence:** Operates without external computational resources, suitable for remote or constrained environments.
- **Scalability:** Supports deployment in personal cars, commercial fleets, and industrial settings.
- **Safety Impact:** Positions the system as a transformative tool in combating fatigue-related incidents.

VII. APPLICATIONS

The system is applicable across multiple domains:

- Personal Vehicles: Enhances individual driver safety by preventing drowsy driving.
- Fleet Management: Ensures operator alertness, reducing liability and improving efficiency.
- Public Transportation: Increases passenger safety through reliable monitoring.
- Industrial Operations: Monitors workers handling heavy machinery to prevent accidents.
- Healthcare: Monitors patient alertness for medical applications.
- Education: Studies fatigue effects in academic settings.
- Smart Vehicles: Integrates into connected transportation systems due to its adaptable hardware module.

VIII. RESULTS AND DISCUSSION

Testing demonstrated the system’s ability to detect drowsiness with high precision, achieving consistent performance in controlled environments. The hardware module accurately identified eye closure patterns, triggering alerts promptly when thresholds were exceeded. Visual interfaces displayed real-time feedback, enhancing user interaction and awareness.

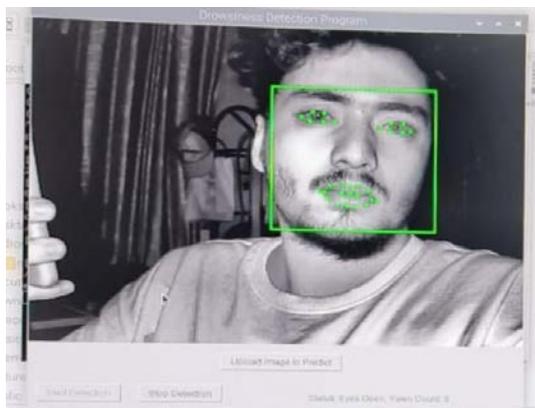


Figure 5: Interface: Open Eyes Detected

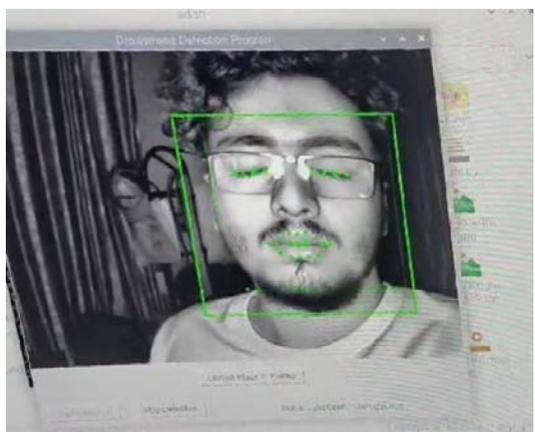


Figure6:Interface: Closed Eyes Detected

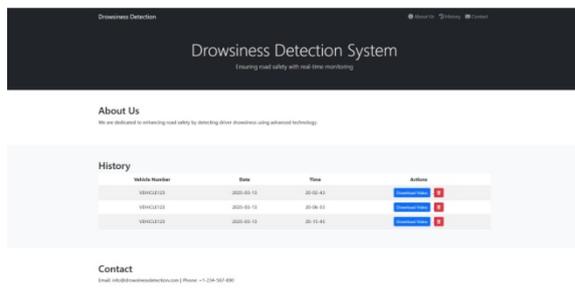


Figure7:Drowsiness Event Log

Challenges included occasional difficulties with glasses-wearing drivers, suggesting areas for algorithmic improvement. Environmental factors like lighting variations were mitigated through preprocessing, though further optimization could enhance robustness.

The results affirm the system’s potential as a reliable safety tool, with real-world testing planned to validate its broader efficacy.

IX. CONCLUSION

The drowsiness detection system using a hardware module presents an innovative, practical solution for enhancing driver safety. Its effective integration of affordable hardware and advanced vision techniques delivers real-time monitoring and alerting, significantly reducing the risk of fatigue-related accidents. The system’s design prioritizes accessibility and efficiency, making it a viable option for widespread adoption. This work lays a foundation for future advancements in safety technology, demonstrating the power of hardware-driven solutions in addressing critical challenges. Its success highlights the importance of interdisciplinary approaches, combining engineering and computer science to create impactful innovations for modern transportation needs.

X. FUTURE ENHANCEMENT

Future developments could enhance the system’s capabilities significantly. Integrating GPS and GSM modules would enable location tracking and remote alerts, improving response times in critical situations. Adding mask detection could address limitations with obstructed facial features, ensuring accuracy in diverse scenarios. Further enhancements might include AI-based predictive analytics to anticipate drowsiness before it occurs, leveraging historical data and driver behavior patterns. Incorporating

additional sensors, such as heart rate monitors, could provide a more comprehensive assessment, while connectivity with smart vehicle systems could enable automated interventions, advancing the system's role in intelligent transportation.

REFERENCES

- [1] R. Florez et al., "Real-Time Embedded System for Driver Drowsiness Detection Based on Visual Analysis of the Eyes and Mouth Using Convolutional Neural Network and Mouth Aspect Ratio," 2024.
- [2] F. Safarov et al., "Real-Time Deep Learning-Based Drowsiness Detection: Leveraging ComputerVision and Eye-Blink Analyses for Enhanced Road Safety," *Sensors*, vol. 23, no. 14, p. 6459, 2023.
- [3] Y. Li et al., "A CNN-Based Wearable System for Driver Drowsiness Detection," 2023.
- [4] M. Jain et al., "Real-Time Driver Drowsiness Detection using Computer Vision," 2021.
- [5] A. K. Biswal, "IoT-Based Smart Alert System for Drowsy Driver Detection," 2021.