

HAAR CASCADE ALGORITHM FOR MICROSLEEP DETECTION

Norkhushaini Awang ^{1*} and Ahmad Mirza Azhar ²

^{1*2}Faculty of Computer & Mathematical Sciences,
Universiti Teknologi MARA, 40450 Shah Alam Selangor,
Malaysia

^{1*}nor_awang@uitm.edu.my, ²2021470424@student.uitm.edu.my

ABSTRACT

Drowsy driving, particularly due to microsleep episodes, is a significant cause of traffic accidents, with existing solutions being often too costly or limited for widespread adoption. This project addresses this critical gap by developing a cost-effective, real-time Internet of Things (IoT)-based anti-microsleep alarm system. The system's development followed a four-stage process: Planning, Design, Development, and Evaluation. During the development phase, the system was built in Python using OpenCV and dlib for real-time facial analysis and the Haar Cascade algorithm for efficient facial feature detection. Key metrics like the Eye Aspect Ratio (EAR) and lip distance were monitored to identify signs of drowsiness and yawning. A comprehensive feedback loop was implemented using MQTT for communication between the Python backend and a Node-RED dashboard, with eSpeak and the Slack API providing aural and textual alerts. A finding from the evaluation, however, was a sensitivity to environmental factors as the distance between the driver and the camera increased, the system's accuracy in detecting drowsiness, yawning, and microsleep declined, leading to an increased risk of false negatives. Based on these results, future research should focus on enhancing the core algorithm to be more resilient to variable lighting and distance, thereby reducing false positives and negatives. Further work is also recommended to explore the system's integration with vehicle-specific infrastructure, develop more scalable data storage solutions, and conduct extensive long-term testing to validate its performance in diverse real-world driving conditions, which will pave the way for its commercial viability and broader adoption.

Keywords: Driver Safety, Haar Cascade Algorithm, Internet of Things (IoT), Microsleep Detection, Node-RED, Real-time Monitoring

Received for review: 17-07-2025; Accepted: 11-09-2025; Published: 01-10-2025
DOI: 10.24191/mjoc.v10i2.6845

1. Introduction

Driver fatigue, particularly in the form of microsleep episodes, remains a critical and persistent contributor to vehicular accidents. These brief lapses in consciousness severely impair a driver's ability to respond to critical sensory information, directly compromising safe vehicle operation (National Department of Transportation, 2023). The severity of this issue is discussed by the National Highway Traffic Safety Administration (NHTSA) estimates drowsy driving causes approximately 100,000 crashes annually in the United States, and the AAA Foundation for Traffic Safety reports that it accounts for 10% of all accidents (Foundation for Traffic Safety, 2018; National Department of Transportation, 2023). While existing interventions, such as rest breaks and caffeine, have proven to have limited effectiveness



This is an open access article under the CC BY-SA license
(<https://creativecommons.org/licenses/by-sa/3.0/>).

(Dawson et al., 2021), a gap persists in practical, widely adopted technological solutions. Current systems, including lane departure warnings and some advanced sleepiness detection technologies, are often restricted by their high cost and limited availability, hindering widespread implementation (Skorucak et al., 2020). This critical need for effective, affordable alternatives has prompted a surge in research focusing on Internet of Things (IoT)-based solutions.

Recent studies have explored the use of unobtrusive wearable technologies and physiological sensors to create real-time driver drowsiness monitoring systems. These devices track various parameters such as eyelid closure, head posture, and brain activity. Machine learning algorithms analyze anomalies in this sensor data to detect microsleep episodes and the early onset of sleepiness (Jabbar et al., 2020; National Department of Transportation, 2023). A key advancement is the use of IoT connectivity, which facilitates the rapid transmission of alerts to the driver upon detecting a problematic state (Pauli et al., 2021). The integration of strain sensors, for instance, has shown promise in identifying microsleep events through abrupt reductions in muscle tone. In a simulated driving environment, an alarm system successfully detected these episodes and triggered alerts within an average of 0.96 seconds of onset. As sensor technology continues to advance and 5G network infrastructure expands, the commercialization of these rapid-response, IoT-enabled devices becomes increasingly feasible. Ultimately, the widespread adoption of these systems holds significant potential to mitigate the risks associated with driver fatigue and substantially decrease the occurrence of traffic accidents.

2. Literature Review

Microsleep is a condition characterized by brief periods of unconsciousness, typically lasting between 1 and 15 seconds, and is often caused by fatigue from insufficient rest (Zaleha et al., 2021). Factors such as long-distance driving, certain health conditions like Obstructive Sleep Apnea (OSA), and specific road or weather conditions can contribute to its occurrence (Biswal et al., 2021; Pham et al., 2023). Microsleep events, which are often accompanied by abrupt reductions in muscle tone, can be identified through strain sensor readings. For instance, an IoT-based alarm system was successfully tested in a driving simulator, where it detected microsleep episodes and triggered alerts within an average of 0.96 seconds of their onset. The integration of IoT technologies has accelerated the development of these advanced systems. Wearable physiological sensors, coupled with machine learning algorithms, provide a more comprehensive understanding of driver alertness by analyzing multiple parameters simultaneously, including eye movements and heart rate variations (Sudarshan et al., 2023). The ability of these systems to process data in real time and provide prompt interventions is crucial for mitigating accidents caused by driver drowsiness. Ongoing advancements in sensor technology and the expanding 5G network infrastructure could further facilitate the commercialization of these solutions, making them more accessible and effective in reducing traffic accidents related to driver fatigue.

The proposed project addresses the issue of driver microsleep, a factor in traffic accidents. Existing literature highlights the severity of this problem, with drowsy driving being a major contributor to a substantial number of accidents annually (Foundation for Traffic Safety, 2018; National Department of Transportation, 2023). Previous research has explored the characteristics of microsleep, including its causes and categories (Zaleha et al., 2021; Skorucak et al., 2020; Pham et al., 2023; Biswal et al., 2021; Jabbar et al., 2020; Sumitha & Subha, 2020). Past research work has underscored the need for practical and effective solutions to mitigate the risks associated with microsleep. Past studies have been defining the problem and its physiological underpinnings, they often fall short in providing a real-time, integrated, and scalable solution. Many existing approaches rely on laboratory settings which are limited by computational constraints, making them impractical for widespread vehicular integration. The current project addresses this gap by developing a real-time, Internet of Things (IoT)-based system to detect microsleep events. This approach is a

combination of techniques for feature extraction, interconnected technologies for data processing and alert generation. By integrating these elements, the project offers a deployable and immediate solution compared to the primarily theoretical and laboratory-based contributions. This comparison highlights the project's contribution to the field not just in identifying the problem, but in providing a practical, and technologically advanced solution that can be integrated into real-world applications.

The implementation of the Haar Cascade algorithm is an element in the real-time detection of facial features, specifically eye closure. This algorithm is well-regarded for its computational efficiency, which makes it highly suitable for processing continuous video streams and promptly identifying the subtle changes in eye state that are indicative of microsleep episodes (Balcero-Posada et al., 2022). The proposed system monitors and analyzes data from a camera module. Its primary focus is on two key metrics which are EAR and the lip distance. By continuously evaluating these parameters, the system can discern specific patterns associated with the onset and occurrence of microsleep. This approach is predicated on the physiological changes that occur during a microsleep event, which are reliably captured through these quantifiable facial metrics. The discussions presented above are summarized in Table 1, which provides a comprehensive overview of the literature review. This table systematically compares key findings and contributions from various studies, highlighting their direct relevance to the current research on microsleep detection.

Table 1 Literature Review Comparison Table

Issues	Author(s)	Key Findings/ Contributions	Relevance to the Study
Microsleep Characteristics	Zaleha et al. (2021); Skorucak et al. (2020); Pham et al. (2023); Biswal et al. (2021); Jabbar et al. (2020); Sumitha and Subha (2020)	Defines microsleep, its causes, and categories.	Provides foundational understanding of the problem being addressed.
IoT Integration in Microsleep Detection	Pauli et al. (2021); National Department of Transportation (2023); Balcero-Posada et al. (2022)	Explores the use of IoT for enhanced microsleep detection and real-time monitoring.	Highlights the technological context and potential solutions.
IoT and Vehicle Safety	Sudarshan et al. (2023)	Discusses the role of IoT in improving vehicle safety through fatigue detection systems.	Emphasizes the importance of IoT in the proposed solution.
Microsleep Detection Technologies	Balcero-Posada et al. (2022); Sudarshan et al. (2023)	Reviews sensors, hardware tools, and algorithms used for microsleep detection.	Informs the selection of appropriate technologies for the project.
Impact of Drowsy Driving and Need for Solutions	Foundation for Traffic Safety (2018); Dawson et al. (2021)	Highlights the dangers of drowsy driving and the necessity for effective detection and prevention systems.	Justifies the significance and purpose of the research.

In conclusion, the current research landscape provides a foundation for the development of driver fatigue detection systems. While prior studies have defined the nature of microsleep, its causes, and the associated risks of drowsy driving, a critical gap remains in

the implementation of practical and integrated solutions. The literature highlights the potential of Internet of Things (IoT) technologies and various detection algorithms, but a comprehensive, real-time system that combines these elements for widespread application has not yet been fully realized. Therefore, this project addresses a vital need by synthesizing existing knowledge to create an innovative and effective system. This not only builds upon the foundational understanding of microsleep but also offers a tangible, technologically advanced solution to a well-documented and urgent safety problem.

3. Research Methodology

Figure 1 shows a flow chart, outlines the methodology for developing the microsleep detection system. The process is divided into four main stages which are planning, design, development, and evaluation. The planning phase establishes the project's foundation by defining the problem statement, studying related works, and setting clear objectives and a project scope. This initial work provides the necessary context and direction for the subsequent stages. Following this, the design phase focuses on creating the system's architecture. It involves selecting the appropriate hardware and software, designing the system's schematics, and creating a user-friendly interface. The development phase then moves into the hands-on creation of the system. This stage includes writing and compiling the code, building the system, establishing a database, and developing a dashboard with notifications. Finally, the evaluation phase rigorously tests the system's performance. In this phase, key parameters are identified, and the system's accuracy in detecting drowsiness, yawning, and microsleep is tested and analyzed to ensure its overall effectiveness.

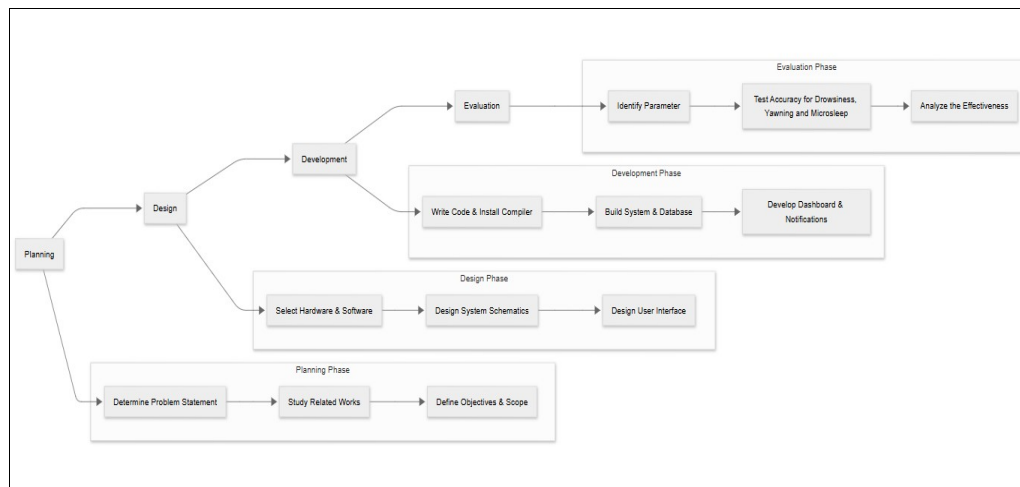


Figure 1. System Development Flowchart

3.1 Haar Cascade Algorithm

The Haar Cascade algorithm, introduced by Viola and Jones in 2001, stands as a foundational machine learning-based approach for real-time object detection in computer vision (Bade and Sivaraja, 2020). This technique employs a cascade function, a series of stages containing classifiers that identify specific Haar-like features. These features are essentially rectangular filters that measure the difference in intensity between adjacent image regions. The algorithm's computational efficiency, crucial for its real-time performance, is achieved through the use of an integral image to rapidly calculate these features (Bade and Sivaraja, 2020).

The classifier is trained using the AdaBoost algorithm, which iteratively selects and combines the most discriminative features from a vast pool of positive and negative training

images to form a robust classification model. Once trained, this model can efficiently scan images at various scales and positions to detect the target object, making it particularly well-suited for real-time applications like face detection (Jain et al., 2018). Although deep learning methods have since emerged with superior accuracy and versatility, the Haar Cascade algorithm's enduring relevance stems from its speed and computational simplicity. In the context of developing IoT applications for driver microsleep detection, this efficiency is a critical advantage. The algorithm enables the system to continuously monitor a driver's face and eye movements, providing a reliable and low-latency method for detecting signs of drowsiness and triggering timely alerts to prevent accidents. This makes the Haar Cascade algorithm an invaluable component, ensuring effective operation within the constrained computing environments often associated with IoT devices. Its application here mirrors broader trends in machine learning where efficient algorithms are vital for practical solutions, such as in crop yield prediction (Fashoto et al., 2021) and regression model analysis (Adewoye et al., 2021).

3.2 Project Development

The project development sequence diagram illustrates the operational workflow of the proposed IoT-based microsleep alarm system, delineating the interactions between the driver and the system's components, and emphasizing the temporal sequence of actions and data exchanges critical for the detection and mitigation of microsleep events. As shown in Figure 2, the process initiates with the Driver engaging with the Headband Sensors through device usage. These sensors, integral to the system's functionality, are designed to capture and transmit pertinent biometric data indicative of driver drowsiness. The data includes parameters such as eye and eyelid movements.

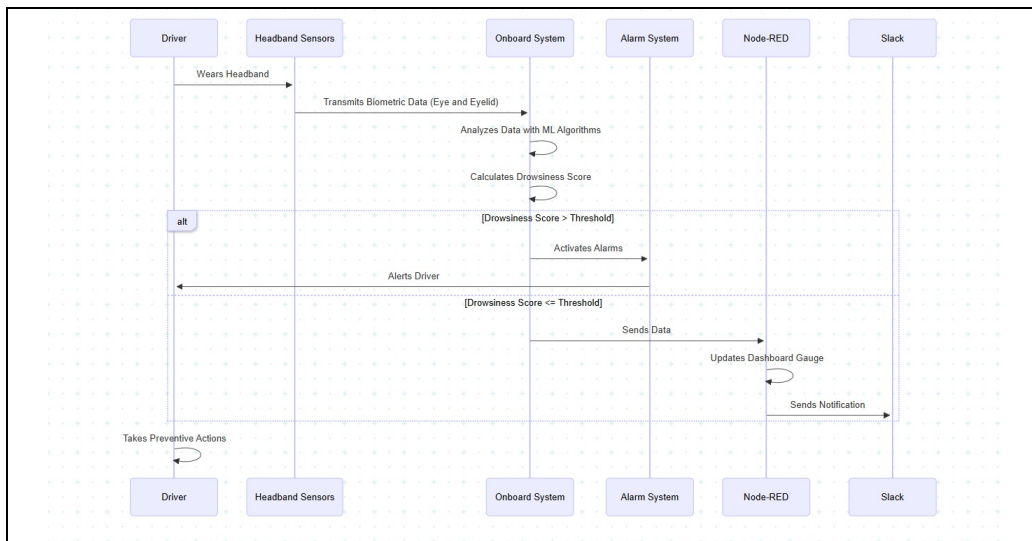


Figure 2. Project Development Sequence Diagram

Upon acquisition by the Headband Sensors, the biometric data is continuously transmitted to the Onboard System. This system, characterized as a processing and alarm device mounted in the car, constitutes the central processing unit, responsible for the real-time analysis of the incoming sensor data. The Onboard System employs machine learning algorithms to analyze the received biometric data, with the objective of deciphering the driver's state of alertness. A core function of this analysis involves the computation of a Drowsiness Score, a quantitative metric derived from indicators such as decreased heart rate variability, changes in prefrontal cortex activity, increased eyelid closure time, and slowed blink rates. The subsequent sequence of operations is conditional, predicated on the value of the calculated Drowsiness Score. If the Drowsiness Score surpasses a predefined threshold,

the Onboard System triggers the Alarm System. This action is designed to provide multi-modal alerts, incorporating auditory, visual, and tactile cues, to the Driver, thereby prompting immediate corrective actions.

On the other hand, if the Drowsiness Score remains below the established threshold, the Onboard System transmits the processed data to Node-RED. Node-RED, an IoT application platform, is then responsible for updating the dashboard gauge to provide a visual representation of the driver's drowsiness level. Additionally, Node-RED facilitates the transmission of notifications to Slack, a communication platform, via a webhook, enabling the logging of driver alertness status and the potential alerting of remote stakeholders. The concluding interaction depicted in the sequence diagram involves the Driver's potential response, which may include actions such as initiating a rest stop or modifying driving behavior, based on the alerts and information provided by the system. In summary, the sequence diagram illustrates the system's operational dynamics, from the initial capture of biometric data to the provision of alerts and information to the driver.

4. Result

Based on the provided class diagram in Figure 3, the NodeRED class functions as a pivotal interface for data visualization and external communication within the system. It encapsulates key attributes such as a `dashboardGauge`, representing a user interface element for visual data presentation, and a `webhookURL`, facilitating outgoing notifications to external services. Its operational capabilities are defined by methods including `updateDashboard(float)`, responsible for refreshing the visual display with relevant data, and `sendNotification(String, String)`, enabling alerts or information dissemination. Significantly, NodeRED's `receiveData(OnboardSystemData)` method allows it to acquire processed information, specifically `OnboardSystemData` instances containing drowsiness scores and sensor readings, directly from the OnboardSystem. This data is then displayed to the Driver, making NodeRED the primary channel for conveying real-time system insights and alerts to the user, thereby bridging the analytical backend with actionable driver awareness.

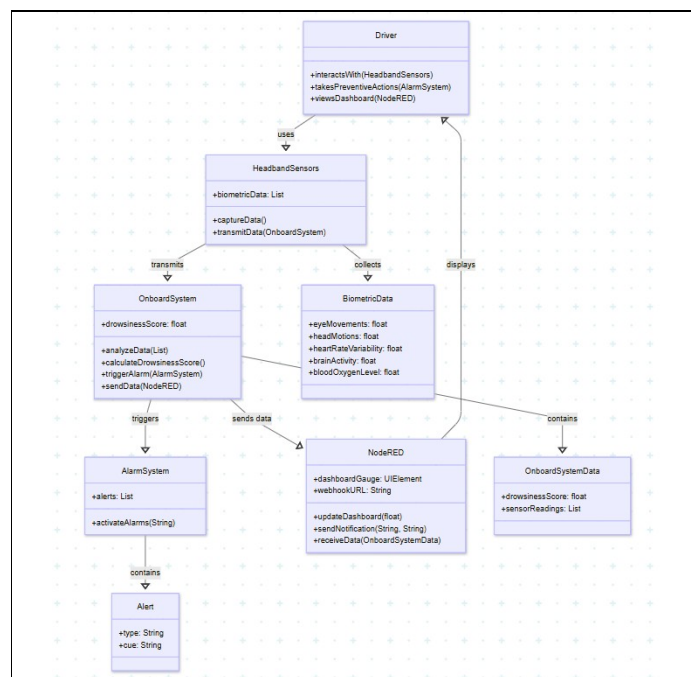


Figure 3. Node Red Class Diagram

The deployment of eSpeak on Windows systems typically involved a structured, sequential installation and configuration process. Initially, users obtained the eSpeak executable from its official repository or other verified software distribution channels. Subsequently, the installation routine was executed, guiding users through on-screen prompts to complete the setup. During this phase, customization options, such as specifying the installation directory and selecting supplementary components, were available. Post-installation, eSpeak could be utilized via the command-line interface or integrated into software applications using programming languages such as Python or C++. The command-line interface facilitated text-to-speech conversion through specific commands, allowing users to define speech attributes including voice type, pitch, and speaking rate. For programmatic integration, developers leveraged eSpeak's Application Programming Interface (API) or dedicated wrapper libraries to seamlessly embed text-to-speech capabilities into their applications. Furthermore, post-installation customization options enabled users to refine voice attributes, language preferences, and pronunciation rules, thereby personalizing the voice synthesis output to meet specific requirements. This comprehensive process, encompassing a straightforward initial setup followed by detailed customization and integration steps, enabled effective utilization of its text-to-speech functionalities, as illustrated in Figure 4.

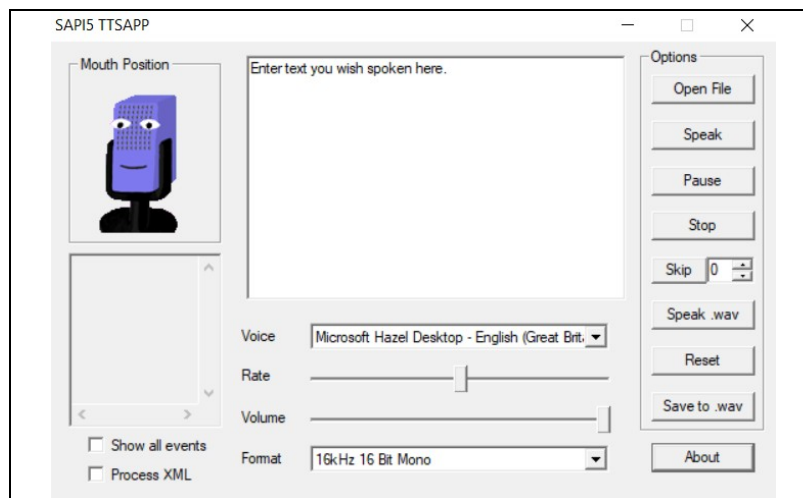


Figure 4. Text-to-Speech Conversion

This project's implementation focuses on leveraging the capabilities of the OpenCV and dlib libraries within a Python environment to establish a robust system for real-time facial analysis and weariness detection. The foundational step involved importing essential modules for various operational aspects. These included `scipy.spatial.distance` for geometric calculations, `imutils` for streamlined image processing, and `numpy` for efficient numerical operations. Concurrent execution was enabled through `threading`, while `argparse` facilitated command-line argument handling. Core functionalities were supported by `dlib` for sophisticated facial landmark detection and `cv2` in OpenCV for comprehensive image and video processing. Furthermore, `pyttsx3` was integrated for text-to-speech (TTS) capabilities, complemented by modules for data serialization to manage data structures effectively. Communication protocols were established using `paho.mqtt.client` for MQTT messaging, and `os` was utilized for interacting with the operating system, as illustrated in Figure 5. This integration of libraries and modules underpins the system's ability to perform sophisticated real-time facial analysis for detecting driver fatigue.



Figure 5. Facial Analysis

The system's implementation for weariness detection and real-time facial analysis is developed in Python, leveraging robust and widely used libraries such as OpenCV and dlib. A critical initial step is the importation of modules, including `scipy.spatial.distance` and `numpy`, which are essential for the mathematical computation of fatigue metrics like the Eye Aspect Ratio (EAR) and lip distance. The `imutils` library is strategically employed to optimize the video processing pipeline by enabling efficient access to video streams, frame resizing, and grayscale conversion. Facial detection and the precise extraction of 68 facial landmarks are accomplished using the `dlib` package with a pre-trained model. This is a fundamental component, as the accuracy of EAR and lip distance calculations is directly dependent on the precise identification of these landmarks. As depicted in Figure 4, within the main execution loop, video frames are continuously read via `cv2` and then subjected to `dlib`'s face detection and shape prediction techniques.

Concurrently, the system actively monitors these metrics for indicators of driver tiredness or yawning, triggering an immediate and critical alarm upon detection. This real-time analysis is a feature designed to provide a response to potential safety hazards. For external data analysis and system integration, the MQTT client publishes the computed EAR and lip distance values to designated topics in JSON format. This is a step for enabling the system to communicate with other devices or dashboards, as shown in Figure 6. Visual feedback is a continuous and integral aspect of the system's operation. OpenCV (`cv2`) displays processed frames, which are text annotations indicating the alert status, EAR, and lip distance. This visual information provides feedback to the driver and aids in system debugging. Furthermore, the system is designed with a provision for a smooth program exit, ensuring proper resource cleanup by closing active windows, disconnecting the MQTT client, and terminating the video stream, which is vital for facilitating seamless API integrations and broader data exchange.

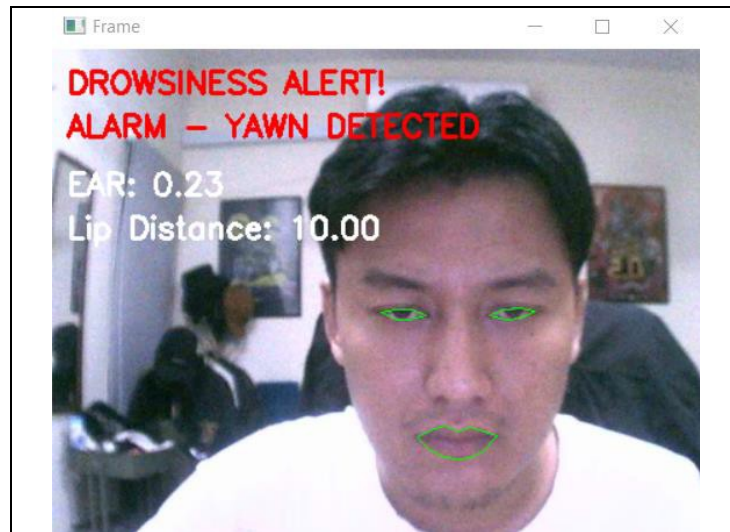


Figure 6. Real-time Fatigue Detection

A primary challenge in developing a real-time driver fatigue detection system is the occurrence of both false positives and false negatives. As seen in Table 2, the system's performance is dependent on environmental factors, particularly the distance between the driver and the camera. This variability can lead to issues, such as false positives and false negatives. False positives, where the system incorrectly identifies a non-drowsy state as fatigue, often arise from momentary facial occlusions or natural actions like sneezing. These unwarranted alerts can cause driver frustration and erode confidence in the system's reliability, potentially leading to the alerts being ignored. Conversely, false negatives, which occur when genuine signs of drowsiness are missed, present a critical safety risk. This failure to detect fatigue can be attributed to suboptimal conditions such as poor lighting, head position changes, or a greater distance from the camera, which can diminish the accuracy of facial landmark detection. The system's core metrics, such as EAR and lip distance, may not consistently fall below the necessary thresholds under these circumstances. To address these issues, mitigation strategies are essential. Enhancing the algorithm to differentiate between brief eye closures and genuine fatigue patterns is crucial for reducing false positives. Similarly, developers must focus on making the system more resilient to variable lighting and distance to minimize false negatives and ensure its primary objective of improving driver safety is met.

Table 2 Performance Analysis by Distance

Distance (cm)	Effectiveness	Accuracy Drowsiness	Accuracy Yawning	Accuracy Microsleep	Explanation
30-40cm	Excellent	85-90%	85-90%	90%+	Stable feature detection due to ideal lighting and landmark visibility
50-60cm	Good	75-85%	75-85%	85-90%	Consistent feature detection under adequate lighting conditions

Distance (cm)	Effectiveness	Accuracy Drowsiness	Accuracy Yawning	Accuracy Microsleep	Explanation
70-90cm	Fair	60-70%	60-70%	70-80%	Unstable feature detection resulting from smaller facial landmarks
>100cm	Poor	<50%	<50%	<60%	Feature detection failure due to loss of landmarks and reduced face bounding box accuracy

The developed system for real-time fatigue detection emphasized the integration of robust technologies and adherence to coding standards. It primarily leveraged OpenCV and dlib for facial recognition and landmark identification, utilizing EAR and lip distance calculations as key metrics for monitoring driver gaping and fatigue in real-time video data. The implementation featured a comprehensive feedback loop, employing the MQTT protocol for seamless communication between the Python backend and Node-RED, complemented by the Slack API for notifications, MongoDB for data storage, and eSpeak for aural alerts. This successfully demonstrated real-time fatigue detection capabilities, while also highlighting the need for improved data aggregation and filtering methods to address encountered challenges and guide future enhancements of the technology.

5. Conclusion

In conclusion, this project successfully developed and validated a real-time driver fatigue detection system using a combination of the Haar Cascade algorithm, OpenCV, and dlib within a Python environment. The system's effectiveness was demonstrated through its ability to accurately monitor key metrics such as EAR and lip distance, and to trigger timely alerts via an IoT-based alarm system and notifications to a Slack channel. While the current system shows significant promise, future research should focus on mitigating key limitations, particularly the occurrence of false positives and false negatives, which are highly sensitive to environmental factors like lighting and camera distance. Addressing these challenges will require enhancing the core algorithm with more advanced machine learning models that are more resilient to these variables. Furthermore, future work could explore integrating the system with vehicle-specific infrastructure, developing a more scalable and robust data storage solution beyond MongoDB, and conducting extensive long-term testing to validate its performance in diverse real-world driving conditions, paving the way for its commercial viability and widespread adoption.

Acknowledgement

The authors wish to express their sincere gratitude to the Fakulti Sains Komputer dan Matematik, Universiti Teknologi MARA (UiTM) Shah Alam, for their invaluable support and provision of resources throughout this research. Special appreciation is also extended to the RIG Cybersecurity and Digital Forensic for their consistent support and contributions to this study.

Funding

The author(s) received no specific funding for this work.

Author Contribution

In this study, Author1 contributions encompassed the comprehensive literature review, the methodological framework, and the overarching supervision of the article's composition. Author2's primary role was the execution of fieldwork, followed by the analysis and subsequent presentation of the findings within the results and discussion sections.

Conflict of Interest

It is declared that the authors have no conflicting interests.

References

- Adewoye, K. B., Rafiu, A. B., Aminu, T. F. and Onikola, I. O. (2021) 'Investigating the Impact of Multicollinearity on Linear Regression Estimates', *Malaysian Journal of Computing*, 6(1), p. 698.
- Agar, G., Oliver, C., Spiller, J., & Richards, C. (2023). The developmental trajectory of sleep in children with Smith-Magenis syndrome compared to typically developing peers: a 3-year follow-up study. *Sleep Advances: A Journal of the Sleep Research Society*, 4(1), <https://doi.org/10.1093/sleepadvances/zpad034>
- Bade, A. and Sivaraja, T. (2020) 'Evolution of Face Detection Techniques in Digital Images', *ASM Science Journal*, 26.
- Balcero-Posada, P. E., Calderon-Montealegre, A. J., Navarro-Beltran, J. C., Marentes, L. A., Carrillo-Pinzon, J. Y., & Herrera-Quintero, L. F. (2022). A practical approach based on IoT, Video, Audio, and ITS for Micro-Sleep detection in public transportation systems. In *2022 IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (pp. 1-6). IEEE. <https://doi.org/10.1109/ICVES56941.2022.9987101>
- Biswal, A. K., Singh, D., Pattanayak, B. K., Samanta, D., & Yang, M. H. (2021). IoT-based smart alert system for drowsy driver detection. *Wireless Communications and Mobile Computing*, 2021. <https://doi.org/10.1155/2021/6627217>
- Dawson, D., Sprajcer, M., & Thomas, M. (2021). How much sleep do you need? A comprehensive review of fatigue related impairment and the capacity to work or drive safely. *Accident Analysis and Prevention*, 151, Article 105955. <https://doi.org/10.1016/j.aap.2020.105955>
- Fashoto, S., Mbunge, E., Ogunleye, G. and Van den Burg, J. (2021) 'Implementation of Machine Learning for Predicting Maize Crop Yields Using Multiple Linear Regression and Backward Elimination', *Malaysian Journal of Computing*, 6(1), p. 679.
- Foundation for Traffic Safety (2018) *Hit-and-Run Crashes: Prevalence, Contributing Factors and Countermeasures*.

- Jabbar, R., Shinoy, M., Kharbeche, M., Al-Khalifa, K., Krichen, M. and Barkaoui, K. (2020) ‘Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application’, *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT 2020*, pp. 237–242.
- Jain, U., Choudhary, K., Gupta, S., & Privadarsini, M. J. P. (2018). Analysis of Face Detection and Recognition Algorithms Using Viola Jones Algorithm with PCA and LDA. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 945-950). IEEE. <https://doi.org/10.1109/ICOEI.2018.8553811>
- National Department of Transportation, U. (2023) *State Traffic Data: 2021*.
- Pauli, M. P., Pohl, C., & Golz, M. (2021). Balanced Leave-One-Subject-Out Cross-Validation for Microsleep Classification. *Current Directions in Biomedical Engineering*, 7(2), 147–150. <https://doi.org/10.1515/cdbme-2021-2038>
- Pham, N., Dinh, Tuan, Kim, T., Raghebi, Z., Bui, N., Truong, H., Nguyen, T., Banaei-Kashani, F., Halbower, A., Dinh, Thang, Nguyen, P. and Vu, T. (2023) ‘Detection of Microsleep Events with a Behind-the-Ear Wearable System’, *IEEE Transactions on Mobile Computing*, 22(2), pp. 841–857.
- Radhika, E. G., Bharath, S., Ezhilarasan, S., & Shankar, M. (2022). Container based Driver Fatigue System using IoT. *7th International Conference on Communication and Electronics Systems, ICCES 2022 - Proceedings*, 413–420. <https://doi.org/10.1109/ICCES54183.2022.9835937>
- Skorucak, J., Hertig-Godeschalk, A., Achermann, P., Mathis, J., & Schreier, D. R. (2020). Automatically Detected Microsleep Episodes in the Fitness-to-Drive Assessment. *Frontiers in Neuroscience*, 14, 8. <https://doi.org/10.3389/fnins.2020.00008>
- Sudarshan, P., Bhardwaj, V., & Virender. (2023). Real-time Driver Drowsiness Detection and Assistance System using Machine Learning and IoT. In *2023 8th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1128-1132). IEEE. <https://doi.org/10.1109/ICCES57224.2023.10192831>
- Sumitha A and Subha R (2020) ‘Micro sleep Detection Technique’, *International Journal of Innovative Research in Technology (IJIRT)*, 6(11), pp. 242–248.
- Zaleha, S. H., Wahab, N. H. A., Ithnin, N., Ahmad, J., Zakaria, N. H., Okereke, C. and Huda, A. K. N. (2021) ‘Microsleep Accident Prevention for SMART Vehicle via Image Processing Integrated with Artificial Intelligent’, *Journal of Physics: Conference Series*, 2129(1), pp. 1–5.