Driver Drowsiness Detection System using Facial Analysis

Francisco Lozano
Depaul University
CSC 481 Section 701
Flozano2@depaul.edu

Abstract

Drowsiness is a significant factor in road accidents, especially among long-distance drivers, making effective detection systems essential for enhancing road safety. This paper proposes a computationally efficient, image-based method for detecting driver drowsiness using single-frame inputs rather than continuous video. The approach leverages eyelid coverage as the primary indicator, calculated through eye landmarks and contour analysis, which allows for real-time processing with minimal computational resources. The study utilizes the Driver Drowsiness Dataset (DDD) and applies preprocessing techniques, such as histogram equalization and noise-based data augmentation, to enhance eye region visibility. Two algorithms were compared: one based on the Eye Aspect Ratio (EAR) and the other on eyelid coverage. The eyelid coverage algorithm achieved higher accuracy (65%) and balanced precision and recall for detecting both "Drowsy" and "Non-Drowsy" states. While the results are promising, further refinement, including advanced feature extraction and threshold calibration, could improve the system's sensitivity to subtle drowsiness cues. Future work will explore the integration of additional facial landmarks and the application of machine learning models, such as Support Vector Machines (SVMs), for a more robust and accessible solution in driver monitoring systems.

1. Introduction

Driver drowsiness is a major risk factor in road safety, contributing to a significant number of accidents worldwide. Fatigue-related incidents, particularly affecting truck drivers on long night shifts and long-distance bus operators, lead to numerous injuries and fatalities each year. Effective drowsiness detection systems have therefore become essential to save lives by identifying early signs of fatigue and alerting drivers in real-time. Current detection methods often rely on continuous video streaming and computationally intensive machine learning models, which may be impractical in resource-limited settings. Common approaches fall into three categories: vehicle-based methods, which monitor driving metrics like steering and lane positioning; behavior-based methods, which assess visual cues such as eye closure; and physiological methods, which use intrusive sensors to track signals like heart rate and brain activity.

This study seeks to address limitations in existing behavior-based systems by exploring efficient, image-based drowsiness detection techniques. We experimented with a system using image processing and facial landmark detection based on single image inputs. Our system focuses on eyelid coverage analysis through eye landmarks and contour detection, aiming to achieve accurate drowsiness assessment with minimal computational demands.

2. Background

Numerous studies have aimed to develop effective driver drowsiness detection systems, primarily using video-based monitoring methods. Traditional approaches focus on continuous tracking of facial cues, such as eye closure duration, blink frequency, or head posture, to assess driver alertness. For example, Kumar et al. (2018) [5] investigated a system that detects drowsiness by tracking eye and head movements in real-time video feeds. While effective, such systems require high computational power to process video frames continuously, making them impractical for many real-world applications where resources may be limited.

As deep learning techniques have advanced, recent research has explored the potential of deep neural networks (DNNs) to improve drowsiness detection. Nasri et al. (2022) [6] used DNNs to classify drowsiness based on complex facial features, achieving promising results. These DNN-based systems can capture subtle signs of fatigue; however, they still require substantial computational resources and typically need high-end hardware to run efficiently, which limits their accessibility and scalability for in-vehicle applications.

To address these challenges, researchers are investigating image-based approaches that reduce computational demand. By analyzing drowsiness from single image frames instead of continuous video, such systems aim to balance efficiency and accuracy. For instance, image-based techniques can analyze key facial landmarks, particularly around the eyes, to estimate eyelid coverage—a critical indicator of drowsiness—using minimal data input. This paper builds on these advances by exploring an image-based approach that leverages eyelid and eye contour analysis to detect drowsiness in a computationally efficient manner, making it suitable for resource-constrained environments and potentially expanding the accessibility of drowsiness detection technology.

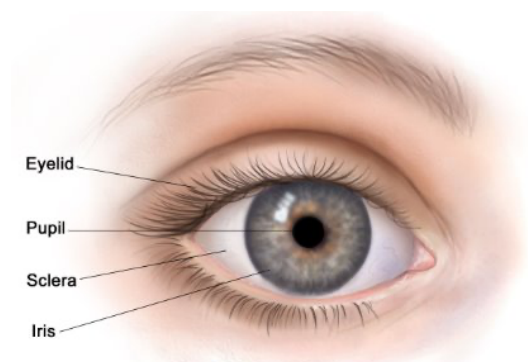

Figure 1. Eye diagram

In this project, we focus on analyzing key eye features such as the Pupil, Sclera, Iris, and Eyelid. We assume that when a person is drowsy, their eyelids will cover more of their pupil, sclera, and iris, leading to what is commonly referred to as "droopy" eyes. By analyzing the relationship between eyelid coverage and these eye features, we aim to detect drowsiness based on eye movements.

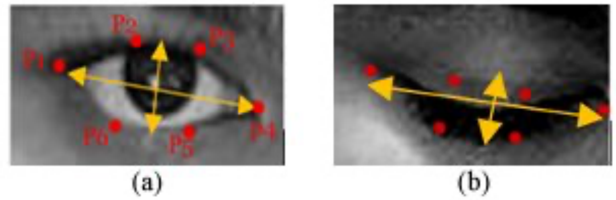

Figure 2. Showing EAR on an eye [11]

Additionally, the Eye Aspect Ratio (EAR) [11], which quantifies the openness of the eye, will be used as a baseline for comparison. A fully closed eye corresponds to an EAR of 0, making it a useful indicator for assessing the degree of eyelid coverage. However, our approach moves beyond EAR to incorporate more detailed contour-based analysis for better accuracy in detecting varying levels of drowsiness. This work seeks to offer a more computationally efficient method for driver drowsiness detection that could be used in real-time, even in resource-limited environments like in-vehicle systems.

3. Methods

The proposed drowsiness detection algorithm (Algorithm 2) uses single-frame images to assess driver drowsiness. The algorithm preprocesses the image, extracts facial features, calculates eyelid coverage, and evaluates drowsiness based on a threshold.

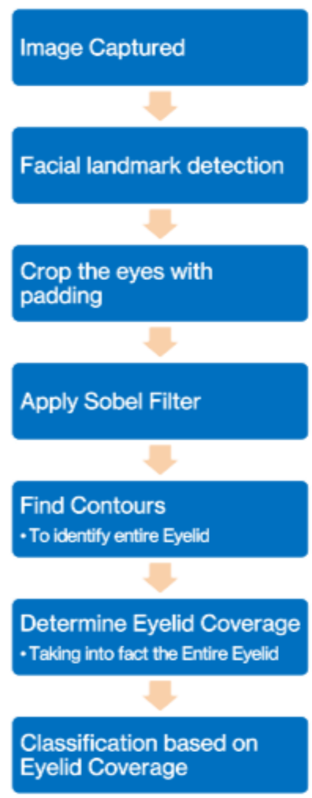

Figure 3. Algorithm 2 Pipeline

3.1 Data Preprocessing

The dataset used for this study is the Driver Drowsiness Dataset (DDD), a collection of 41,790 RGB images that includes cropped face images of drivers in two categories: “Drowsy” and “Non-Drowsy” (Nasri, 2022) [6]. The dataset [12], available on Kaggle, was constructed by extracting individual frames from the Real-Life Drowsiness Dataset videos using VLC software, followed by region-of-interest (ROI) extraction with the Viola-Jones algorithm to focus on drivers’ faces. Each image in the dataset has dimensions of 227 x 227 pixels, which standardizes input dimensions for streamlined processing.

To ensure consistency and improve drowsiness detection, the preprocessing pipeline isolates the eye region from each image using a combination of face detection and facial landmark prediction. This is essential to accurately locate and crop eye regions, which are then padded to further standardize dimensions across the dataset. This cropping and padding process enables consistent eyelid coverage calculations, a crucial aspect of detecting drowsiness accurately without the variability that could arise from different face sizes or orientations.

Beyond standard cropping, histogram equalization was applied to enhance contrast in the eye regions, highlighting critical features such as eyelid contours. Initial tests with edge filters were conducted, but these produced suboptimal results, leading to the decision to use histogram equalization as the preferred contrast enhancement technique. Additionally, data augmentation was performed to increase the dataset size to 61.8k images, incorporating variations with Gaussian and Salt & Pepper noise [13]. These augmentations improve model robustness by providing diverse image samples, simulating different lighting conditions, and enhancing the system's generalization capabilities.

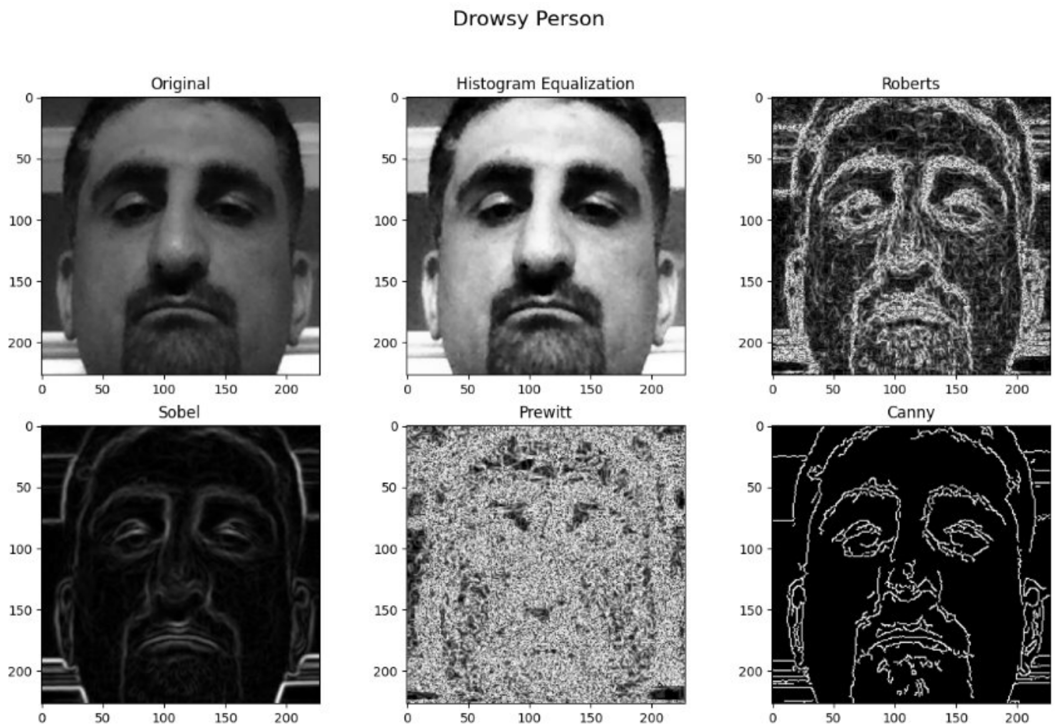

Figure 4. Filters that we experimented with

This preprocessing approach allows the system to operate with consistent and enhanced image quality, which is essential for accurate feature extraction in the drowsiness detection algorithm. By focusing on the eye regions and using augmented images, the preprocessing pipeline supports the objective of real-time, resource-efficient drowsiness detection.

3.2 Feature Extraction

The core of our drowsiness detection algorithm relies on accurately extracting and analyzing eye region

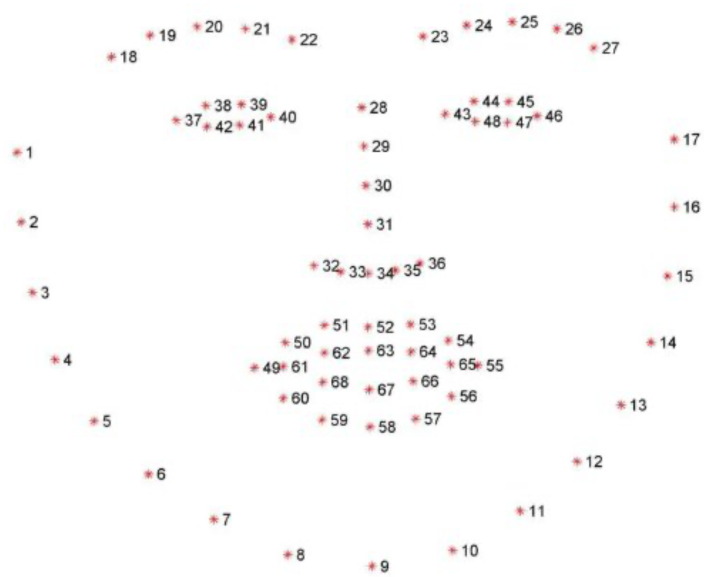

Figure 5. Dlib 68 facial landmark coordinates visualization [4]

features to determine eyelid coverage, a key indicator of drowsiness. To achieve this, we employ an ensemble of regression trees algorithm, specifically designed for rapid facial landmark detection, as described by Kazemi and Sullivan (2014) [4]. This method enables us to quickly identify specific points on the face, particularly around the eyes, ensuring precise measurements that adapt to the dimensions of each driver's face.

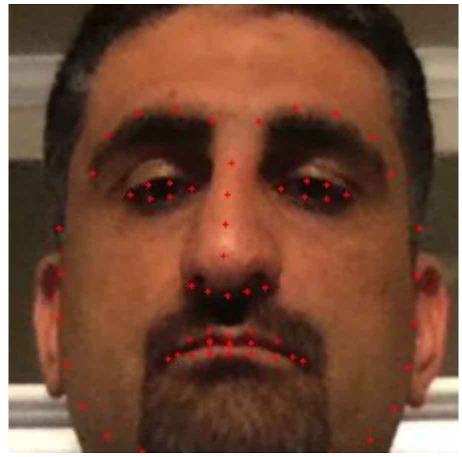

Figure 6. Facial Landmarks on "Drowsy" person

Using python's dlib library's pre-trained face detector, an implementation of the regression trees algorithm, we detect facial landmarks on each preprocessed image. For our application, the focus is on a subset of landmarks around the eyes. Because of this we only used the coordinates associated with the eyes.

By applying this pipeline, we efficiently locate the landmarks necessary for monitoring eye behavior. Extracted landmarks around the eyes serve as the basis for computing eyelid coverage in real-time, allowing for efficient detection without the need for continuous video streaming. The facial landmark predictor, combined with the isolated eye landmarks, serves as the backbone for our feature extraction, enabling accurate and low-latency drowsiness detection. This approach is ideal for real-time applications, as it reduces computational demands and focuses only on critical eye-region data.

3.3 Eyelid Coverage Calculation

The detection of drowsiness relies on calculating eyelid coverage, which is determined by the position of the eyelids relative to the eye's vertical landmarks. By measuring the distance between the upper and lower eyelids, we estimate the extent to which the eyelids are covering the pupil. This is done using specific eye landmarks and eyelid contours, adjusting them to match the coordinates of the original image to ensure consistency in measurements.

Steps for Eyelid Coverage Calculation

1. **Vertical Distance Calculation**: We first measure the vertical distance between the upper and lower eye landmarks (points A and B), identified through facial landmark detection, to calculate the distance between open eye regions. This provides a baseline for determining the degree of eye openness or closure. We called this pupil coverage.
2. **Contour Adjustment**: Contours of the eyelids are identified using a Sobel edge detection filter and then adjusted to align with the original image's coordinate system. This step is crucial for maintaining consistency in coverage measurements, especially as eye images are cropped with padding for easier processing.
3. **Coverage Ratio Calculation**: The coverage ratio is calculated by comparing the distance between eyelid contours to eye height (pupil coverage) in a normalized range of 0 to 1. This ratio

represents the eyelid coverage level, with higher values indicating closed eyelids and a potential drowsiness state.

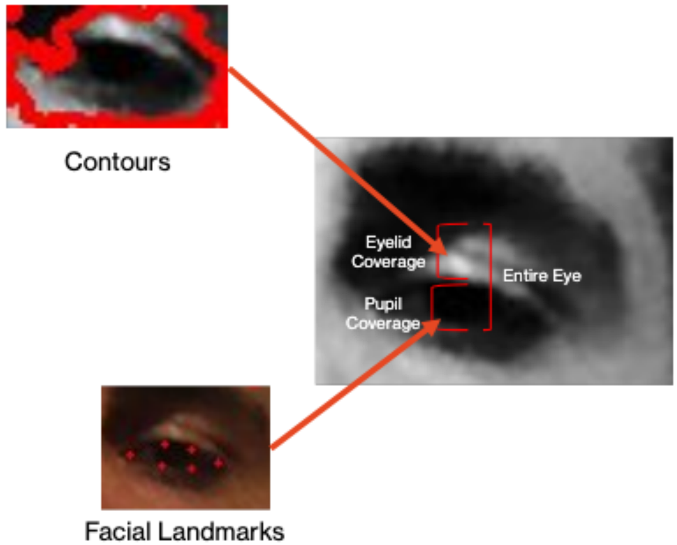

Figure 7. Diagram showing our Eyelid Coverage Steps

To ensure accuracy in eyelid coverage calculation, we defined hyperparameters such as the padding size around the eye region, the eyelid coverage threshold for drowsiness, and the kernel size for the Sobel filter. These parameters optimize feature extraction and contour detection in the image preprocessing pipeline.

3.4 Drowsiness Detection

The final step in our system involves classifying a driver as "Drowsy" or "Non-Drowsy" based on eyelid coverage measurements. Drowsiness detection leverages the coverage ratio of the eyelids, which we calculate using eye landmarks and contour analysis. When the eyelid coverage exceeds a predefined threshold (0.55 in our case), it indicates that the driver's eyelids are sufficiently closed to classify them as "Drowsy." This threshold value was chosen after extensive experimentation.

The threshold for eyelid coverage (0.55) was determined by analyzing various test cases and fine-tuning the balance between sensitivity and specificity. The chosen threshold achieved a favorable trade-off, ensuring the detection of drowsy states while minimizing false alarms.

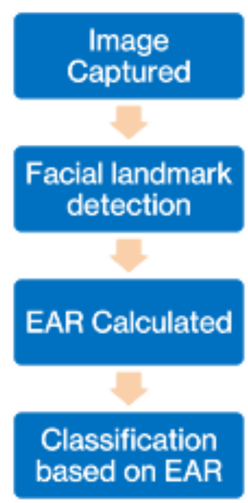

Figure 8. Algorithm 1 Pipeline

4 Experimental Setup

To evaluate the performance of our eyelid coverage-based drowsiness detection algorithm, we designed an algorithm (Algorithm 1) inspired by previous approaches, specifically the work of Kumar et al. (2018) [5]. Kumar's system employed the Eye Aspect Ratio (EAR) method, which continuously tracked the eye's openness level across video frames. In our experiment, we adapted the EAR-based method to operate on single-image inputs, allowing a direct comparison with our eyelid coverage algorithm in a more resource-efficient single-frame context.

The EAR-based method calculates the ratio of vertical to horizontal eye landmarks, providing a measure of eye openness. While the EAR is effective for continuous video streams, applying it to single-frame images allowed us to simulate a computationally efficient drowsiness detection scenario and directly evaluate the effectiveness of eyelid coverage in capturing drowsiness signs.

The algorithms were assessed using standard performance metrics such as Precision, F1-Score, Confusion Matrix, etc. To ensure robustness, we evaluated these metrics by running both algorithms on a 20% randomly sampled subset of our dataset. The test subset provided a balanced representation of drowsiness states, allowing us to assess each algorithm's ability to distinguish between the two labels.

The experiment was also conducted on an environment optimized for real-time applications, testing each algorithm's suitability for lightweight, real-time driver drowsiness detection. For example, the test ran on a standard CPU, rather than a GPU, to simulate low-resource environments typically found in embedded systems. The results of this experiment provided key insights into the efficiency of eyelid coverage in detecting drowsiness compared to the established EAR method, validating our approach as a feasible alternative for low-computational, real-time driver monitoring systems.

5. Results

The algorithm achieved a classification accuracy of 65%, which indicates a moderate overall performance in distinguishing between "Drowsy" and "Non-Drowsy" states.

- **Precision (Drowsy), 0.70**: This indicates the algorithm's reliability in correctly identifying "Drowsy" instances. In other words, when the system classifies a driver as drowsy, it has a 70% chance of being correct. This suggests that the system is fairly effective at minimizing false positives, making it useful in environments where it is crucial to avoid incorrect drowsiness alerts.
- **Recall (Drowsy), 0.59**: While the precision is relatively high, recall for "Drowsy" states is somewhat lower. This indicates that the system is missing about 41% of true "Drowsy" cases. This could be attributed to the limited sensitivity of the current feature extraction methods or threshold settings, which might not fully capture all drowsiness features.
- **F1-Score**: For drowsy detection, the F1-score was 0.64, while for non-drowsy detection, it was slightly higher at 0.67. This suggests that the system is a little more reliable in identifying non-drowsy cases than drowsy ones, although both categories are balanced.

The results indicate that while the algorithm performs well in identifying non-drowsy cases, there is room for improvement in detecting drowsy states. One possible factor contributing to the lower recall could be the challenge of distinguishing subtle variations in eyelid coverage or slight eye movements associated with drowsiness. In comparison to other algorithms:

- **Algorithm 1 (EAR-based approach)**: Algorithm 1 achieved an accuracy of 58%, which is lower than the eyelid coverage-based approach. However, Algorithm 1 showed better performance in identifying "Drowsy" states, with a high precision for drowsy detection. This algorithm performed better in a video stream setting where continuous monitoring allows for better detection of prolonged low EAR values. This is a limitation when applied to single-frame analysis, since a low EAR could also mean an eye blink.
- **Algorithm 2 (Eyelid Coverage-based approach)**: Algorithm 2, which used eyelid coverage for drowsiness detection, had an accuracy of 65%. This algorithm was equally effective in identifying both "Drowsy" and "Non-Drowsy" states, which makes it a more balanced solution. The hyperparameters for this algorithm were set with an eyelid coverage threshold of 0.55, eye padding of 5 pixels, and a Sobel filter kernel size of 7. These settings enabled more precise contour detection and eyelid coverage analysis, contributing to improved accuracy and the better overall balance between precision and recall.

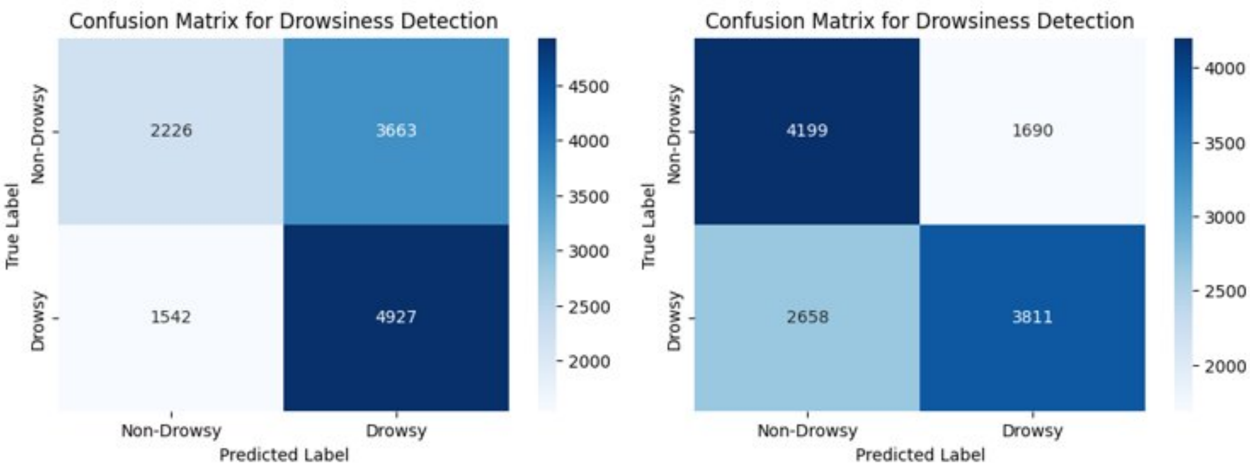

Figure 9. Algorithm 1 & 2 Confusion Matrix

5. Conclusion

This study demonstrates the potential of eyelid coverage as an effective and computationally efficient indicator for driver drowsiness detection on a single image. By analyzing eyelid coverage from eye landmarks and contour detection, the system offers a promising approach to monitor driver alertness, particularly in resource-constrained environments. While the system performed well in detecting non-drowsy states, further refinement is necessary to improve its sensitivity in detecting drowsiness, especially in borderline cases where a driver is just blinking. To enhance the system's performance, several improvements could be made, including:

- **Refining feature extraction**: Developing more advanced algorithms for detecting and processing eye contours could help the system capture subtler signs of drowsiness that were missed with the current approach.
- **Threshold calibration**: Adjusting the eyelid coverage threshold, potentially using machine learning techniques like cross-validation, could lead to more accurate classification between drowsy and non-drowsy states.
- **Incorporating additional features**: Integrating other facial landmarks, such as head position and movement, could provide complementary data that enhances the system's overall robustness and accuracy.

Although the eyelid coverage-based method demonstrates promising results, further research and fine-tuning are needed to improve the precision and recall for drowsy detection. Additionally, expanding the dataset, optimizing parameters, and exploring advanced feature extraction techniques could significantly improve detection accuracy. Future work could also involve training machine learning models, such as Support Vector Machines (SVM), on facial landmark features and eyelid contours, to create a more sophisticated one image drowsiness detection system. These advancements have the potential to enhance real-time driver monitoring systems in a resource-limited setting.

References

[1] Albadawi, Y., Takruri, M., & Awad, M. (2022). A review of recent developments in driver drowsiness detection systems. Sensors (Basel), 22(5), 2069. https://doi.org/10.3390/s22052069

[2] Daud, M. A. F., Ismail, A. P., Tahir, N. M., Daud, K., Kasim, N. M., & Mohamad, F. A. (2022). Real-time drowsy driver detection using image processing on Python. 2022 IEEE 12th International Conference on Control System, Computing and Engineering (ICCSCE), 131-136. https://doi.org/10.1109/ICCSCE54767.2022.9935627

[3] Guo, X. (2016). LIME: A method for low-light image enhancement. In Proceedings of the 24th ACM International Conference on Multimedia, 87-91. https://doi.org/10.1145/2964284.2967188

[4] Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 1867-1874. https://www.semanticscholar.org/paper/One-millisecond-face-alignment-with-an-ensemble-of-Kazemi-Sullivan/d78b6a5b0dcaa81b1faea5fb0000045a62513567

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

[6] Nasri, I., Karrouchi, M., Snoussi, H., Kassmi, K., & Messaoudi, A. (2022). Detection and prediction of driver drowsiness for the prevention of road accidents using deep neural networks techniques. In Bennani, S., Lakhrissi, Y., Khaissidi, G., Mansouri, A., & Khamlichi, Y. (Eds.), WITS 2020: Lecture Notes in Electrical Engineering, Vol. 745. Springer, Singapore. https://doi.org/10.1007/978-981-33-6893-4_6

[7] Ni, F., Fu, Z., Cao, Q., & Zhao, Y. (2008). Image processing method for eyes location based on segmentation texture. Sensors and Actuators A: Physical, 143(2), 439-451. https://doi.org/10.1016/j.sna.2007.11.033

[8] Albadawi, Y., Takruri, M., Awad, M. (2022). Review of recent developments in driver drowsiness detection systems. National Center for Biotechnology Information. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8914892/

[9] Kumar, A., Patra, R. (2018). Driver drowsiness monitoring system using visual behaviour and machine learning. IEEE Xplore. Retrieved from https://ieeexplore.ieee.org/document/8405495

[10] Ananthi, S., Sathya, R., Vaidehi, K., Vijaya, G. (2023). Drivers drowsiness detection using image processing and I-Ear techniques. IEEE Xplore. Retrieved from https://ieeexplore.ieee.org/document/10142501

[11] Soukupova, T., & Čech, J. (2016). Real-time eye blink detection using facial landmarks. Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. Retrieved from https://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf

[12] Raw dataset: https://www.kaggle.com/datasets/ismailnasri20/driver-drowsiness-dataset-ddd

[13] Preprocessed dataset: https://www.kaggle.com/datasets/franciscolozdatasci/csc481-project-dataset

[14] Code repository: https://github.com/FranciscoLozCoding/csc481_Project