Jurnal Ilmiah Informatika dan Komputer (Informatech)
Volume 2, Nomor 2, December 2025, page 152-157
DOI : 10.69533
Website : *https://ejournal.rizaniamedia.com/index.php/informatech*

E-ISSN: 3047-4752

Real-Time Driver Microsleep Detection Using Lightweight MobileViT and Haar Cascade

Muhammad Alfariz Rasyid[1], Syahda Mawarda Hutagalung[2*], Supiyandi[3], Aidil Halim Lubis[4]

[1,2*,4)]Universitas Islam Negeri Sumatera Utara, Jl. Lapangan Golf, Desa Durian Jangak, Kecamatan Pancur Batu, Kabupaten Deli Serdang, Sumatera Utara, 20353
[3)]Universitas Pembangunan Panca Budi, Jl. Jend. Gatot Subroto Km 4,5, Sei Sikambing, Medan, Sumatera Utara, 20122
[1]*email : m.alfarizrasyid77@gmail.com*
[*2]*email: syahadamawardahutagalung@gmail.com*
[3]*email: supiyandi.mkom@gmail.com*
[4]*email: aidilhalimlubis@uinsu.ac.id*

(Article Received: 10 October 2025; Article Revised: 29 November 2025; Article Published: 11 December 2025;)

ABSTRACT *– Microsleep is a brief and involuntary loss of awareness that increases the risk of driving accidents. This study proposes a real-time microsleep detection system using Haar Cascade for eye localization and the lightweight MobileViT-XXS model for eye-state classification. The model was trained on a public dataset and achieved a training accuracy of* ***99.49%****, a test accuracy of* ***98%****, and a real-time accuracy of* ***90–94%****. A microsleep event is detected when the eyes remain closed for* ***≥ 2 seconds****. While the method performs well under controlled conditions, real-time testing revealed technical limitations such as sensitivity to lighting variation, non-frontal head pose, and motion, which affect detection stability and represent common robust-vision challenges. Despite these limitations, the system runs efficiently on CPU-only hardware and demonstrates strong potential as a lightweight early-warning system to support driving safety. Future research may explore expanding dataset diversity, improving environmental adaptation, and deploying the system on embedded or mobile platforms to enhance robustness and scalability.*

Keywords *- Driver Monitoring, Haar Cascade, Microsleep, MobileViT, Real-Time Detection*

Deteksi Microsleep Pengemudi Secara Real-Time Menggunakan Leightweight MobileViT dan Haar Cascade

ABSTRAK *– Microsleep merupakan kehilangan kesadaran secara singkat dan tidak disengaja yang dapat meningkatkan risiko kecelakaan saat berkendara. Penelitian ini mengusulkan sistem deteksi microsleep secara real-time dengan menggunakan Haar Cascade untuk pelokalan mata dan model ringan MobileViT-XXS untuk klasifikasi kondisi mata. Model dilatih menggunakan dataset publik dan menghasilkan akurasi pelatihan* ***99.49%****, akurasi pengujian* ***98%****, serta akurasi real-time pada kisaran* ***90–94%****. Microsleep terdeteksi ketika mata tetap tertutup selama* ***≥ 2 detik****. Meskipun metode ini menunjukkan performa tinggi pada kondisi terkontrol, pengujian real-time mengungkapkan keterbatasan teknis seperti sensitivitas terhadap pencahayaan, posisi kepala tidak frontal, gerakan kepala, dan pantulan kacamata, yang memengaruhi stabilitas deteksi dan merupakan tantangan umum dalam sistem visi komputer yang membutuhkan robustnes tinggi. Meskipun demikian, sistem tetap berjalan efisien pada perangkat berbasis CPU dan memiliki potensi kuat sebagai solusi peringatan dini untuk meningkatkan keselamatan berkendara. Penelitian selanjutnya dapat difokuskan pada peningkatan variasi dataset, adaptasi lingkungan, serta implementasi pada perangkat embedded atau mobile untuk meningkatkan skalabilitas dan performa nyata.*

Kata Kunci *– Deteksi Kantuk, Haar Cascade, Microsleep, MobileViT, Real-Time*

Jurnal Ilmiah Informatika dan Komputer (Informatech)
Volume 2, Nomor 2, December 2025, page 152-157
DOI : 10.69533
Website : *https://ejournal.rizaniamedia.com/index.php/informatech*
E-ISSN: 3047-4752

1. INTRODUCTION

Driver fatigue is widely acknowledged as a major factor contributing to road accidents, and among its manifestations, microsleep is considered the most dangerous. Microsleep occurs suddenly and typically lasts only one to three seconds, yet during this brief lapse of awareness the driver may lose control of the vehicle, significantly increasing the likelihood of serious or fatal crashes. Because of these risks, developing a reliable, lightweight, and real-time microsleep detection system remains an important research direction in intelligent transportation and driver safety monitoring.

Recent studies have introduced a variety of approaches using computer vision and deep learning. Chari et al. [1] explored a hybrid model combining deep learning and machine learning for detecting drowsiness based on facial features. Although this approach achieved promising accuracy, the prediction mechanism focused on single-frame classification, making it difficult to distinguish normal blinking from prolonged eye closure associated with microsleep. Another study, Real-Time Drowsiness Detection. Begum et al. [2], implemented Haar Cascade for eye localization combined with a temporal eye-closure analysis. While this method successfully achieved real-time detection, the classification relied on conventional rule-based processing, which is less adaptive to variations in eye shape, and visual conditions. Meanwhile, Delwar et al. [3] applied a deep learning model for eye-state recognition that performed well in controlled settings but required high computational resources and GPU support, limiting its feasibility for deployment in low-power embedded systems.

These findings indicate that existing approaches either rely on conventional feature-based classification, lack temporal awareness for identifying microsleep, or require high-performance hardware to operate effectively. Therefore, there remains a need for a method that is computationally efficient, capable of temporal interpretation, and adaptable to visual variation for use in real driving environments.

To address these gaps, this study proposes a real-time microsleep detection system that integrates Haar Cascade for region-of-interest localization with a lightweight MobileViT-based model for classifying eye states. A temporal threshold (≥ 2 seconds) is applied to differentiate normal blinking from potential microsleep events. This hybrid design aims to balance speed, accuracy, and computational efficiency, enabling the system to run in real time on non-GPU devices while maintaining reliable detection performance in real-world usage scenarios.

2. LITERATURE RVIEW

The development of driver drowsiness detection systems using computer vision continues to grow, as this approach is considered more convenient compared to biological sensor-based methods such as EEG or EOG. Fu et al. [4] state that monitoring eye and facial conditions through a camera can provide real-time information without requiring direct physical contact with the driver. Similarly, Fonseca et al. [5] report that deep learning-based models are now widely adopted as the primary method for microsleep detection through classification of open- or closed-eye states.

Technically, CNN architectures have dominated research in this field for a long period, although recent trends indicate a shift toward Vision Transformer-based approaches that offer more stable performance under changes in illumination and head pose [6]. To improve processing efficiency, especially on embedded or in-vehicle systems, lightweight models such as MobileViT have been designed to maintain high accuracy while reducing computational requirements [7].

In the stage of detecting the face or eye region, classical approaches such as Haar Cascade remain relevant due to high processing speed. Studies by Ilmadina et al. [8] demonstrate that using Haar Cascade during preprocessing before deep learning classification can improve eye-state detection accuracy. Meanwhile, Makowski et al. [9] emphasize that the duration of eye closure is a more reliable parameter for detecting microsleep compared to single indicators like blink frequency.

Based on the reviewed studies, lightweight deep learning models combined with classical and modern detection approaches represent the primary direction for real-time drowsiness detection research. However, limitations such as lighting variations, glasses usage, and head movement remain significant challenges requiring further investigation for real-world implementation .

3. RESEARCH METHODS

This study applies a quantitative experimental approach to develop and evaluate a real-time microsleep detection system based on eye imagery. The research process includes dataset collection, data preprocessing, Lightweight MobileViT model

training, performance evaluation, and real-time system implementation using the Haar Cascade technique.

3.1 Dataset

The dataset used in this study consists of eye images belonging to two classification categories: **open-eye** and **closed-eye** states. All images were collected from a public dataset hosted on the Kaggle platform. The original dataset was arranged into two main directories, namely train and test, following a common structure used in image classification tasks. Prior to model development, the dataset was manually reviewed to ensure the absence of corrupted files, duplicates, and mislabeled samples.

Table 1. Dataset Structure

Dataset Type	Closed	Open	Total
Train Set	2.377	2.257	4.634
Validation Split (20% of training)	480	446	926
Test Set	349	469	818
Overall Total	3.206	3.172	6.378

As shown in Table 1, the dataset maintains relatively balanced proportions between the two classes, which is crucial for preventing model bias toward a dominant label. Although the dataset already included separate train and test directories, the contents of the training directory were further divided into 80% training data and 20% validation data. This additional split was intentionally performed to monitor model performance during training, fine-tune hyperparameters, and detect early signs of overfitting without exposing the model to the test data. Keeping the test set untouched until final evaluation ensures an unbiased and reliable assessment of the model's real-world generalization capability.

3.2 Data Preprocessing

The preprocessing stage was carried out to standardize the dataset and ensure compatibility with the MobileViT model. Each input image underwent several transformation steps including:

1. Image resizing, where all samples were resized to 256 × 256 pixels to match the expected input dimensions of the MobileViT architecture.
2. Normalization, performed using ImageNet mean and standard deviation values to align the dataset distribution with the pretrained backbone configuration.
3. Tensor conversion, where the images were transformed from PIL format into PyTorch tensors for compatibility with the deep learning framework.

In addition, during the real-time phase, the eye region was first extracted using the Haar Cascade classifier to isolate the region of interest (ROI) before applying the same preprocessing steps. This additional stage ensured that the model received consistent, centered, and noise-reduced inputs during inference.

3.3 Model Architecture

The model applied in this study is **MobileViT-XXS**, a lightweight variant of the Vision Transformer architecture designed for efficient image processing. MobileViT employs a self-attention mechanism that enables the model to learn broader spatial dependencies within visual features, allowing it to capture structural patterns more effectively than traditional localized approaches.

In this work, the model was adapted for binary classification to distinguish between open-eye and closed-eye states. The final layer was modified to produce a probability output, which serves as the decision basis for eye-state recognition in each video frame.

Due to its compact design, low memory requirements, and fast inference performance, this architecture is well-suited for real-time microsleep detection and can operate reliably even on devices with limited computational resources.

3.4 Model Training

The model was trained using PyTorch with the following configuration:

Table 2. Model Training Configuration

Parameter	Value
Epoch	5
Batch Size	8
Learning Rate	0.0008
Optimizer	Adam
Loss Function	Binary Cross-Entropy (BCELoss)

As shown in Table 2, the model was trained using a learning rate of 0.0008, which allows gradual and stable convergence during optimization. The Adam optimizer was selected due to its adaptive update mechanism, making it suitable for transformer-based models such as MobileViT.

Training was performed over five epochs with a batch size of eight. During training, validation monitoring was applied to track performance and prevent overfitting. The best-performing model was automatically saved as drowsy_model.pth based on the highest validation accuracy recorded.

3.5 Microsleep Detection Implementation

The trained MobileViT-XXS model was deployed in a real-time microsleep detection system using a webcam as the input device. Each captured frame is processed to detect the face and eye region using the

Haar Cascade classifier, and the detected Region of Interest (ROI) is then used as input for classification.

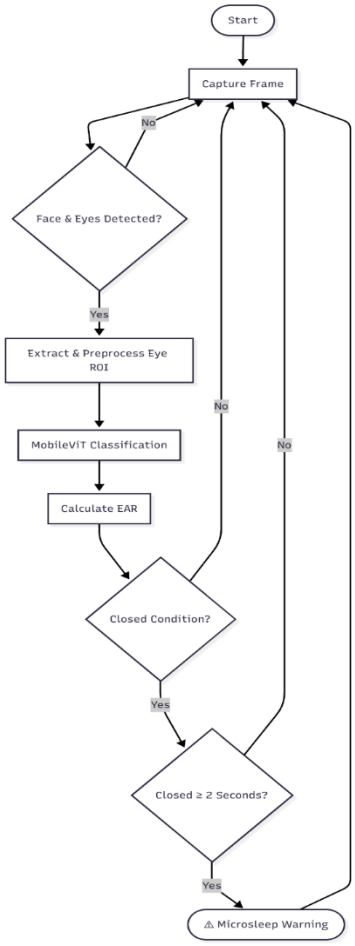

Figure 1. Real-Time Microsleep Detection Flowchart

As shown in Figure 1, the detected eye ROI is processed and classified using the MobileViT model to identify whether the eyes are open or closed. Simultaneously, the system calculates the Eye Aspect Ratio (EAR) using MediaPipe, providing an additional decision-support mechanism to improve accuracy under varying lighting, angle, or facial conditions.

If the eye remains closed for 2 seconds or longer, the system identifies the event as a potential microsleep and issues a visual alert. This duration-based method allows the system to differentiate normal rapid blinking from prolonged involuntary eye closure, making it more suitable for real-world driver monitoring applications.

3.6 Testing Environment and Hardware Setup

The system was tested in a controlled environment to ensure stable and consistent real-time performance. The testing was carried out on standard hardware without GPU acceleration to evaluate whether the system could operate efficiently under practical and commonly available computing conditions.

Table 3. Experimental Environment and System Configuration

Component	Specification
Programming Language	Python
Libraries	PyTorch, OpenCV, timm, torchvision, MediaPipe
Mode;	MobileViT-XXS
Device	CPU + Webcam
Output	Real-time prediction, EAR montoring and microsleep alert

As shown in Table 3, the system was implemented using Python with supporting frameworks such as PyTorch for model inference, OpenCV for video processing, and MediaPipe for computing the Eye Aspect Ratio (EAR). The trained model was executed on a CPU-powered device with a standard webcam serving as the video input source. This configuration was chosen to evaluate system feasibility under realistic hardware limitations and demonstrate that the proposed method can operate effectively without requiring GPU acceleration.

4. Result And Discussion

4.1 Model Training Results

The model was trained for five epochs, and the performance consistently improved throughout the training process. The training accuracy increased gradually across epochs, while the validation loss showed a decreasing trend, indicating that the model was effectively learning to distinguish between open and closed eye conditions.

Table 4. Model Training Performance per Epoch

Epoch	Train Accuracy	Validation Accuracy	Validation Loss	Overfitting Gap
1	96.57%	99.68%	0.0209	-3.10%
2	99.27%	99.78%	0.0090	-0.51%
3	98.60%	99.89%	0.0033	-1.29%
4	98.65%	99.14%	0.0207	-0.48%
5	99.49%	99.68%	0.0130	-0.19%

As shown in Table 4, the gap between training and validation accuracy remained small, ranging from –3.10% to –0.19%, indicating that the model generalizes well and does not show clear signs of overfitting during training. The consistently high accuracy may be influenced by the uniform characteristics of the dataset, such as balanced classes and consistent lighting. However, such results may not fully represent real-world performance, where lighting variation, head pose changes, and occlusions are common. Therefore, further evaluation using a more diverse dataset and additional regularization or hyperparameter optimization may be required to reduce potential overfitting risk and improve robustness.

4.2 Model Testing Results

To further evaluate generalization performance, the trained model was tested using unseen data.

Table 5. Model Testing Performance

Parameter	Result
Test Accuracy	98%
Test Loss	0.0568

As shown in Table 5, the model achieved near-perfect accuracy on the test dataset. This outcome reinforces the strong performance observed during training; however, such near-ideal accuracy may also indicate that the evaluation dataset remains relatively controlled and does not fully represent the complexities of real-world operating conditions.

4.3 Implementation Results of the Microsleep Detection System

Following successful testing on static images, the model was deployed in a real-time application using a webcam. The system detected the user's face, extracted the eye region, and performed classification into two possible states: **OPEN** or **CLOSED**. Additionally, the system monitored the duration of eye closure to determine potential microsleep events using a threshold of **≥ 2 seconds**.

The following is an example of system output during real-time testing using a webcam:

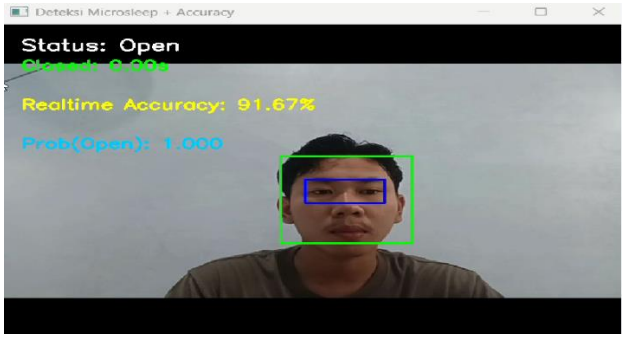

Figure 2. Detection Result: Eye Open State

As shown in **Figure 2**, the system correctly identified the eye as open, and no alert was triggered.

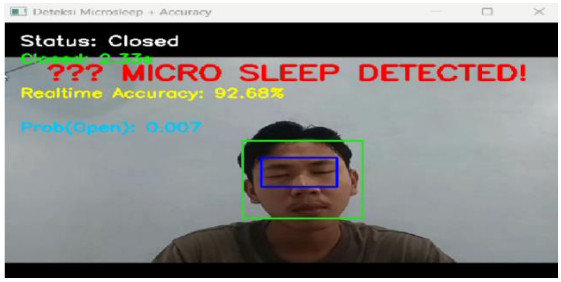

Figure 3. Detection Result: Eye Open State

As shown in **Figure 3**, once the eyes remained closed beyond the two-second threshold, the system issued a warning indicating a possible microsleep event.

The accuracy during real-time execution was calculated by comparing the system's predictions with the estimated eye condition detected during each valid frame. Accuracy was computed as the percentage of correct predictions relative to the total number of analyzed frames. Using this method, the real-time accuracy ranged between **90–94%**. This result shows a noticeable gap compared to the near-perfect performance during dataset testing, primarily due to additional real-world challenges such as lighting variations, non-frontal face angles, movement, and glasses reflections.

4.4 Discussion

Overall, the MobileViT-XXS model showed strong performance when tested on structured image data, indicating that it can effectively differentiate between open and closed eye states under controlled conditions. However, when implemented in real-time, the performance was not as consistent, suggesting that environmental variability—such as lighting differences, head movement, changing face angles, or partial occlusion—affects system stability. This gap between controlled testing and real-time usage highlights the challenge of adapting machine learning models to dynamic environments. Although the system still functioned reliably and demonstrated the ability to detect microsleep behavior, further refinement in model training, dataset diversity, and real-time processing strategies may be necessary to improve robustness and ensure consistent performance in practical applications.

5. CONCLUSION

This study demonstrates that the MobileViT-XXS model is capable of achieving high performance in eye-state classification, with near-perfect accuracy on the dataset and real-time accuracy ranging between 90–94%. These results indicate that lightweight deep learning models can operate efficiently on non-GPU devices, serving as a practical contribution to the informatics field in developing real-time microsleep detection systems. However, the decrease in accuracy during real-time implementation shows that dynamic environments remain a major challenge. Therefore, future work should focus on increasing dataset variability, adapting the model to diverse environmental conditions, and exploring deployment on embedded or mobile platforms to ensure better scalability and more stable performance.

BIBLIOGRAPHY

[1] G. Shiva, S. Chari, and J. A. Prashant, "Real-time driver drowsiness detection based on integrative approach of deep learning and

machine learning model," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 39, no. 1, pp. 592–602, 2025, doi: 10.11591/ijeecs.v39.i1.pp592-602.

[2] S. Begum, S. Iqbal, and M. Foorqan, "Real-Time Drowsiness Detection," *IRE Journals*, vol. 9, no. 6, pp. 140–145, 2025, doi: DOI: https://doi.org/10.64388/IREV9I6-1712571.

[3] T. S. Delwar, M. Singh, S. Mukhopadhyay, A. Kumar, and D. Parashar, "AI- and Deep Learning-Powered Driver Drowsiness Detection Method Using Facial Analysis," pp. 1–24, 2025.

[4] B. Fu, F. Boutros, C. Lin, and N. Damer, "A Survey on Drowsiness Detection – Modern Applications and Methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 3, pp. 1–23, 2024.

[5] T. Fonseca, "Drowsiness Detection in Drivers : A Systematic Review of Deep Learning-Based Models," *Appl. Sci.*, vol. 15, pp. 1–43, 2025, doi: https://doi.org/10.3390/app15169018.

[6] T. M. Hassan, W. Y. Ibrahim, H. A. Ali, and H. S. Marie, "Deep Learning-Based Driver Drowsiness Detection Using Facial Expression Analysis Faculty of Artificial Intelligence , Delta University for Science and Technology , Gamasa 35712 , Faculty Artificial Intelligence , Delta University for Science and Technolog," *Delta Univ. Sci. J.*, vol. 07, no. 2, pp. 164–176, 2024, doi: https://dusj.journals.ekb.eg.

[7] N. Datta, T. Mahmud, M. Begum, M. T. Aziz, and D. Islam, "Convolutional neural network-based real-time drowsy driver detection for accident prevention," *TELKOMNIKA*, vol. 23, no. 3, pp. 682–693, 2025, doi: 10.12928/TELKOMNIKA.v23i3.26059.

[8] H. Z. Ilmadina, D. Apriliani, and D. S. Wibowo, "Deteksi Pengendara Mengantuk dengan Kombinasi Haar Cascade Classifier dan Support Vector Machine," vol. 7, no. 1, pp. 1–7, 2022.

[9] S. Makowski, P. Prasse, L. A. J, and T. Scheffer, "Detection of Drowsiness and Impending Microsleep from Eye Movements," *Proc. Mach. Learn. Res.*, vol. 1, pp. 1–18, 2023.

[10] M. Venkateswarlu and V. R. R. Ch, "DrowsyDetectNet: Driver drowsiness detection using lightweight CNN with limited training data," *IEEE Access*, vol. 12, pp. 1–15, 2024, doi: 10.1109/ACCESS.2024.3440585.

[11] D. Salem and M. Waleed, "Real-time drowsiness detection via convolutional neural networks and transfer learning," *J. Eng. Appl. Sci.*, vol. 71, no. 122, pp. 1–10, 2024

[12] R. Florez et al., "A real-time embedded system for driver drowsiness detection," *Sensors*, vol. 24, no. 7810, pp. 1–15, 2024, doi: 10.3390/s24087810.

[13] A. Phan et al., "Driver drowsiness detection and smart alerting using deep learning and IoT," *Multimedia Tools and Applications*, pp. 1–23, 2023, doi: 10.1007/s11042-023-15987-2.

[14] M. Venkateswarlu et al., "CNN-ViT: A multi-feature learning approach for driver drowsiness detection," *Pers. Ubiquitous Comput.*, pp. 1–12, 2025.

[15] A. Bhanja et al., "A real-time driver drowsiness detection system," *ROBOSEM Journal*, pp. 1–10, 2025.

[16] S. Priyanka, R. Kumar, and A. Ahmed, "Data fusion-based multimodal deep learning for drowsiness recognition," *Computers & Electrical Engineering*, vol. 109, pp. 1–15, 2024, doi: 10.1016/j.compeleceng.2024.109050.

[17] R. Iswahyudi, A. F. Ihsan, and I. M. M. Matin, "Deteksi kantuk menggunakan CNN berdasarkan kedipan mata," *J. Fis. Teknol.*, vol. 4, no. 2, pp. 45–53, 2024.

[18] A. Turki, S. Ammar, M. Karray, and M. Ksantini, "Facial expression-based drowsiness detection system for driver safety using deep learning techniques," in *Proc. 16th ICAART*, vol. 3, pp. 726–733, 2024, doi: 10.5220/0012386000003655.

[19] J. Chao, K. Li, and W. Sun, "Real-time microsleep detection using attention-based deep learning," *IEEE Trans. Human-Machine Systems*, vol. 54, no. 1, pp. 112–123, 2025.

[20] H. Wu and L. Zhang, "Mobile vision transformer for lightweight driver fatigue detection," *IEEE Sensors Journal*, vol. 26, no. 4, pp. 2150–2162, 2026,