 https://doi.org/10.24191/jeesr.v29i1.019

Lightweight Driver Drowsiness Detection using Haar Cascade and MobileNetV2

Muhammad Syahir Abdul Shukor and Husna Zainol Abidin*

Abstract— **Driver drowsiness affects the driver's attention while driving which could harm the driver himself and other road users due to lack of alertness and slower reaction times. The risk of having an accident due to the driver drowsiness could be reduced through practical non-intrusive monitoring systems. This paper demonstrates an effective, lightweight proof-of-concept framework for real-time driver monitoring under laboratory webcam conditions. Full on-road vehicular validation remains subject to future field trials under naturalistic driving environments. The detection framework combines Haar Cascade and MobileNetV2 Convolutional Neural Network (CNN) where Haar Cascade is utilized for face detection and extraction of the facial Region of Interest (ROI), while MobileNetV2 classifies the driver's eye states and yawning behaviour. The system was implemented in Python using OpenCV and TensorFlow and integrated with a standard webcam. The performance of the proposed lightweight architecture, MobileNetV2 was compared with VGG16 using the same eye state and yawning datasets. The results show that MobileNetV2 outperforms VGG16 where the accuracy for eye state classification is 98.17% and 88.07% respectively. In terms of yawning, the accuracy for MobileNetV2 is 78.60%, while VGG16 is 66.51%. The findings demonstrate that the integration of Haar Cascade and MobileNetV2 provides an efficient, lightweight, and practical approach for real-time driver drowsiness detection, with promising applications in intelligent transportation systems and vehicle safety technologies.**

Index Terms—**CNN, computer vision, driver drowsiness detection, eye state, Haar Cascade, MobileNetV2, yawning.**

I. INTRODUCTION

Road traffic accidents continue to pose a significant challenge to public safety, resulting in approximately 1.19 million fatalities each year worldwide [1]. Among the major contributing factors, driver drowsiness due to fatigue has been identified as a critical cause because it reduces alertness, slows reaction time, and impairs decision-making, thereby increasing the risk of traffic accidents [2]. Naturalistic driving studies have also shown that drowsiness is associated with a meaningful proportion of crashes [3]. In Malaysia, the Malaysian Institute of Road Safety Research (MIROS) reported that microsleep is responsible for approximately 20% of road accidents [2]. These observations motivate the development of monitoring systems that can identify visible fatigue indicators and provide timely warnings without requiring the driver to wear intrusive sensors.

The rapid advancement of Artificial Intelligence (AI) and computer vision has accelerated the development of vision-based driver monitoring systems. Recent studies have shown that Convolutional Neural Networks (CNNs) can effectively recognize visual fatigue indicators, including eye closure and yawning, directly from facial images [3], [4]. Despite these advances, many existing approaches depend on computationally demanding deep learning models or dedicated hardware such as infrared cameras and wearable sensors. Furthermore, their detection performance may deteriorate under challenging conditions, including poor illumination, facial occlusion, and head pose variation, limiting their suitability for practical, low-cost applications [5], [6]. Recent studies continue to explore both lightweight CNNs and more complex transformer-based architectures for real-time drowsiness monitoring, illustrating the trade-off between computational demand and recognition performance.

This paper presents an effective, lightweight proof-of-concept framework for real-time driver monitoring under laboratory webcam conditions that combines the Haar Cascade algorithm with the MobileNetV2 deep learning model. Haar Cascade is utilized to detect the driver's face and extract the facial Region of Interest (ROI), while MobileNetV2 is responsible for classifying eye state and yawning behaviour from images captured by a standard webcam. To enhance driving safety, the proposed system generates both visual and audible alerts whenever prolonged eye closure or yawning is detected. The performance of the proposed approach is assessed in terms of accuracy, precision, recall, F1-score, and confusion matrix, with the aim of providing a lightweight, cost-effective, and reliable solution for real-time driver fatigue monitoring. The proposed architecture was designed with computational efficiency in mind while maintaining satisfactory detection performance.

The study offers three key technical contributions to lightweight driver monitoring. First, a cascaded dual-ROI architecture combining low-overhead Haar Cascade localization with two independently fine-tuned MobileNetV2 classifiers for sub-region eye closure and facial yawning patterns, minimizing computational redundancy. Secondly, a comprehensive empirical benchmark evaluating trade-offs in

This manuscript is submitted on August 30, 2026, revised on October 4, 2026, accepted on October 13, 2026 and published on October 31, 2026.

Muhammad Syahir Abdul Shukor was a student with the Faculty of Electrical Engineering, Universiti Teknologi MARA (e-mail: 2023686464@student.uitm.edu.my).

Husna Zainol Abidin is with the Faculty of Electrical Engineering, Universiti Teknologi MARA (e-mail: husnaza@uitm.edu.my).

*Corresponding author
Email address: husnaza@uitm.edu.my

1985-5389/© 2026 The Authors. Published by UiTM Press. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

accuracy against deep CNN baselines (VGG16) and finally a continuous frame-buffered decision framework incorporating a two second bilateral eye-closure threshold and yawning frequency rules to minimize false alarms caused by natural blinking. The scope of the evaluation is limited to the available public image datasets and functional webcam tests under typical indoor lighting; therefore, the study does not claim validated on-road performance.

II. Related Works

Driver drowsiness detection has been extensively studied as an effective approach to reduce fatigue-related road accidents. Most existing research focuses on the use of computer vision, image processing, and deep learning to continuously monitor driver behaviour through facial features captured by an in-vehicle camera [5]-[7]. Unlike sensor-based approaches, vision-based systems provide a non-intrusive solution by analysing indicators such as eye closure, blinking frequency, head movement, and yawning.

A driver fatigue detection system based on image processing and conventional machine learning, obtaining an accuracy of 96.67% using data collected from 30 drivers was proposed by Aboagye et al. [5]. On the other hand, Yousif [6] adopted a fuzzy logic approach that analysed head and eye movements from video sequences and achieved approximately 90% detection accuracy. Although these methods produced encouraging results, their effectiveness is highly influenced by the selected algorithms, dataset characteristics, and testing environments.

Among deep learning techniques, Convolutional Neural Networks (CNNs) have shown excellent performance in image classification and are widely adopted for driver monitoring applications. Li et al. [4] developed an edge computing-based fatigue detection system that combined facial alignment with a CNN model, achieving 88.6% detection accuracy in a simulated driving environment while Jothimani et al. [7] proposed a real-time driver alert system using a pre-trained CNN, reporting an accuracy of 92%. These studies demonstrate the capability of CNN models to perform reliable fatigue detection under real-time conditions.

Recent lightweight driver drowsiness detection studies have increasingly adopted efficient CNN and object-detection architectures. MobileNet-based models have been explored because of their reduced parameter count and suitability for edge deployment, while recent approaches have integrated MobileNet backbones into YOLO frameworks to balance detection accuracy and computational efficiency. For example, lightweight YOLOM-Net employs MobileNetV3 within a YOLOv8 framework, while YOLO-FDCL incorporates MobileNetV4 to reduce model complexity while maintaining strong detection performance [8].

Other recent studies have also demonstrated the effectiveness of MobileNetV3 and MobileNetV2 for lightweight driver-fatigue classification [9].

CNN architectures for image classification include VGG16, GoogLeNet, ResNet, and MobileNetV2 [10]-[20]. VGG16 employs a deep architecture consisting of 16 layers with small 3×3 convolution filters, enabling effective feature extraction but requires relatively high computational resources [15]. GoogLeNet introduces the Inception module, which performs multiple convolution operations simultaneously to improve computational efficiency while maintaining high classification accuracy. ResNet utilizes residual connections to address the vanishing gradient problem, allowing much deeper networks to be trained effectively and improving recognition performance [19]. In contrast, MobileNetV2 is specifically designed for mobile and embedded vision applications by employing depthwise separable convolutions and inverted residual blocks, significantly reducing computational complexity while maintaining competitive classification accuracy [20].

Owing to its lightweight architecture and fast inference speed, MobileNetV2 was selected as the primary CNN model in this paper for real-time driver drowsiness detection. On top of that, its depthwise separable convolutions and inverted residual blocks reduce computational complexity relative to conventional deep CNNs while retaining useful image representations [15]. VGG16 was used as a comparison architecture because it is a well-established CNN baseline with substantially greater computational demand [12]. Compared with these approaches, the proposed framework focuses on two direct behavioural indicators, namely eye closure and yawning, followed by temporal verification for alert generation.

Based on the reviewed studies, computer vision combined with artificial intelligence provides a promising solution for real-time driver drowsiness detection. Nevertheless, challenges remain in terms of computational efficiency, limited dataset diversity, and reduced robustness under poor illumination, facial occlusion, and head pose variations. To address these issues, this paper focuses on the Haar Cascade algorithm for rapid face localisation with MobileNetV2 for eye state and yawning classification. The proposed framework is designed to deliver accurate real-time fatigue detection while maintaining low computational complexity, allowing deployment on a standard webcam without requiring specialized hardware.

III. Methodology

The proposed driver drowsiness detection system was developed based on Deep Learning (DL). DL utilizes artificial neural networks with multiple layers to automatically extract image features, making it highly effective for computer vision applications such as driver drowsiness detection [16], [17]. Convolutional Neural Networks (CNNs) are DL models widely adopted for image classification tasks. A standard CNN architecture typically incorporates convolutional layers, pooling layers, dropout, flattening, and fully connected layers as depicted in Fig. 1. These components collectively facilitate effective feature extraction and accurate classification [11], [18], [19].

Fig. 2 illustrates the overall workflow of the proposed system. This paper adopted a quantitative experimental approach to develop and evaluate a real-time driver drowsiness detection system. The proposed system integrates computer vision and DL techniques, where Haar Cascade is employed for face detection and facial region extraction, while a

MobileNetV2-based CNN is used to classify eye state (open or closed) and yawning behaviour. The overall methodology consists of six phases, namely dataset preparation, CNN model development, model training, system integration, real-time fatigue detection, and performance evaluation.

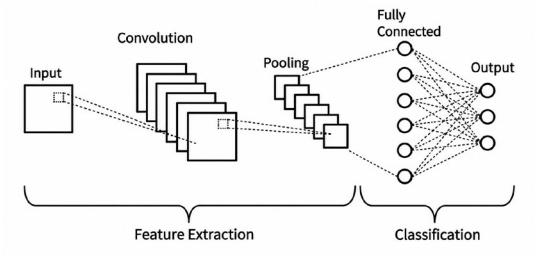

Fig. 1. Structure of CNN layers [19].

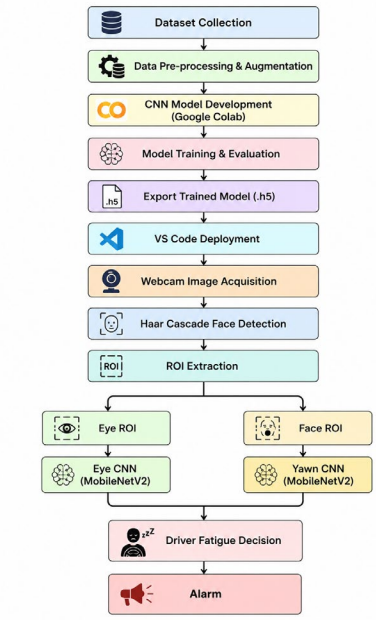

Fig. 2. Methodology of the system

A. Dataset Collection and Preprocessing

Two publicly available image datasets obtained from Kaggle [21] were utilized in developing the system. The Eye State Dataset contains 1,452 cropped facial sub-region images, perfectly balanced between 726 'Open' and 726 'Closed' eye samples. The Yawning Dataset comprises 1,448 full-face images, evenly distributed between 724 'Yawn' and 724 'No-Yawn' instances, captured across approximately 60 distinct human subjects under varying indoor and vehicular ambient lighting. All images were standardized to 224 × 224 × 3 in RGB format. Preprocessing included pixel intensity normalization between 0 and 1. To enhance model generalization, dynamic data augmentation was applied during training using TensorFlow's ImageDataGenerator, comprising around 15% random rotations, 20% zoom variations, horizontal flips, brightness adjustments between 0.8 and 1.2, and 10% spatial shifts. The dataset was partitioned using an 85%/15% macro-split, with 15% reserved as a testing set. The remaining 85% was then divided into training and validation subsets using an 80%/20% ratio. The breakdown of the samples used is as follows:

- Eye State Model (1,452 images): 987 images were allocated for training, 247 for validation, and 218 for testing. The testing set comprised 109 Open-eye and 109 Closed-eye images.
- Yawning Model (1,448 images): 986 images were allocated for training, 247 for validation, and 215 for testing. The testing set comprised 109 Yawn and 106 No-Yawn images.

As the publicly available image repository contains pre-extracted frame samples without consistent subject-level identifiers, complete subject-independent separation between the training, validation, and testing sets could not be ensured. This limitation is therefore acknowledged as a constraint of the experimental design. Fig. 3 illustrates sample images from the eye state and yawning datasets.

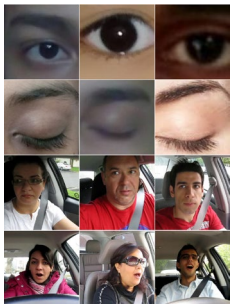

Fig. 3. Sample images of the eye state and yawning datasets.

B. CNN Model Development

The eye state and yawning classifiers were developed separately using MobileNetV2 through transfer learning in Google Colab. The pre-trained ImageNet weights were adopted while the original classification layer was removed. A new classification head consisting of a Global Average Pooling layer, a fully connected layer with 128 neurons, a Dropout layer of 0.5, and a sigmoid output neuron was added to perform binary classification [13]. During training, the feature extraction layers were frozen while only the newly added classifier layers were optimized.

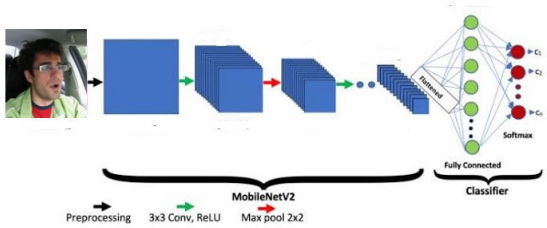

Fig. 4. MobileNetV2 architecture with the proposed classification head.

The CNN models were trained using the Adam optimizer with a learning rate of 0.0001, batch size of 32, and binary cross-entropy loss, following a standard transfer learning setup for MobileNetV2-based image classification [20], [21]. Training was configured for a maximum of 20 epochs, with Early Stopping (patience = 3) applied to prevent overfitting by halting the process when validation loss failed to improve for three consecutive epochs.

C. Real-Time System Development

The trained CNN models were integrated into a real-time application developed in Visual Studio Code using Python. OpenCV was used to acquire live video frames from a standard webcam. Haar Cascade classifiers were employed to detect the driver's face from each video frame. After face localization, the detected facial region was extracted as the Region of Interest (ROI). The eye region was identified using Haar Cascade eye detection, while the entire facial region was supplied to the yawning detection model. The extracted ROIs were resized to 224 × 224 pixels before being passed to their respective CNN models for classification. The system is summarized in Fig. 5.

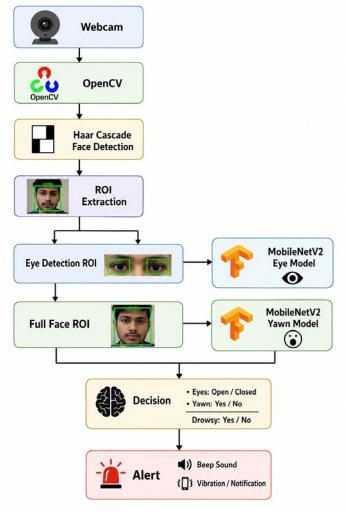

Fig. 5. Real-time detection system.

The eye state detection model classifies each detected eye as either open or closed, while the yawning model determines whether the driver is yawning. The classification results are displayed directly on the video frame using coloured labels, where green indicates a normal condition and red indicates a fatigue-related event. When prolonged eye closure or yawning is detected, the system activates an audible warning alarm to alert the driver. The detected eye regions are classified into Open or Closed states using the trained eye CNN model. Simultaneously, the detected facial image is classified into Yawn or No Yawn using the second CNN model. Both prediction results are displayed on the user interface together with their corresponding confidence scores. These predictions are continuously updated for every captured video frame to enable real-time driver monitoring.

The outputs generated by the eye state classifier and yawning classifier are combined to determine the driver's fatigue condition. If prolonged eye closure or continuous yawning is detected beyond a predefined threshold, the system generates an audible alarm and displays a warning message on the screen. This mechanism helps notify the driver immediately, reducing the possibility of fatigue-related accidents.

The system was developed and implemented in Python 3.11, utilizing TensorFlow 2.16, Keras, OpenCV 4.9, and NumPy, with Visual Studio Code serving as the primary development environment. Live video input was captured using a standard webcam for both model testing and real-time evaluation. Performance of the trained CNN models was assessed through standard metrics including accuracy, precision, recall, and F1-score. Additionally, real-time experiments were carried out under typical indoor lighting conditions to verify the overall functionality and practicality of the driver drowsiness detection system.

D. Performance Study

The performance of the proposed driver drowsiness detection system was evaluated in terms of accuracy, loss, precision, recall, F1-score, and the confusion matrix. These parameters provide a comprehensive assessment of the classification performance of the proposed CNN models.

Accuracy measures the proportion of correctly classified samples among all testing samples and is defined as:

$$Accuracy = \frac{TN+TN}{TP+TN+FP+FN} \quad (1)$$

where *TP, TN, FP*, and *FN* denote True Positive, True Negative, False Positive, and False Negative, respectively [23].

Loss represents the error between the predicted output and the ground truth. Since this study involves binary classification, the Binary Cross-Entropy (BCE) loss function was employed.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(P_i) + 1 - y_i)\log(1 - p_i)] \quad (2)$$

where y_i is the ground-truth label, p_i is the predicted probability, and N is the total number of testing samples [23].

Precision shows the proportion of correctly predicted positive samples among all predicted positive samples which is calculated as follows:

$$Precision = \frac{TP}{TP+TN} \quad (3)$$

Recall determines the proportion of actual positive samples correctly identified by the classifier.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

F1-score is the harmonic mean of precision and recall, providing a balanced evaluation of the classifier performance and it is determined as follows:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (5)$$

The confusion matrix was also used to visualize the classification performance by presenting the numbers of true positives, true negatives, false positives, and false negatives, allowing a more detailed analysis of the prediction results [22].

IV. Results and Discussion

The performance of the proposed system that utilized MobileNetV2 is compared with VGG16 because it is a well-established CNN baseline with substantially greater computational demand [12]. The comparison is intended to evaluate classification performance within the present dataset and training configuration.

As illustrated in Fig. 6, MobileNetV2 demonstrates superior performance and faster convergence, starting at a high accuracy of approximately 78% training and 85% validation and rapidly reaching a stable plateau between 95% and 98% by epoch 5. In contrast, VGG16 starts much lower (around 51% training and 63% validation) but shows a steady, gradual improvement throughout the training process, eventually climbing to between 90% and 94% accuracy after epoch 12.5.

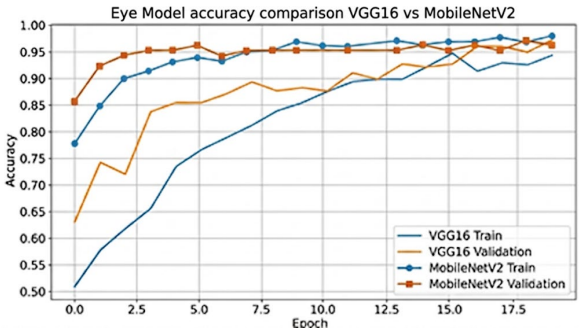

Fig. 6. Eye state model accuracy.

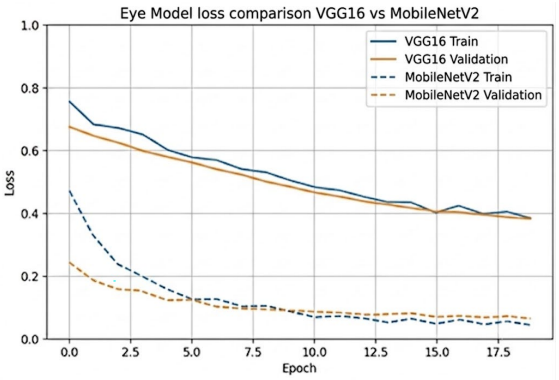

Fig. 7. Eye state model loss.

Fig. 7 shows that MobileNetV2 reduced both training and validation loss more rapidly and reached a substantially lower final loss than VGG16. It begins with a training loss of approximately 0.47 and a validation loss of 0.24, both of which sharply decline within the first 5 epochs before plateauing below 0.10 for the remainder of training. In comparison, the VGG16 model begins with a much higher loss starting around 0.76 for training and 0.68 for validation and exhibits a much slower, gradual decline, only reaching a loss value of roughly 0.38 by epoch 19.

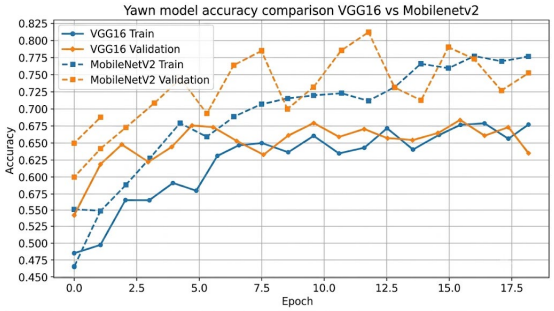

Fig. 8. Yawning model accuracy.

Fig. 8 depicts that MobileNetV2 significantly outperforms VGG16, initiating at higher baselines of 55% training and 65% validation. It also shows a trajectory that peaks at a validation accuracy of over 81% around epoch 12 before settling in the range between 74% and 78%. The VGG16 model shows training accuracy of approximately 46% and a validation accuracy of 54%, steadily climbing to peak just under 68% accuracy by the final epoch.

As illustrated in Fig. 9, MobileNetV2 demonstrates a more efficient optimization curve, starting with a training loss near 0.78 and a validation loss of 0.60, before rapidly dropping within the first few epochs and continuing a steady descent down to a minimum loss of approximately 0.42 to 0.45. Meanwhile, the VGG16 model begins with a training loss of 0.73 and validation loss around 0.69, showing a much slower, gradual convergence rate that hovers significantly higher throughout, only minimizing down to roughly 0.60 to 0.62 by the final epoch. MobileNetV2 consistently maintains lower error rates, making it the better-performing architecture for this task.

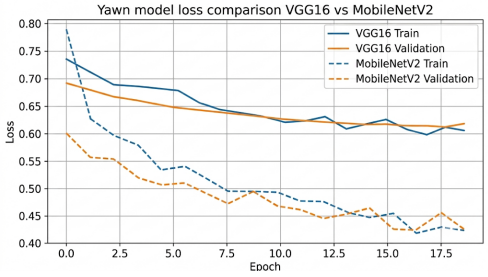

Fig. 9. Yawning model loss.

TABLE I. PERFORMANCE ANALYSIS FOR EYE STATE MODEL

CNN Model	Accuracy (%)	Loss	Precision	Recall	F1-Score
MobileNetV2	98.17	0.0494	0.98	0.98	0.98
VGG16	88.07	0.3870	0.88	0.88	0.88

Table I shows the performance comparison of MobileNetV2 and VGG16 models for eye state. It can be clearly seen that MobileNetV2 significantly outperforms VGG16. MobileNetV2 offers a 10% accuracy improvement, together with substantially lower loss and higher precision, recall and F1-score, indicating that MobileNetV2 is the stronger architecture for this eye state dataset.

TABLE II. PERFORMANCE ANALYSIS FOR YAWNING MODEL

CNN Model	Accuracy (%)	Loss	Precision	Recall	F1-Score
MobileNetV2	78.60	0.4496	0.83	0.78	0.78
VGG16	66.51	0.6111	0.69	0.66	0.65

Similarly, as tabulated in Table II, MobileNetV2 demonstrated superior performance in the yawning model. The proposed model achieved an overall accuracy of 78.60%, while VGG16 achieved only 66.51%. MobileNetV2 also produced higher precision (0.83), recall (0.78), and F1-score (0.78), compared to VGG16, which recorded values of 0.69, 0.66, and 0.65, respectively.

The performance gap between eye-state classification and yawning detection stems from three domain-specific computer vision challenges:

- Eye state classification relies on binary visual features within a tight, highly localized ROI. Conversely, yawning involves complex spatial dynamics across the full face.
- Yawning ROIs contain confounding facial expressions such as talking, singing, smiling, or speaking, which closely resemble early-stage yawning gestures.

- Full face images in the yawning dataset incorporate background distractors such as beards, vehicle headrests, seatbelts, and shadows as well as high intra-class variance.

Overall, the experimental results indicate that MobileNetV2 consistently outperformed VGG16 for both classification tasks. The lightweight architecture of MobileNetV2, together with its depthwise separable convolution mechanism, enables efficient feature extraction while maintaining high classification accuracy, making it more suitable for real-time driver drowsiness detection.

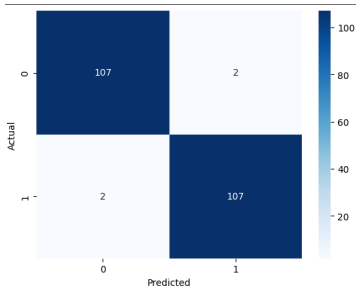

Fig. 10. Confusion matrix for eye state model using MobileNetV2.

The confusion matrix in Fig. 10 shows that MobileNetV2 correctly classified 214 out of 218 testing samples, with only four misclassifications. Specifically, 107 closed-eye images and 107 open-eye images were correctly recognized, while only two samples from each class were incorrectly classified.

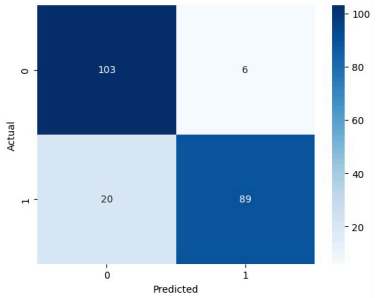

Fig. 11. Confusion matrix for eye state model using VGG16.

On the other hand, as shown in Fig. 11, the VGG16 model correctly classified 192 out of 218 testing samples, with 103 closed-eye images and 89 open-eye images correctly identified. However, six closed-eye images were incorrectly classified as open, while 20 open-eye images were misclassified as closed. These classification errors indicate that VGG16 had difficulty distinguishing certain eye states, particularly open-eye images. These results confirm the robustness and reliability of MobileNetV2 for eye-state classification.

The confusion matrix in Fig. 12 shows that the MobileNetV2 model correctly classified 169 out of 215 testing samples, where 105 non-yawning images and 64 yawning images were correctly identified. However, four non-yawning images were incorrectly classified as yawning, while 42 yawning images were misclassified as non-yawning.

Compared to the VGG16 model indicated in Fig. 13, VGG16 correctly classified 143 out of 215 testing samples, where 94 non-yawning images and 49 yawning images were correctly identified. However, 15 non-yawning images were incorrectly classified as yawning, while 57 yawning images were misclassified as non-yawning. These classification errors indicate that VGG16 had difficulty distinguishing yawning behaviour, particularly actual yawning events.

The Haar Cascade algorithm was employed as the first stage of the proposed driver drowsiness detection system to detect the driver's face and localize the eye region before classification using the MobileNetV2 model. By extracting only the facial ROI, unnecessary background information was eliminated, allowing the CNN classifier to focus on the most relevant facial features while reducing computational complexity.

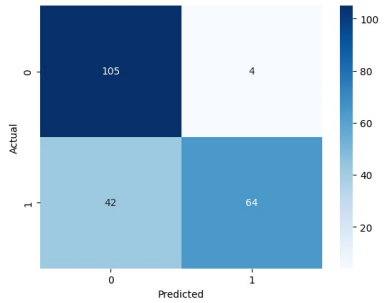

Fig. 12. Confusion matrix for yawning model using MobileNetV2.

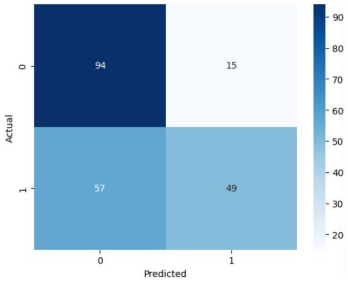

Fig. 13. Confusion matrix for yawning model using VGG16.

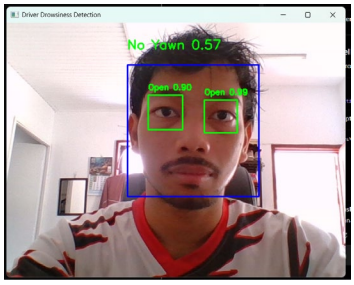

Fig. 14. Normal driving condition.

Fig. 14 shows the successful detection of the driver's face and eyes under normal driving conditions. The face was enclosed using a blue bounding box, while the detected eye regions were highlighted using green bounding boxes. The extracted eye regions were subsequently passed to the MobileNetV2 classifier to determine whether the driver's eyes were open or closed.

As can be seen in Fig. 15, the Haar Cascade detector successfully localized the driver's face, while MobileNetV2 classified both eyes as closed. To distinguish normal eye

blinking from actual driver drowsiness, the system only activates the "Drowsiness Detected" warning when both eyes remain continuously closed for more than two seconds.

In order to reduce false alarms caused by normal physiological actions such as blinking, the system does not generate an alert based on a single-frame prediction. Instead, consecutive predictions are monitored over a short period of time before an alert is activated. For yawning, a fatigue warning is triggered when a yawn is detected continuously for approximately 1.5 seconds, or when two or more yawning events occur within a 60-second period. For eye closure, a critical drowsiness alert is triggered when the eyes remain continuously closed for approximately 2 seconds. These temporal thresholds help distinguish normal blinking and brief facial movements from sustained signs of driver fatigue or drowsiness.

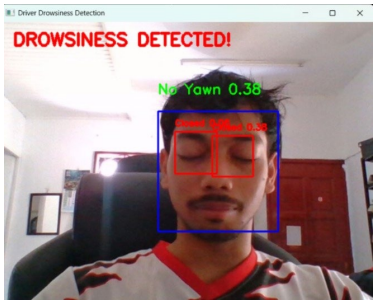

Fig. 15. Closed eye condition.

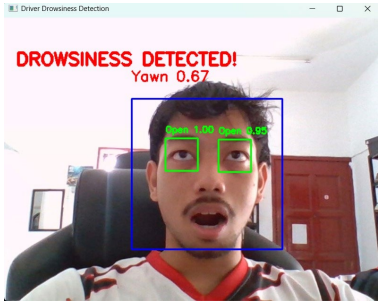

Fig. 16. Yawning condition.

Fig. 16 illustrates the proposed system detecting a yawning event. Although Haar Cascade only performs face localization, it successfully identified the driver's facial region, enabling the MobileNetV2 yawning classifier to analyse the cropped face and classify the driver's condition as "Yawn". Once a yawning event was detected, the system generated a visual warning indicating potential driver fatigue.

The experimental results demonstrate that Haar Cascade provided stable and reliable face localization under normal indoor lighting conditions. The algorithm operated efficiently in real time and served as an effective preprocessing stage before CNN-based classification. Its lightweight implementation contributed to faster processing speed by reducing the amount of image data analysed by the MobileNetV2 models.

V. Conclusion and Future Work Recommendation

This paper presented an effective, lightweight proof-of-concept framework for real-time driver monitoring under laboratory webcam conditions by integrating the Haar Cascade algorithm with the MobileNetV2 Convolutional Neural Network. Haar Cascade was employed for face detection and facial ROI extraction, while MobileNetV2 classified eye states and yawning behaviour. Experimental results demonstrated that MobileNetV2 outperformed VGG16 while the integration of Haar Cascade and MobileNetV2 provides an effective, lightweight, and practical solution for real-time driver drowsiness detection, demonstrating strong potential for future intelligent transportation and Advanced Driver Assistance Systems (ADAS).

Future work will focus on testing under different conditions such as low lighting, different head positions, glasses, and partial face occlusion. Although the proposed architecture was designed to be computationally efficient, this study did not perform a comprehensive hardware-level efficiency evaluation. Computational metrics such as inference latency, FPS, peak memory consumption, and energy usage may vary depending on the deployment platform and software configuration. Future work will therefore benchmark the proposed method on representative CPU, GPU, and edge-computing platforms to establish its suitability for real-time deployment.

Acknowledgment

The authors would like to acknowledge the Faculty of Electrical Engineering, Universiti Teknologi MARA for the support in conducting the project.

References

[1] World Health Organization, “Road traffic injuries,” WHO. Accessed: Feb. 13, 2026. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[2] Ministry of Transport Malaysia, “Malaysia road fatalities index,” 2019. Accessed: Jun. 18, 2025. [Online]. Available: https://www.mot.gov.my/en/land/safety/malaysia-road-fatalities-index

[3] J. M. Owens, “Prevalence of drowsy driving crashes: Estimates from a large-scale naturalistic driving study,” AAA Foundation for Traffic Safety, Washington, DC, USA, 2018. Accessed: Feb. 23, 2023. [Online]. Available: https://aaafoundation.org/prevalence-drowsy-driving-crashes-estimates-large-scale-naturalistic-driving-study/

[4] X. Li, J. Xia, L. Cao, G. Zhang, and X. Feng, “Driver fatigue detection based on convolutional neural network and face alignment for edge computing device,” *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 235, no. 10–11, pp. 2699–2711, Sep. 2021, doi: 10.1177/0954407021999485.

[5] I. A. Aboagye, W. Owusu-Banahene, K. Amexo, K. A. Boakye-Yiadom, R. A. Sowah, and N. L. Sowah, “Design and development of computer vision-based driver fatigue detection and alert system,” in *Proc. IEEE Int. Conf. Adapt. Sci. Technol. (ICAST)*, 2021, doi: 10.1109/ICAST52759.2021.9681943.

[6] J. H. Yousif, “Fuzzy logic approach for detecting drivers’ drowsiness based on image processing and video streaming,” *TechRxiv*, preprint, Nov. 2021, doi: 10.36227/techrxiv.16777927.v2.

[7] S. Jothimani, K. Madhumitha, V. Narmadha, and R. Naveena, “Driver fatigue prediction with real time driver alert system using deep learning,” in *Proc. Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC)*, 2022, pp. 462–465, doi: 10.1109/ICAAIC53929.2022.9792677.

[8] S. Zhao, Y. Peng, Y. Wang, G. Li, and M. Al-Mahbashi, “Lightweight YOLOM-Net for automatic identification and real-time detection of fatigue driving,” *Comput., Mater. Continua*, vol. 82, no. 3, pp. 4995–5017, 2025, doi: 10.32604/cmc.2025.059972.

[9] M. Vimala, M. Nandhini, S. Jasmine, P. Ranjith Kumar, and S. Ramasamy, “Implementation of a lightweight deep learning model for

detecting driver fatigue," *Franklin Open*, vol. 15, Art. no. 100547, Jun. 2026, doi: 10.1016/j.fraope.2026.100547.

[10] Z. Liu, L. Zhang, X. Wang, and G. Yao, "Research on subway driver behavior recognition method based on multi-scale feature fusion CNN," in *Proc. 5th Int. Conf. Artif. Intell., Big Data Algorithms (CAIBDA)*, Beijing, China, 2025, pp. 800–804, doi: 10.1109/CAIBDA65784.2025.11182629.

[11] M. Kara, "What is the function of the pooling layer in a convolutional neural network (CNN)?" *Medium*, Oct. 19, 2023. Accessed: Feb. 13, 2026. [Online]. Available: https://markmkara.medium.com/what-is-the-function-of-the-pooling-layer-in-a-convolutional-neural-network-cnn-ae12641682b1

[12] P. Kashyap, "Understanding dropout in deep learning: A guide to reducing overfitting," *Medium*, Oct. 30, 2024. Accessed: Feb. 13, 2026. [Online]. Available: https://medium.com/@piyushkashyap045/understanding-dropout-in-deep-learning-a-guide-to-reducing-overfitting-26cbb68d5575

[13] S. Balaji, "Binary image classifier CNN using TensorFlow," Medium, Aug. 29, 2020. Accessed: Jun. 7, 2023. [Online]. Available: https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697

[14] M. Rezaei, "Samples of closed-eyes from our IntelliEye dataset," *ResearchGate*, 2016. Accessed: Jun. 18, 2025. [Online]. Available: https://www.researchgate.net/figure/Samples-of-closed-eyes-from-our-IntelliEye-dataset_fig10_311825380

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015, doi: 10.48550/arXiv.1409.1556.

[16] "What is deep learning?" IBM. Accessed: Feb. 13, 2026. [Online]. Available: https://www.ibm.com/think/topics/deep-learning

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, 2012, pp. 1097–1105. [Online]. Available: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

[18] J. Mahadeokar and G. Pesavento, "Open sourcing a deep learning solution for detecting NSFW images," *Yahoo Engineering*, Sep. 30, 2016. [Online]. Available: https://yahooeng.tumblr.com/post/151148689421/open-sourcing-a-deep-learning-solution-for

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR*), Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, *"MobileNetV2: inverted residuals and linear bottlenecks," in Proc. 2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.*

[21] S. Raju, "yawn_eye_dataset_new," Kaggle, Jul. 7, 2020. [Online]. Available: https://www.kaggle.com/datasets/serenaraju/yawn-eye-dataset-new

[22] I. Luwani et al., "A hybrid edge-AI framework for low-cost driver drowsiness detection using EAR and MobileNetV2 in Nigerian driving environments," J. Eng. Logic Model. Res., vol. 12, no. 5, Jun. 2026, doi: 10.70382/sjelmr.v12i5.030.

[23] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.