JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

Development of a Real-Time Smartphone Based Drowsiness Detection System Using MobileNet and Google ML Kit

Ade Zulkarnain Hasibuan[1*], Munjiat Setiani Asih[2]
[1*]Samudra University, Langsa, Indonesia
[2]Harapan Medan University, Medan, Indonesia
[1*]adezulhsb@unsam.ac.id, [2]munjiat.stth@gmail.com
[*)] adezulhsb@unsam.ac.id

Abstract-The rapid development of smartphone technology has increased the duration of device usage, which can lead to eye fatigue, decreased concentration, and drowsiness. Many users continue using smartphones despite experiencing signs of fatigue, potentially affecting health and productivity. Therefore, this study aims to develop a real-time drowsiness detection system for smartphone users based on Android using Google ML Kit and MobileNetV1. The system utilizes Google ML Kit for face detection and MobileNetV1 for classifying eye conditions into open and closed states. The dataset used is a combination of a public dataset from Kaggle (dataset_B_Eye_Images) and additional data collected independently to improve model generalization. The model was trained and further optimized through a fine-tuning process. Experimental results show that the model achieved an accuracy of approximately 96%, with balanced precision, recall, and F1-score values. The confusion matrix analysis indicates improved performance after fine-tuning. In real-time implementation, the system operates at 6.0–7.3 FPS with a latency of 60–72 ms per frame and a notification response time of less than 1 second. The system demonstrates robustness under varying lighting conditions, achieving accuracy up to 100% in bright conditions, 98% in normal conditions, and 95% in low-light conditions. However, performance decreases when users wear glasses due to reflection interference. Overall, the results indicate that MobileNetV1 is effective for real-time drowsiness detection on mobile devices, although further improvements are needed to enhance system robustness under diverse user conditions.

Keywords: Drowsiness Detection, MobileNetV1, Google ML Kit, Real-Time System, Android Application

1. INTRODUCTION

The rapid development of smartphone technology has made these devices an essential part of daily life, supporting activities such as communication, entertainment, learning, and work. The high intensity of smartphone usage over prolonged periods can lead to eye fatigue, decreased concentration, and drowsiness due to continuous exposure to screens. Many users tend to continue using their smartphones despite experiencing signs of fatigue, such as drooping eyes, slow blinking, head nodding, and frequent yawning. These conditions can negatively affect eye health, reduce productivity, and increase the risk of accidents, particularly when smartphones are used while walking or driving [1][2]. Therefore, the development of an automatic and real-time drowsiness detection system for smartphone users has become increasingly essential.

Drowsiness detection is an important topic in the fields of computer vision and deep learning. Previous studies have primarily focused on the context of vehicle drivers to reduce the risk of accidents caused by fatigue. Drowsiness detection systems generally utilize visual indicators such as eye state (open or closed), blink frequency, eye closure duration (PERCLOS), yawning, and head position. Visual-based approaches are considered more practical than physiological methods such as EEG, ECG, EOG, and EMG, as they do not require additional sensors and are easier to implement on common devices such as smartphone cameras [3][4].

Several studies have successfully developed image-based drowsiness detection systems in real time. The study entitled “Driver Drowsiness Detection in Real-time” utilized a smartphone camera to detect eye states (open and closed) in real time using a MobileNet-based model, achieving an accuracy of over 98%. This study demonstrates that smartphone cameras can serve as an effective and practical medium for detecting drowsiness without requiring additional hardware [2]. In addition, the study “Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN” implemented a system pipeline consisting of face detection, extraction of eye and mouth regions, condition classification, and automatic fatigue decision-making, achieving an accuracy of 93.623%. This approach indicates that combining multiple visual indicators can improve the reliability of drowsiness detection systems [5].

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

In the development of real-time systems on mobile devices, the selection of lightweight deep learning models is a crucial factor. Conventional CNN models such as VGG16, ResNet50, and InceptionV3 offer high accuracy; however, they require substantial computational resources, making them less suitable for implementation on smartphones. Previous studies have shown that lightweight models such as MobileNet can provide a favorable trade-off between accuracy and inference speed. The study entitled "Comparing MobileNet-SSD and YOLOv3 Learning Architecture for Real-time Driver's Fatigue Detection" reported that MobileNet-SSD achieved an inference time of approximately 135.5 ms, which is significantly faster than YOLOv3 at 497.51 ms, with only a minor reduction in accuracy [6]. Another study comparing several lightweight CNN models found that MobileNetV3 Small achieved an accuracy of 94.69% with a latency of 8 ms and a model size of only 9.9 MB, making it highly suitable for real-time implementation on mobile devices [7].

One of the widely used lightweight CNN architectures is MobileNet. MobileNetV1 employs depthwise separable convolution to reduce the number of parameters and computational cost without significantly compromising performance. The study entitled "MobileNet-Based Architecture for Distracted Human Driver Detection of Autonomous Cars" demonstrated that MobileNetV1 is capable of performing predictions with an inference time of approximately 0.01 seconds per image [8]. Furthermore, the study "TensorRT Optimized Driver Drowsiness Detection System Using Edge Device" reported that MobileNetV1 achieved an accuracy of 98.27% with high performance when optimized on edge devices [1]. In addition, the study "Efficient and Robust Driver Fatigue Detection Framework Based on the Visual Analysis of Eye States" utilized MobileNetV1 as an eyes-CNN model to detect eye conditions, achieving an accuracy of 98.28% and a processing speed of 38 FPS [3].

In addition to eye condition analysis, other indicators such as yawning are also widely used as markers of drowsiness. The study entitled "Drowsiness Detection Based on Yawning Using Modified Pre-trained Model MobileNetV2 and ResNet50" demonstrated that MobileNetV2 is capable of detecting yawning in real time with an accuracy of 98%, while ResNet50 achieved an accuracy of 99% [9]. This finding indicates that facial expression analysis can serve as an additional indicator to improve the accuracy of drowsiness detection systems.

Several studies have also developed more advanced approaches by incorporating temporal and multimodal information. The study entitled "Fatigue Driving Recognition Network" combined Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to exploit temporal information across frames, achieving a validation accuracy of 98.96% [10]. Other studies have employed multimodal approaches based on facial images and physiological signals to improve accuracy, achieving up to 98.41% [11]. However, these approaches require higher computational resources, making them less suitable for real-time implementation on smartphones.

Despite the extensive research conducted in this field, most studies have focused on vehicle drivers and have been implemented on computers, GPU servers, or edge devices. Direct implementation on Android smartphones remains relatively limited. Furthermore, several studies still rely on face detection methods such as Haar Cascade or MTCNN, which exhibit limitations under low-light conditions, varying face angles, or higher computational demands [12][5]. Therefore, this study utilizes Google ML Kit for real-time face detection on Android devices, as this framework is more lightweight and stable for mobile environments.

MobileNetV1 is not only known for its lightweight architecture but also for its high computational efficiency due to the use of depthwise separable convolution (DSC). This technique separates the convolution process into depthwise convolution and pointwise convolution, thereby significantly reducing the number of parameters and computational cost compared to standard convolution. This approach makes MobileNetV1 highly suitable for implementation on mobile and embedded devices with limited computational and memory resources[13].

Other studies have also shown that MobileNet-based lightweight CNN provide a good trade-off between accuracy and system efficiency. In mobile vision implementations, the use of overly complex models may improve accuracy, but it also increases latency, memory usage, and inference time. Therefore, lightweight models are considered more effective for real-time smartphone systems because they are capable of delivering sufficiently high performance with lower resource consumption compared to conventional CNN[14][15]. This approach is particularly important for Android-based applications that must simultaneously perform camera processing, face detection, image preprocessing, and model inference in real time.

Furthermore, several recent studies have proposed modifications to the MobileNetV1 architecture to improve feature extraction quality without significantly increasing model complexity. One of the approaches involves combining MobileNetV1 with bottleneck layers and residual transitions to enhance the model's ability to recognize visual features while maintaining a lightweight parameter structure. The results of these studies indicate that lightweight CNN approaches are still capable of delivering competitive classification performance while preserving computational efficiency on devices with limited resources[13]. This demonstrates that MobileNetV1

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

remains one of the most relevant and effective architectures for the development of real-time computer vision systems on mobile devices.

In addition to model architecture, the use of transfer learning and fine-tuning has also been proven to improve the performance of lightweight CNN models. Several studies have shown that the use of pretrained MobileNetV1 combined with a fine-tuning process can enhance the model's generalization capability and improve classification performance across various image datasets[16] [17]. This approach is important in computer vision-based research because the model can leverage previously learned feature representations, making the training process more efficient and stable.

On the other hand, various studies have shown that the performance of mobile-based computer vision systems is highly influenced by real-world environmental conditions, such as lighting variations, image quality, and objects experiencing occlusion or certain visual disturbances. Therefore, real-world testing becomes an important aspect in evaluating the robustness of the developed system[15][13]. In the context of smartphone-based drowsiness detection, these challenges become even more complex because the system must be capable of detecting the user's eye condition in real time under various dynamic usage conditions.

Based on the aforementioned problems, this study aims to develop a real-time drowsiness detection system for smartphone users on the Android platform by utilizing Google ML Kit for face detection and MobileNetV1 for classifying drowsiness conditions based on facial images or eye regions. MobileNetV1 is selected due to its lightweight architecture, fast inference speed, and suitability for devices with limited computational resources. The proposed system is expected to operate in real time with high accuracy and low latency. The system performance is evaluated using metrics such as accuracy, precision, recall, and F1-score, as well as real-time performance indicators including FPS and latency. This study has been tested on a limited scale involving smartphone users to evaluate system performance, accuracy level, and user response toward the developed application. The testing results indicate that the system is capable of operating properly according to the designed functionalities and provides a sufficiently optimal user experience on smartphone devices. This study is expected to provide a preventive solution to help smartphone users reduce device usage when experiencing drowsiness and to serve as a reference for the development of computer vision-based digital health applications.[18][19][20].

2. RESEARCH METHODOLOGY

The developed system integrates Google ML Kit for face detection and MobileNetV1 for classifying drowsiness conditions based on facial images or eye regions. The research process consists of dataset collection, data preprocessing, model training, system implementation on Android devices, and comprehensive evaluation of both model performance and real-time system performance.

2.1 Dataset Collection

The dataset used in this study is a combination of a public dataset and a self-collected dataset. The public dataset was obtained from the Kaggle platform, namely the dataset_B_Eye_Images, which contains eye images categorized into two classes:

1. Open Eyes
2. Closed Eyes

This dataset was selected because its structure aligns well with the research objective of detecting drowsiness based on eye conditions. To improve the model's generalization capability in real-world application scenarios, this study also incorporates an additional dataset collected using an Android smartphone camera. The data were collected from:

1. The researcher;
2. Several other participants.

The data collection process was conducted under various conditions, including:

1. Bright lighting conditions;
2. Low-light conditions;
3. Users with and without glasses;
4. Different head positions (frontal, downward, and slightly tilted).

The combination of the public dataset and the self-collected dataset aims to enhance the robustness of the model in handling real-world usage conditions. After data collection, the dataset was divided into two subsets:

1. Training set = 80%
2. Validation set = 20%

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

.

The data were randomly split to ensure a balanced class distribution.

2.2 Image Preprocessing and Data Augmentation

Before being used in the training process, the image data underwent a preprocessing stage to align with the model input requirements and improve data quality. The preprocessing steps include:

1. Image Resizing: All images were resized to 224 × 224 pixels to match the input size required by the MobileNetV1 model.
2. Normalization: Pixel values were normalized to the range of 0–1 to ensure more stable training and faster convergence.
3. Data Labeling: Each image was assigned a label based on its category:
   a. 0 = closed eyes
   b. 1 = open eyes
4. Eye Region Cropping: For the self-collected dataset, the eye region was cropped to ensure that the model focuses on relevant features.

   To enrich data variability and reduce the risk of overfitting, data augmentation techniques were applied, including:
   a. Rotation
   b. Horizontal flip
   c. Zoom
   d. Brightness adjustment
   e. Width shift
   f. Height shift

Data augmentation was applied automatically during the training process.

2.3 Training of the MobileNetV1 Model

The classification model used in this study is MobileNetV1, as it has a small model size and enables fast inference on mobile devices. The model architecture consists of:

1. MobileNetV1 base model as a feature extractor;
2. Global Average Pooling layer;
3. Dense layer;
4. Dropout layer to prevent overfitting;
5. Output layer with Softmax activation for binary classification.

The model training parameters are as follows:

1. Optimizer: Adam
2. Learning rate: 0.0001
3. Batch size: 32
4. Number of epochs: 20

The model training and evaluation processes were conducted using Python with the following libraries:

1. TensorFlow
2. Keras
3. Scikit-learn

2.4 Evaluation of the Classification Model

Model evaluation was conducted in the Python environment using testing data. The model performance was assessed using a confusion matrix with the following metrics:

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

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall} \quad (4)$$

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

2.5 Implementation of the System on Android Devices

After the model achieved satisfactory performance, it was converted into TensorFlow Lite (.tflite) format to enable deployment on the Android application. The system implementation process consists of the following stages:

1. The smartphone camera captures frames in real time;
2. Google ML Kit detects the user's face;
3. The face or eye region is cropped;
4. The cropped image is fed into the MobileNetV1 model;
5. The model predicts the eye condition;
6. If the eyes remain closed for several consecutive frames, the application triggers a notification or alert.

System workflow:

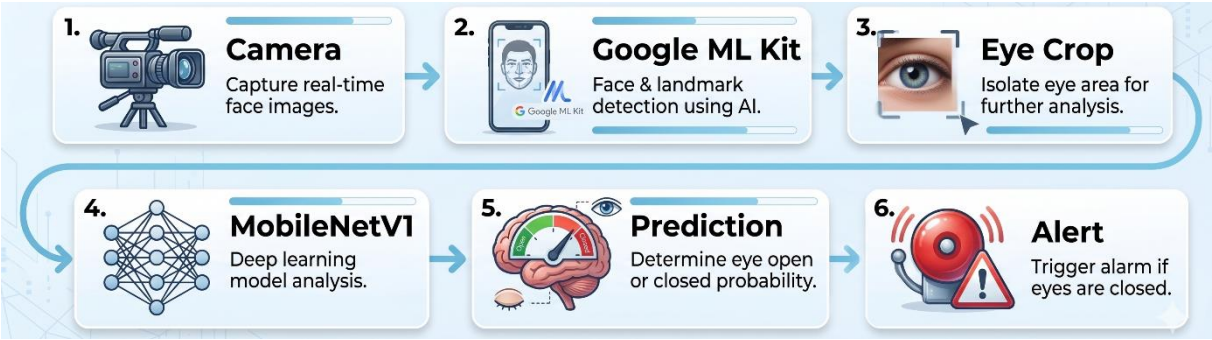

Figure 1. Overall System Workflow

2.6 Evaluation of the Real-Time System

After the system was implemented on Android, real-time performance evaluation was conducted, including:

1. FPS (Frame Per Second): to measure the number of frames processed per second;
2. Latency: to measure the inference time per frame;
3. Real-world condition testing, including:
   a. Bright lighting;
   b. Low-light conditions;
   c. Users wearing glasses;
   d. Various head positions.

Through this evaluation, the system's capability to operate in real time on Android smartphones can be assessed.

3. RESULT AND DISCUSSION

3.1 Results of Training and Evaluation

The classification model in this study was developed using the MobileNetV1 architecture and trained using a combination of a public dataset from Kaggle (dataset_B_Eye_Images) and a self-collected dataset. The training process was conducted with a data split of 80% for training and 20% for validation. In the initial training phase, the model achieved a training accuracy of 97.03% with a training loss of 0.0764. Meanwhile, during validation, the model obtained a validation accuracy of 95.45% with a validation loss of 0.1106. The relatively small gap between training and validation accuracy indicates that the model does not suffer from overfitting and demonstrates good generalization capability.

To further improve the model performance, a fine-tuning process was performed by unfreezing several layers of MobileNetV1 and retraining the model using a smaller learning rate. The results of fine-tuning show an improvement in model performance, with the validation accuracy increasing to 96.18% and a validation loss of 0.1158. The accuracy improvement of 0.52% indicates that fine-tuning successfully enhanced the model's ability to learn data patterns more effectively. Although the validation loss slightly increased, this remains within an acceptable range and does not indicate performance degradation. In classification tasks, accuracy improvement is often prioritized as it reflects the model's ability to correctly predict unseen data. Further evaluation was conducted using a classification report after the fine-tuning process. The results show that the model achieved an overall accuracy of 96% on the test data.

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

For the closed-eye (Closed) class, the model achieved a precision of 97%, recall of 96%, and F1-score of 96%. Meanwhile, for the open-eye (Open) class, the model achieved a precision of 96%, recall of 97%, and F1-score of 96%. The high precision and recall values for both classes indicate that the model has strong capability in distinguishing between open and closed eye conditions. Furthermore, the balanced F1-scores across both classes demonstrate that the model performs consistently and is not biased toward any particular class. Therefore, the fine-tuned model exhibits more optimal performance compared to the model before fine-tuning.

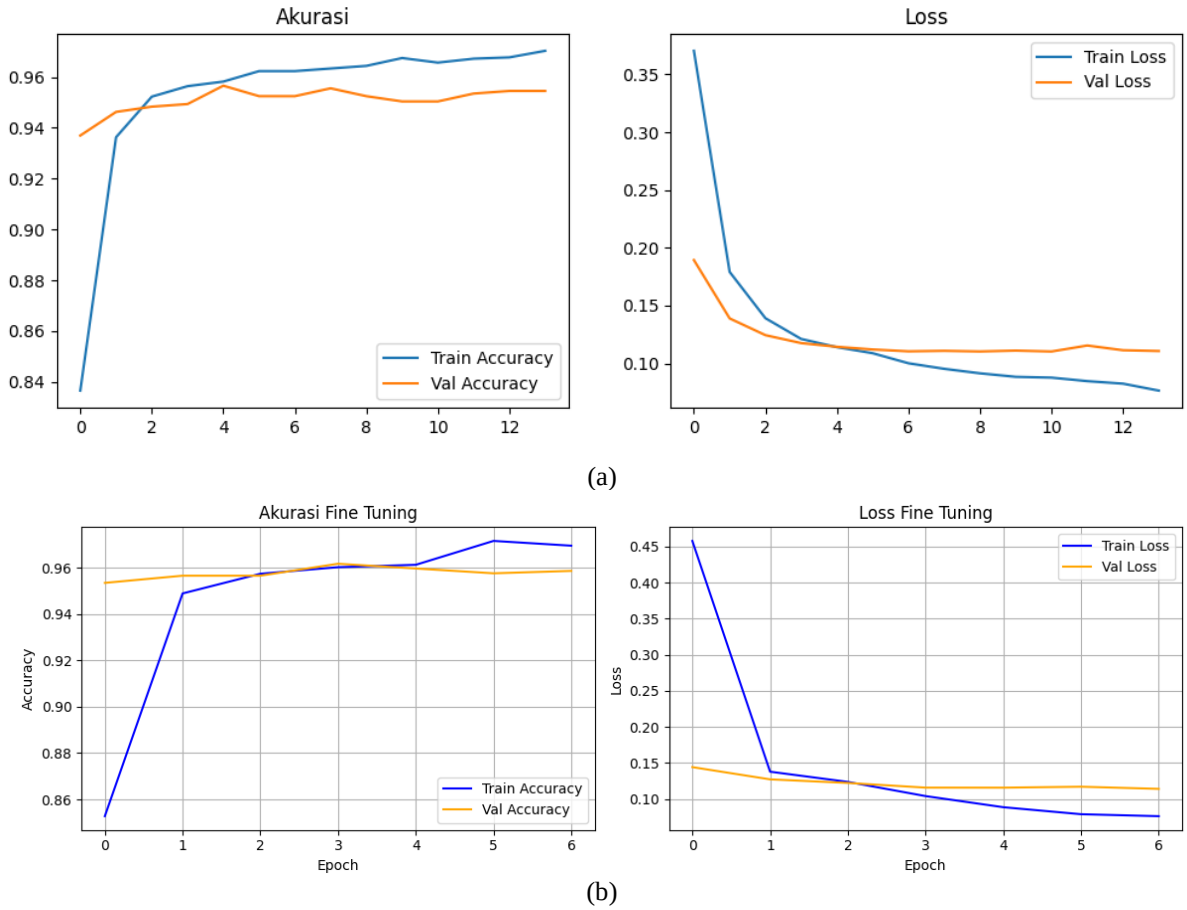

Figure 2. (a) Training and Validation Curves Before Fine Tuning (b) Training and Validation Curves After Fine Tuning

3.1 Analysis of the Confusion Matrix

Model performance evaluation was conducted using a confusion matrix to analyze the distribution of model predictions across each class, both before and after the fine-tuning process. This analysis aims to identify the types of classification errors and to assess improvements in model performance more comprehensively. Based on the results before fine-tuning, the model demonstrated the following classification distribution: a total of 455 closed-eye (Closed) instances were correctly predicted, while 21 closed-eye instances were misclassified as open (Open). For the open-eye class, 465 instances were correctly classified, whereas 27 instances were misclassified as closed. These results indicate that although the model already exhibits good performance, there are still notable classification errors, particularly in cases where open-eye conditions are incorrectly predicted as closed.

After the fine-tuning process, an improvement in model performance was observed, as reflected by changes in the confusion matrix. The number of correct predictions for the closed-eye class remained at 455, with the number of misclassifications unchanged at 21 instances. This indicates that the model's ability to detect closed-eye conditions was already optimal during the initial training phase and remained stable after fine-tuning. Meanwhile, for the open-eye class, there was an increase in correct predictions from 465 to 476 instances, along with a reduction in misclassification from 27 to 16 instances. This decrease in classification errors for the open-eye class indicates that the model has improved in distinguishing between open-eye conditions and cases where

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

the eyes are nearly closed or ambiguous. Quantitatively, the number of false negatives decreased by 11 instances, or approximately 40%. This represents a significant improvement, as such classification errors can affect the reliability of the system in distinguishing between normal and drowsy conditions.

In the context of a drowsiness detection system, there are two main types of classification errors: false positives and false negatives. A false positive occurs when a normal condition is incorrectly classified as drowsy, whereas a false negative occurs when a drowsy condition is not detected by the system. Among these two types of errors, false negatives are considered more critical, as they may cause the system to fail in providing warnings to users who are actually in a drowsy state. The evaluation results indicate that the fine-tuning process successfully improved the model's performance by reducing classification errors, particularly in categories that could lead to misinterpretation of the user's condition. Furthermore, the stability of performance in the closed-eye class demonstrates that fine-tuning did not degrade the model's previously strong capability.

Overall, the confusion matrix analysis indicates that the MobileNetV1 model, after undergoing the fine-tuning process, achieves more optimal, stable, and balanced performance in classifying open-eye and closed-eye conditions. With the reduction in classification errors and the increase in correct predictions, the model demonstrates a high level of reliability for implementation in real-time smartphone-based drowsiness detection systems.

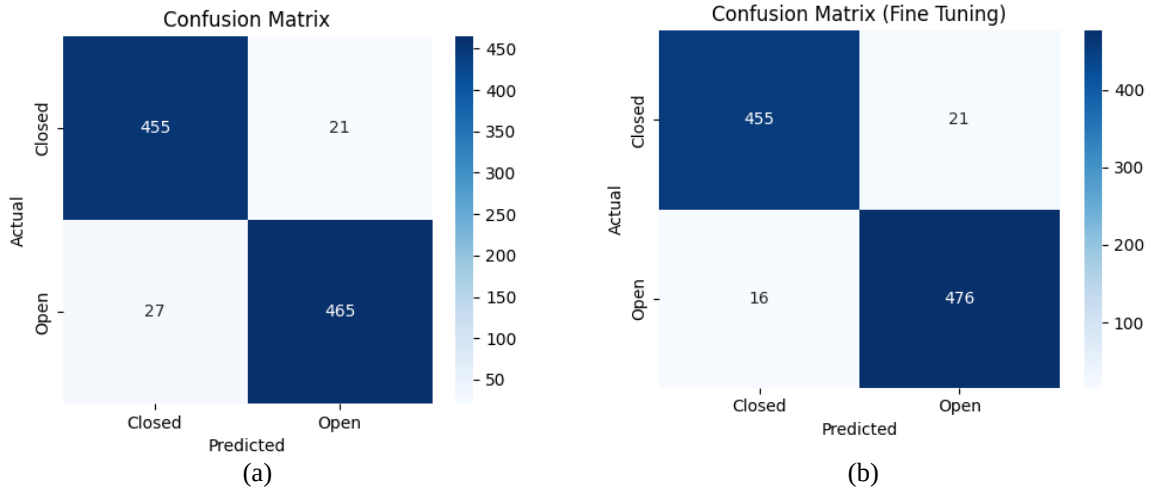

Figure 3. (a) Confusion Matrix Before Fine-Tuning (b) Confusion Matrix After Fine-Tuning of the Proposed Model

3.2 Results of the System Implementation on Android Devices

After the MobileNetV1 model was trained and fine-tuned, the next stage involved deploying the model into an Android-based application using the TensorFlow Lite (.tflite) format. The developed system utilizes the smartphone's front camera to capture frames in real time, followed by face detection using Google ML Kit, eye region cropping, and classification using the MobileNetV1 model. Based on the testing results on Android devices, the real-time system performance was evaluated using several key parameters, including Frame Per Second (FPS), latency per frame, notification response time, and memory usage. The results of the evaluation are presented in Table 1.

Table 1. Real-Time System Evaluation on Android

No	Parameter	Value
1	FPS (Frame per Second)	6.0 – 7.3
2	Latency per Frame	60 ms – 72 ms
3	Notification Response Time	< 1 detik
4	Memory Usage	128 MB

The evaluation results indicate that the system operates at a frame rate ranging from 6.0 to 7.3 frames per second (FPS). This demonstrates that the system is capable of real-time operation, although it still falls within the

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

lower range of real-time performance. This limitation is influenced by the relatively complex computational process, which involves face detection using Google ML Kit as well as the classification of two eye images per frame using the MobileNetV1 model. In terms of inference time, the system achieves a latency of 60 ms to 72 ms per frame. This indicates that the classification process is sufficiently fast and remains within an acceptable range for mobile-based deep learning applications. The relatively stable latency also demonstrates that the MobileNetV1 model can be efficiently implemented on devices with limited computational resources. Furthermore, the system's notification response time when drowsiness is detected is recorded to be less than 1 second. This shows that the system is capable of responding quickly in identifying the user's condition and providing immediate alerts, thereby helping users become aware of their drowsiness in a timely manner.

In terms of resource utilization, the application consumes approximately 128 MB of memory during the detection process. This value remains within an acceptable range for computer vision and deep learning applications on Android devices, considering that the system performs multiple processes simultaneously, including face detection, eye region extraction, and model inference. Overall, the implementation results indicate that the developed drowsiness detection system is capable of operating in real time on Android devices with stable performance. Although the FPS value is relatively lower compared to desktop-based systems, the system is still able to provide accurate detection results with low latency and efficient memory usage. This demonstrates that the combination of MobileNetV1 and Google ML Kit is sufficiently effective for implementation on mobile devices to support real-time drowsiness detection applications.

3.3 Performance Evaluation Under Varying Lighting Conditions

System testing was also conducted to evaluate the effect of lighting conditions on drowsiness detection performance. The evaluation was carried out under three different lighting conditions, namely bright, normal (indoor), and low-light conditions. For each condition, 50 test trials were performed with variations of open-eye and closed-eye states to obtain representative results.

Table 2. Evaluation Under Different Lighting Conditions

No	Condition	Accuracy (%)	Description
1	Bright	100%	Highly stable detection
2	Normal (Indoor)	98%	Stable
3	Low-light	95%	Slight noise

Based on the results presented in Table 2, the system demonstrates excellent performance under bright lighting conditions, achieving an accuracy of 100%. This indicates that under optimal lighting, eye features can be clearly recognized by the system, allowing the classification process to operate at maximum efficiency and stability. Under normal (indoor) lighting conditions, the system achieves an accuracy of 98%, which is still considered very high. This slight decrease is attributed to variations in indoor lighting intensity that may affect image quality, although it does not significantly impact the overall model performance. Meanwhile, under low-light conditions, the system achieves an accuracy of 95%. Although there is a decrease compared to other conditions, the system performance remains within a good category. This reduction is caused by the decreased visual quality of the images, making eye features less distinct and introducing slight noise, which affects the classification process.

Overall, the evaluation results indicate that the system exhibits good robustness against variations in lighting conditions. The system is able to maintain high accuracy across different lighting scenarios, although the best performance is achieved under optimal lighting conditions. This demonstrates that the approach used in this study is sufficiently effective for real-world implementation, where environmental conditions may vary.

3.4 Performance Evaluation Based on User Conditions

Further testing was conducted to evaluate the effect of user characteristics on system performance, particularly the use of glasses. The evaluation was divided into two conditions: users without glasses and users wearing glasses. For each condition, 50 test trials were performed with variations of open-eye and closed-eye states.

Table 3. Evaluation Under Different User Conditions

No	Condition	Accuracy (%)	Description
1	Without Glasses	100%	Excellent performance
2	With Glasses	50%	Affected by reflections on the glasses

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

Based on the results presented in Table 3, the system demonstrates excellent performance for users without glasses, achieving an accuracy of 100%. This indicates that the system is able to detect eye conditions with high accuracy when there are no visual obstructions in the eye region. However, for users wearing glasses, a significant decrease in performance is observed, with an accuracy of 50%. This reduction is primarily caused by light reflections on the surface of the glasses, which interfere with the detection of eye features. These reflections prevent the eye region from being optimally detected, thereby affecting the classification results produced by the model. In addition, the use of glasses may alter the visual appearance of the eyes, such as introducing shadows or distortions around the eye area, which can make it more difficult for the model to recognize patterns learned during the training process.

Overall, the results of this evaluation indicate that the system performs very well for users without glasses, but still exhibits limitations when applied to users wearing glasses. Therefore, further improvements are required, such as incorporating additional datasets that include variations of users with glasses or applying more robust preprocessing techniques, to enhance system performance under these conditions.

3.5 Analysis of the System's Advantages and Limitations

Based on the results of model training, performance evaluation, and system implementation on Android devices, the drowsiness detection system developed in this study exhibits several strengths and limitations that require further analysis. This analysis is essential to provide a comprehensive understanding of the system's capabilities as well as potential directions for future development.

1. System Strengths
   a. One of the main strengths of this system is its ability to perform real-time drowsiness detection on Android smartphones. This is demonstrated by the system's performance, which operates within a range of 6.0–7.3 FPS with a latency of 60–72 ms per frame, enabling the system to respond quickly to the user's condition. In addition, the notification response time of less than 1 second indicates that the system can provide immediate alerts when drowsiness is detected.
   b. In terms of model accuracy, the system shows excellent performance, achieving approximately 96% accuracy after the fine-tuning process. The balanced precision, recall, and F1-score across both classes indicate that the model is capable of consistently distinguishing between open-eye and closed-eye conditions. This is further supported by the confusion matrix results, which show a reduction in classification errors after fine-tuning.
   c. Furthermore, the system demonstrates good robustness against variations in lighting conditions. Based on the evaluation results, the system achieves 100% accuracy under bright conditions, 98% under normal conditions, and 95% under low-light conditions, indicating that it can perform effectively across different environmental settings.
   d. Another advantage is the use of the lightweight and efficient MobileNetV1 model, which makes it suitable for deployment on resource-constrained devices. This is evidenced by the relatively low memory usage of approximately 128 MB, allowing the system to operate without significantly affecting device performance.
2. System Limitations
   a. Despite its overall good performance, the developed system still has several limitations. One of the main limitations is the significant performance degradation when applied to users wearing glasses. Based on the evaluation results, the system accuracy decreases to 50% for users with glasses. This is primarily caused by light reflections on the surface of the lenses, which interfere with the detection of eye features, resulting in suboptimal recognition of the eye region.
   b. In addition, although the system is capable of operating in real time, the achieved FPS is relatively low compared to desktop- or GPU-based systems. This is due to the limited computational capability of mobile devices, as well as the need to perform multiple processes simultaneously, such as face detection, eye region cropping, and model inference.
   c. Another limitation is that the system relies on a single primary indicator, namely the eye state (open or closed). This approach does not yet incorporate other important indicators such as blink duration (PERCLOS), yawning detection, or head movement, which could potentially improve the accuracy and reliability of the system in detecting drowsiness more comprehensively.
   d. Furthermore, the dataset used for training is still limited in variation and may not fully represent all real-world conditions, such as extreme facial variations, the use of additional accessories, or very low-light environments.

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

4. CONCLUSION

This study successfully developed a real-time drowsiness detection system for smartphone users on the Android platform by utilizing a combination of Google ML Kit for face detection and MobileNetV1 for eye state classification. The developed model was trained using a combination of a public dataset and a self-collected dataset, and its performance was further improved through a fine-tuning process. The evaluation results indicate that the model achieves good performance, with an accuracy of approximately 96%, along with balanced precision, recall, and F1-score values across both classes, namely open-eye and closed-eye conditions. The confusion matrix analysis also demonstrates an improvement in model performance after fine-tuning, particularly in reducing classification errors that may affect system reliability. The implementation on Android devices shows that the system is capable of operating in real time with stable performance. The system achieves a processing speed of 6.0–7.3 FPS, with latency ranging from 60 ms to 72 ms per frame, and a notification response time of less than 1 second. In addition, the memory usage of approximately 128 MB indicates that the system remains within an acceptable range for deep learning-based mobile applications. Testing under real-world conditions shows that the system has good robustness to variations in lighting, achieving accuracies of 100% under bright conditions, 98% under normal conditions, and 95% under low-light conditions. However, the system still has limitations when applied to users wearing glasses, where the accuracy decreases to 50% due to light reflections on the lenses. Overall, the developed system is capable of accurately detecting drowsiness in real time on smartphone devices. This study demonstrates that the use of lightweight models such as MobileNetV1 is effective for implementation on mobile platforms. For future work, it is recommended to incorporate more diverse datasets, integrate additional indicators such as yawning detection or temporal analysis, and optimize system performance to improve robustness and adaptability across various user conditions.

REFERENCES

[1] C. Dhasarathan *et al.*, "Tensor RT optimized driver drowsiness detection system using edge device," *Ain Shams Eng. J.*, vol. 16, no. July, 2025, doi: https://doi.org/10.1016/j.asej.2025.103620.

[2] D. Khetan, A. Nawani, A. Aggarwal, and M. S. Kaur, "Driver Drowsiness Detection in Real-time," *Fusion Pract. Appl.*, vol. 7, no. 2, pp. 91–99, 2022, doi: Doi : https://doi.org/10.54216/FPA.070203.

[3] Y. Ling and X. Weng, "Efficient and Robust Driver Fatigue Detection Framework Based on the Visual Analysis of Eye States," *Promet – Traffic&Transportation*, vol. 35, no. 4, pp. 567–582, 2023.

[4] Y. Wang, B. Liu, and H. Wang, "Fatigue Detection Based on Facial Feature Correction and Fusion," in *Journal of Physics: Conference Series*, Purpose-Led Publishing, 2022. doi: 10.1088/1742-6596/2183/1/012022.

[5] Z. Zhao, N. Zhou, L. Zhang, H. Yan, Y. Xu, and Z. Zhang, "Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN," *Hindawi Comput. Intell. Neurosci.*, no. 3, 2020, doi: 10.1155/2020/7251280.

[6] N. Jamil, M. Haziq, M. Fadhil, and M. I. Ramli, "Comparing MobileNet-SSD and YOLO v3 Learning Architecture for Real- time Driver's Fatigue Detection," *Int. J. Acad. Res. Bus. Soc. Sci.*, vol. 11, no. 12, pp. 2409–2419, 2021, doi: 10.6007/IJARBSS/v11-i12/11984.

[7] M. Vimala, M. Nandhini, S. Jasmine, P. R. Kumar, and S. Ramasamy, "Franklin Open Implementation of a Lightweight Deep Learning Model for Detecting Driver Fatigue," *Franklin Open*, vol. 15, 2026, doi: 10.1016/j.fraope.2026.100547.

[8] M. A. B. Abbass and Y. Ban, "MobileNet-Based Architecture for Distracted Human Driver Detection of Autonomous Cars," *Electronics*, pp. 1–14, 2024.

[9] H. Z. Ilmadina, M. Naufal, and D. S. Wibowo, "Drowsiness Detection Based on Yawning Using Modified Pre-trained Model MobileNetV2 and ResNet50," *Matrik J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 22, no. 3, pp. 419–430, 2023, doi: 10.30812/matrik.v22i3.2785.

[10] Z. Xiao, Z. Hu, L. Geng, F. Zhang, J. Wu, and Y. Li, "Fatigue Driving Recognition Network : Fatigue Driving Recognition Via Convolutional Neural Network and Long Short-Term Memory Units," *IET Intell. Transp. Syst.*, pp. 1–7, 2019, doi: 10.1049/iet-its.2018.5392.

[11] S. Cao, P. Feng, W. Kang, Z. Chen, and B. Wang, "Optimized driver fatigue detection method using multimodal neural networks," *Sci. Rep.*, vol. 15, pp. 1–26, 2025.

JITEKH
Jurnal Ilmiah Teknologi Harapan
P-ISSN: 2338-5677 || E-ISSN 2549-6646
Vol. 14, No. 1, March 2026
DOI: https://doi.org/10.35447/jitekh.v14i1.1374

.

[12] O. Jagtap, S. Chaurasia, P. Choudhary, G. Buwade, M. Dhote, and U. B. Aher, "Driver Drowsiness Detection System using Arduino and Deep Learning," *Int. J. Innov. Eng. Sci.*, vol. 8, no. 10, pp. 6–19, 2023.

[13] E. Prasetyo, R. Purbaningtyas, R. D. Adityo, N. Suciati, and C. Fatichah, "Combining MobileNetV1 and Depthwise Separable convolution bottleneck with Expansion for Classifying the Freshness of Fish Eyes," in *INFORMATION PROCESSING IN AGRICULTURE 9*, 2022, pp. 485–496.

[14] C. Zhang, T. Yang, and J. Yang, "Image Recognition of Wind Turbine Blade Defects Using Attention-Based MobileNetv1-YOLOv4 and Transfer Learning," *Sensors*, vol. 22, pp. 1–18, 2022.

[15] F. Su, Y. Zhao, G. Wang, P. Liu, Y. Yan, and L. Zu, "Tomato Maturity Classification Based on SE-YOLOv3-MobileNetV1 Network under Nature Greenhouse Environment," *Argonomy*, vol. 12, pp. 1–15, 2022.

[16] M. M. Mijwil, R. Doshi, K. K. Hiran, O. J. Unogwu, and I. Bala, "Mobilenetv1-Based Deep Learning Model for Accurate Brain Tumor Classification," *Mesopotamian J. Comput. Sci.*, vol. 2023, pp. 29–38, 2023.

[17] A. Pundir *et al.*, "Enhancing Gait Recognition by Multimodal Fusion of MobileNetv1 and Xception Features Via PCA for OaA - SVM Classification," *Sci. Rep.*, vol. 14, pp. 1–17, 2024, doi: 10.1038/s41598-024-68053-y.

[18] 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 Contin*, vol. 82, no. 3, 2025, doi: 10.32604/cmc.2025.059972.

[19] A. A. Minhas, S. Jabbar, M. Farhan, and M. N. U. Islam, "A Smart Analysis of Driver Fatigue and Drowsiness Detection Using Convolutional Neural Networks," *Multimed. Tools Appl.*, vol. 81, pp. 26969–26986, 2022.

[20] G. Zhou, J. You, Q. Wu, J. Liu, and Y. Luo, "MobileNet-AFF : Multi-Scale Feature Fusion for Fatigue Driving Detection," in *International Conference on Computer, Vision and Intelligent Technology*, Associaton for Computing Machinery, 2023.