Mobile Application Development for Drowsy Driving Alert Using Convolutional Neural Networks

Yuda Gabriel[1], Adzikirani [2*], Dianthy Marya [3]

[1,2,3] Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

yudagabriel7@gmail,com, adzikirani@polinema.ac.id, dianthy@polinema.ac.id

Abstract — **Drowsiness is a leading cause of traffic accidents globally, necessitating real-time detection systems to enhance driver safety. This study addresses this need by designing and developing an Android-based mobile application for drowsiness detection utilizing a hybrid Convolutional Neural Network (CNN) approach. The system employs two specialized CNN models: one for classifying eye states (open/closed) and another for mouth states (yawn/no_yawn). To improve accuracy and reduce false positives, particularly during speech, the system integrates a geometric filter based on Mouth Aspect Ratio (MAR) calculations. The application, built with Android Studio and Kotlin, uses the device's front camera for real-time monitoring and provides multi-modal alerts (audio, vibration, flashlight, notification) upon detecting drowsiness. Model training achieved high accuracies of 99.82% for eye detection and 98.02% for mouth detection. Real-world testing under normal lighting conditions (300-350 lux) yielded an average detection accuracy of 98.33%, while performance in low-light scenarios (10-25 lux) averaged 88.33%. The application successfully differentiates between drowsiness indicators and normal activities like talking and blinking, demonstrating its potential as a reliable and practical tool for mitigating drowsy driving incidents.**

Keywords: Android Application, Convolutional Neural Network (CNN), Drowsiness Detection, Hybrid Model, Mouth Aspect Ratio (MAR), Real-time Monitoring.

I. INTRODUCTION

Road safety has evolved into a fundamental global challenge, given the significant impact it has on mortality rates and the socioeconomic burden across the globe. One of the primary factors consistently contributing to fatal traffic accidents is the condition of drivers experiencing impaired alertness due to drowsiness (drowsy driving)[1][2]. Physiologically, this condition triggers cognitive dysregulation, leading to slowed reaction times, impaired executive function in decision-making, and a drastic reduction in visual attention factors that collectively increase the likelihood of collisions on the road. Although various safety campaigns have been promoted, internal factors such as sleep deprivation, chronic fatigue, and certain pharmacological effects continue to exacerbate the complexity of this problem, necessitating technology-based intervention systems capable of operating preventively and with precision[3].

As innovation in the field of artificial intelligence accelerates, the integration of computer vision into mobile devices offers strategic opportunities to mitigate these risks through active monitoring[4]. In the field of pattern recognition, Convolutional Neural Networks (CNN)[5][6] algorithms have outperformed traditional methods thanks to their ability to automatically perform hierarchical feature extraction from complex visual data. Through their deep architecture[7], CNNs can identify anomalies in a driver's facial expressions and detect micro-biometric indicators of fatigue, such as changes in eyelid closure duration and the frequency of mouth activity during yawning signs that are difficult to accurately identify through conventional observation[8][9].

However, a review of the current literature reveals significant discrepancies between the performance of theoretical models and their practical implementation in real-time environments[10]. Many existing detection systems still rely critically on internet connectivity for cloud-based processing, which inherently introduces latency issues. Furthermore, classical rule-based methods such as Haar Cascades[11], Support Vector Machines (SVM), or purely geometric calculations like Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) often demonstrate vulnerability to extreme lighting fluctuations, variations in facial angles (pose estimation), and physical obstructions such as the wearing of glasses[12]. On the other hand, highly complex deep learning models often require high computational overhead (hardware overhead), thereby reducing their efficiency when operated on mobile devices with limited resources[13].

To address these challenges, this study proposes the development of an innovative Android application that uses a hybrid approach to detect drowsiness offline[14]. The novelty of this system lies in combining the robust capabilities of a CNN model in recognizing visual patterns with a rule-based system for analyzing mouth aspect ratios[15]. The goal is to improve detection accuracy while reducing false alarms. By processing all computations locally on the device, this app not only ensures stable operation without relying on a network but

---

*Corresponding Author

also provides a fast, responsive, and easily accessible safety tool for the general public, thereby effectively reducing accidents caused by driver fatigue[16].

II. METHOD

The research done is applied research and development (R&D), aimed at making a new software product to solve a real-world problem. The method used several steps: designing the system, getting and preparing the data, building and training a CNN model, making an Android app, putting everything together, and doing thorough testing.

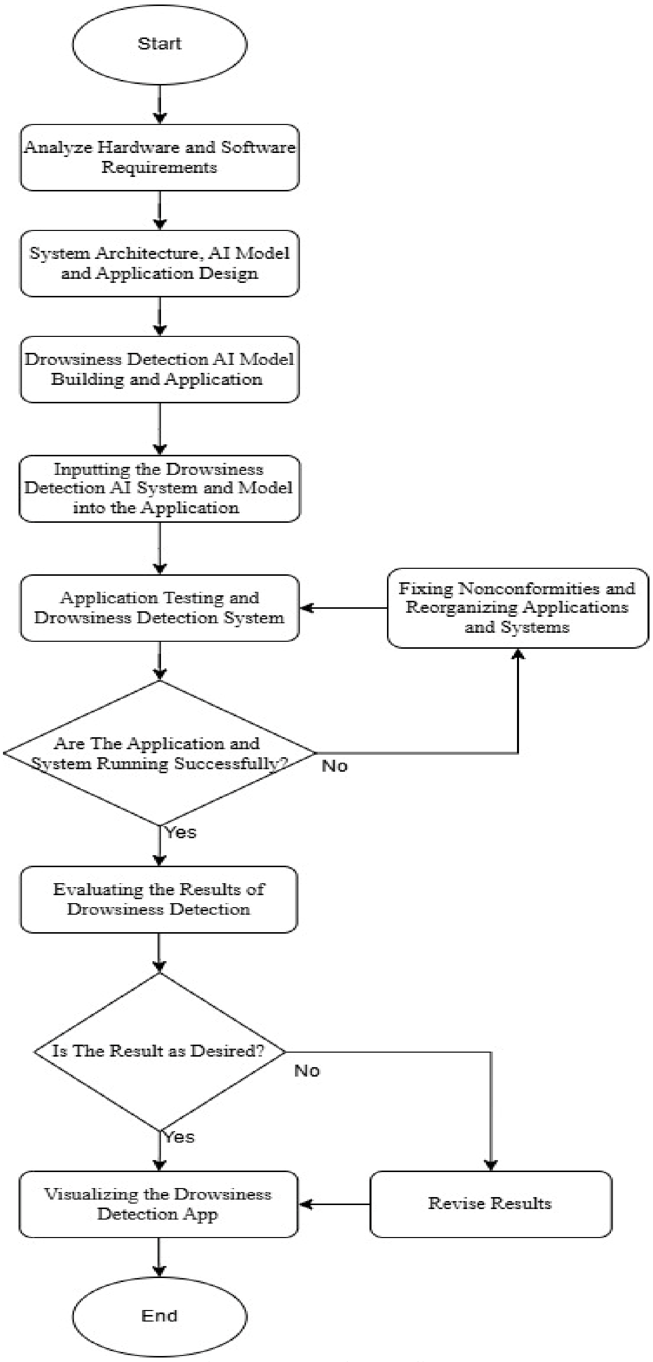


Figure 1. Research stage diagram

Based on Fig. 1, it starts with the front camera of a smartphone capturing live video. Each video frame goes through preprocessing, which involves finding faces and identifying key points on the face using Google's ML Kit. These key points help to find the areas of interest (ROIs) for the eyes and mouth. These ROIs are then sent to two different, specialized TensorFlow Lite models:

1. Eye State Model: A CNN model that looks at the eye area and decides if the eye is open or closed.
2. Mouth State Model: A CNN model that looks at the mouth area and decides if someone is yawning or not yawning.

At the same time, the coordinates of the mouth key points are used to calculate the Mouth Aspect Ratio (MAR). The final decision about drowsiness is made by a hybrid logic module that uses the results from the two CNN models and the MAR value. If drowsiness is detected for a certain amount of time at 1.7 seconds, the app starts multiple alarms to alert the user.

A. *Dataset and Model Hybrid Creation*

The data for this study was obtained from the Kaggle and Roboflow platforms, which contain images of eyes labelled as “Open” and “Closed,” as well as images of mouths labelled as ‘Yawn’ and “No_yawn.” Eye images were converted to RGB format with a size of 128 x 128 pixels, while mouth images were converted to grayscale with a size of 64 x 64 pixels. The data preparation process began by detecting objects using the Haar Cascade method for the eye region and MTCNN for the mouth region in full-face images. The process continues by normalizing pixel values to a range of 0 to 1, followed by data augmentation using techniques such as mirroring, random flipping, rotation, and zooming primarily on the eye image dataset to increase data diversity and prevent overfitting. Finally, the dataset is split into training and validation sets in an 80:20 ratio using a stratified split to maintain balanced class proportions.

B. *Eye Model Architecture*

The CNN model for eyes is specifically designed to classify Closed and Open conditions. This model utilizes eye images in 128 × 128 pixel RGB format as input, which then goes through a series of convolution, normalization, and pooling processes to extract visual features. The purpose of this architecture is for the model to accurately recognize the difference between open and closed eyes despite variations in facial position and lighting. This model is quite deep with four convolution layers that increasingly increase the number of filters, so that the resulting features are more representative.

TABLE I
ARCHITECTURAL CONFIGURATION OF A CNN MODEL FOR EYE CLASSIFICATION

Layers	Specifications/Parameters
Input	128 x 128 RGB Image
Augmentation	Flip, Rotation, Zoom
Conv2D (1)	32 Filters, Kernel 3 x 3
Batch Normalization (1)	-
MaxPooling2D (1)	Pool Size 2 x 2

Conv2D (2)	64 Filters, Kernel 3 x 3
Batch Normalization (2)	-
MaxPooling2D (2)	Pool Size 2 x 2
Conv2D Layer (3)	128 Filters, Kernel 3 x 3
Batch Normalization (3)	-
MaxPooling2D (3)	Pool Size 2 x 2
Conv2D Layer (4)	128 Filters, Kernel 3 x 3
Batch Normalization (4)	-
MaxPooling2D (4)	Pool Size 2 x 2
Flatten	-
Dense (1)	256 Neurons
Dropout (1)	Rate: 0.5
Dense (2)	128 Neurons
Dropout (2)	Rate: 0.3
Output	2 Neurons (Closen/Open)

Table I shows the total number of layers used (including Conv, Pooling, Dense, Dropout, Flatten, Output): ± 20 layers. With total neurons in fully connected layers: 386 (256 + 128 + 2) and Conv2D Kernel: all 3 × 3 in size and the last Pooling: 4 layers of MaxPooling with a kernel size of 2 × 2.

C. Mouth Model Architecture

The CNN model for the mouth is designed to classify Yawn and No_yawn conditions. Unlike the eye model, this model uses 64 × 64 pixel grayscale images as input, making it lighter in computation. Its main purpose is to recognize mouth shape patterns when the subject yawns or does not yawn, by utilizing the convolution process to extract texture and lip contour features. Although the input is simpler, the model architecture still consists of three convolutional layers with gradually increasing filters to obtain a stronger feature representation.

TABLE II
ARCHITECTURAL CONFIGURATION OF A CNN MODEL FOR MOUTH CLASSIFICATION

Layers	**Specifications/Parameters**
Input	64 x 64 Grayscale Image
Conv2D (1)	32 Filters, Kernel 3 x 3
Batch Normalization (1)	-
MaxPooling2D (1)	Pool Size 2 x 2
Conv2D (2)	64 Filters, Kernel 3 x 3
Batch Normalization (2)	-
MaxPooling2D (2)	Pool Size 2 x 2
Conv2D Layer (3)	128 Filters, Kernel 3 x 3
Batch Normalization (3)	-
MaxPooling2D (3)	Pool Size 2 x 2
Flatten	-
Dense (1)	256 Neurons
Dropout (1)	Rate: 0.5
Dense (2)	128 Neurons
Dropout (2)	Rate: 0.3
Output	2 Neurons (Yawn/No_yawn)

Table II shows the total number of layers used (including Conv, Pooling, Dense, Dropout, Flatten, Output): ± 16 layers. With total neurons in the fully connected layer: 386 (256 + 128 + 2) and Conv2D Kernel: all are 3 × 3 in size and the last Pooling: 3 layers of MaxPooling with a kernel size of 2 × 2.

D. Determination of Mouth Aspect Ratio (MAR) in Application

To make yawning detection more accurate, this study adds the Mouth Aspect Ratio (MAR) as a second check after the CNN prediction. MAR is found by dividing the height of the mouth by its width. A higher MAR usually means someone is yawning. To set the first threshold, MAR values were recorded from 10 people in two situations: when they were yawning and when they weren't. The first results are shown in Table III.

TABLE III
MOUTH ASPECT RATIO (MAR) DATA DETERMINATION OF THRESHOLD

Respondents	**Prediction Mouth**	**Mouth Aspect Ratio (MAR)**
R1	Yawn	0.43
R2	No Yawn	0.27
R3	Yawn	0.67
R4	No Yawn	0.34
R5	Yawn	0.58
R6	No Yawn	0.31
R7	**Yawn**	**0.70**
R8	No Yawn	0.20
R9	Yawn	0.63
R10	No Yawn	0.39

The data in Table 3 show that the Mouth Aspect Ratio (MAR) values are distributed differently between the two tested conditions. The MAR values during yawning (Yawn) range from 0.43 to 0.70, while during non-yawning (No Yawn) they range from 0.20 to 0.39. The visual distinction within these value ranges suggests that the MAR parameter is likely a good indicator for distinguishing yawning from other oral activities, such as speaking. However, these values are preliminary references that still require further verification to determine the most appropriate threshold. Therefore, additional testing was conducted using a specialized dataset focused on yawning activity to determine a more accurate threshold, as listed in Table IV.

TABLE IV
MOUTH ASPECT RATIO (MAR) DATA YAWN THRESHOLD DETEERMINATION

Respondents	**Prediction Mouth**	**Mouth Aspect Ratio (MAR)**
R1	Yawn	1.30

R2	Yawn	0.89
R3	Yawn	1.11
R4	**Yawn**	0.80
R5	Yawn	0.86
R6	Yawn	0.95
R7	Yawn	1.17
R8	Yawn	1.28
R9	Yawn	0.84
R10	Yawn	1.05

Analysis of Table IV shows that the mouth area ratio (MAR) values for yawning conditions fall within the range of 0.80 to 1.30. Based on the existing empirical distribution, a threshold value of 0.75 was set as the parameter used for classification. The system works by classifying an activity as 'yawning' if the MAR value exceeds the specified threshold, and at the same time is supported by a positive prediction result from the CNN model. Combining these two parameters has proven to be very helpful in improving the system's accuracy, particularly in reducing false alarms caused by similar activities, such as speaking, so that drowsiness alerts only appear when the condition actually occurs.

E. Determining Time Threshold in Application

In this application, the process involved several experiments to identify the optimal point where detection sensitivity is sufficiently high while maintaining accuracy. This demonstrates that even as the communication medium shifts to digital, the essence of formal structure remains a key pillar in maintaining normal conditions with clear indications of drowsiness (eyes closed for an extended period). Based on test results across various durations, it was determined that a threshold value of 1.7 seconds is the most appropriate to use.

During this period, the system demonstrated the best performance in triggering an alarm without causing many false detections due to natural blinking. This indicates that even though the communication medium has shifted to digital, the core of the formal structure remains the main pillar in preventing the system from becoming overly sensitive and frequently triggering false alarms. Conversely, durations exceeding 1.7 seconds are considered too long for providing early warnings to drivers. Therefore, the 1.7 second threshold was selected as the primary application parameter because it effectively balances safety and user comfort while maintaining a sufficiently high level of accuracy.

F. System Analysis

System analysis is essential for determining the requirements and configuration needed for the development of a research product based on the problem and methods used. The following system analysis is likely to be required.

a. PC Specifications
- CPU Intel Core I7 Generation 12 or AMD Ryzen 7 5000 series
- RAM 16 GB
- Storage 128 GB
- OS (Windows/Linux/MacOS)
- VGA Nvidia RTX Series (Optional)

b. Smart Phone Android
- Dimensity 700, Helio G60 or Snapdragon 600 series
- RAM 6 GB
- Storage 64 GB

c. Applications and Modules
- Android Studio
-Google Colaboratory
- Google Drive
- Kaggle
- Roboflow
- OpenCV
- Tensorflow and Tensorflow.lite
- Numpy

III. RESULTS AND DISCUSSION

A. Eye Detection Model Training Result

The eye detection model was built using a convolutional neural network (CNN) architecture specifically designed to distinguish between two states: open eyes and closed eyes. The system processes RGB images with a resolution of 128×128 pixels. Training was conducted on the Google Colaboratory platform using optimization mechanisms such as EarlyStopping to prevent overfitting when accuracy on the validation data no longer improved, as well as ReduceLROnPlateau, which automatically adjusts the learning rate to ensure a more stable convergence process. Training ran for 23 epochs before automatically stopping, with the model's optimal weights achieved at the 18th epoch. The evaluation results showed excellent performance, with a test accuracy of 99.82% and a loss of 0.0066. The total computation time required was 26,887.50 seconds, approximately 7.4 hours, reflecting the complexity of the model structure and the volume of data used in this research.

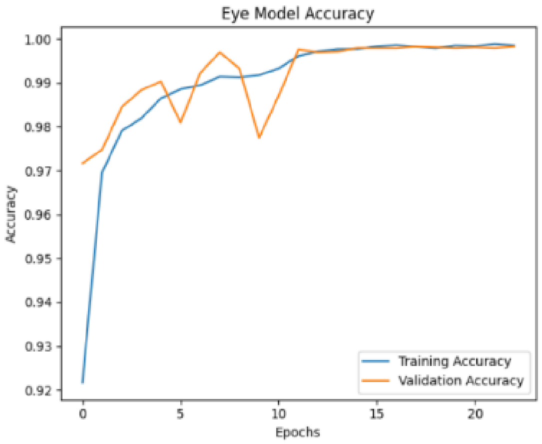


Figure 2. Accuracy eye model graph

Fig. 2 illustrates the model training process by displaying accuracy and loss curves. From the accuracy curve (on the left), it can be seen that the training curve (blue) and the validation curve (orange) show a significant increase early in the process and reach a saturation level above 99% after 12 epochs. This trend indicates that the model can effectively extract features that distinguish between open and closed eyes in a short period of time. Additionally, the small difference between the two curves suggests that the model possesses good adaptability without evident overfitting during training.

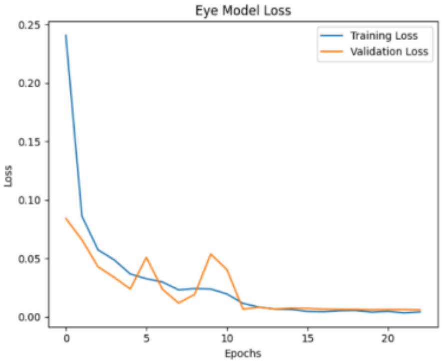


Figure 3. Loss eye model graph

Fig. 3 illustrates these results are supported by the loss graph, which shows a noticeable decrease in error rates from the start of the training process. After the 12th epoch, the loss values on the training and validation data show very low volatility, indicating that the model has achieved optimal convergence. To examine the classification model's performance more comprehensively, an analysis was conducted using a confusion matrix.

TABLE V
CONFUSION MATRIX FOR EYE DETECTION MODEL

Actual/Predicted	**Close**	**Open**
Close	2721	5
Open	6	3439

From Table V show this confusion matrix shows the number of correct and incorrect predictions for each class (open eyes and closed eyes) based on the test data. It is clear that the model correctly classified most of the data. There are 2721 True Positive (TP) results for the Closed class and 3439 True Negative (TN) results for the Open class. Only 6 False Positive (FP) and 5 False Negative (FN) errors were found, showing that the classification errors are very small. Based on these results, the overall accuracy is 99.82%. The precision and recall for the Closed class are 99.78% and 99.82% respectively. These results prove that the model is very reliable in detecting driver eye conditions.

Based on these values, we can calculate performance evaluation metrics as follows:

a. Accuracy: Measures how often the model makes correct predictions overall.

$$\text{Accuracy} = \frac{(TP + TN)}{TOTAL(\text{TP} + \text{TN} + \text{FN} + \text{FP})} \times 100\% \quad (1)$$

$$\text{Accuracy} = \frac{(2721 + 3439)}{(2721 + 5 + 6 + 3439)} \times 100\%$$

Accuracy = 99.82%

b. Precision (for the 'Closed' class): Measures how accurate the model's predictions of 'closed eyes' are.

$$\text{Precission} = \frac{TP}{(\text{TP} + \text{FP})} \times 100\% \quad (2)$$

$$\text{Precission} = \frac{2721}{(2721 + 6)} \times 100\%$$

Precission = 99.78%

c. Recall (Sensitivity for the 'Closed' class): Measures the model's ability to find all actual cases of closed eyes.

$$\text{Recall} = \frac{TP}{(\text{TP} + \text{FN})} \times 100\% \quad (3)$$

$$\text{Recall} = \frac{2721}{(2721 + 5)} \times 100\%$$

Recall = 99.82%

d. F1 Score (for the 'Closed' class): The harmonic mean of Precision and Recall, providing a single number that balances both

$$\text{F1 Score} = 2 \times \frac{(Precission \times Recall)}{(Precission + Recall)} \times 100\% \quad (4)$$

$$\text{F1 Score} = 2 \times \frac{(0.9978 \times 0.9982)}{(0.9978 + 0.9982)} \times 100\%$$

F1 Score = 99.80%

These calculation results show the model has excellent performance. The very high Recall value (99.82%) is very important because it means the model only missed 5 out of 2726 actual cases of closed eyes. The very low number of False Negatives proves that the model is very reliable in detecting drowsiness and has a very low risk of failing to give an alert when needed

B. Mouth Detection Model Training Result

The second model is specifically designed to detect oral states by classifying two conditions: yawning and not yawning. Unlike the eye-detection model, the CNN architecture used in this model is simpler (low-complexity architecture) and utilizes 64×64-pixel grayscale images. The choice of grayscale format aims to reduce computational load and focus feature extraction on morphological aspects, such as the shape and contour lines of the mouth, which serve as the primary indicators for recognizing yawning activity. Training was conducted using the Google Colaboratory platform, which includes the EarlyStopping callback to prevent overfitting and the ReduceLROnPlateau callback to optimize the learning rate so that the model can converge more stably.

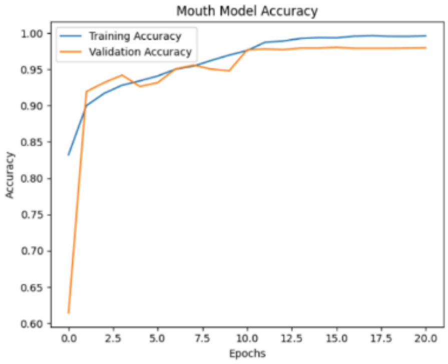


Figure 4. Accuracy mouth model graph

Fig. 4 illustrates the training process for the mouth detection model lasted for 21 training cycles (epochs) and was then automatically terminated by the EarlyStopping feature. The best weights were successfully obtained in the epoch 16. Based on the final evaluation, the model demonstrated excellent performance with a validation accuracy of 98.02% and a loss value of 0.0873. The total computation time required during the training process was 2,605.58 seconds, equivalent to approximately 43 minutes. The model's learning dynamics indicate a consistent convergence pattern. The training accuracy curve (blue) and validation accuracy curve (orange) rise significantly from the start and reach a saturation level above 95% after 10 epochs.

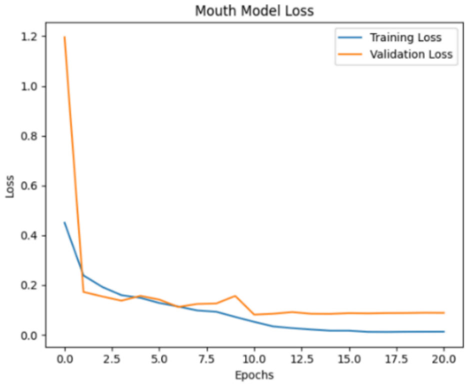


Figure 5. Loss mouth model graph

Although in Fig. 5 there is a slight difference between the two curves at the end of the training phase, this does not indicate signs of overfitting, as the EarlyStopping method has successfully maintained the model's performance stability. This is supported by the loss graph, which shows a sharp drop in values at the beginning of the epochs, indicating that the model is able to capture important features efficiently. The lowest validation loss point was reached between epochs 10 and 16, corresponding to the highest validation accuracy.

To evaluate the model's classification performance more comprehensively, an analysis was conducted using a confusion matrix, as shown in Table VI. The matrix results indicate highly accurate predictions for both classes: 1536 true positives (TP) for the Yawn category and 1535 true negatives (TN) for the No_yawn category. A very small number of classification errors were recorded, namely only 31 False Positive cases and 31 False Negative cases. Based on this data, the model as a whole has an accuracy of 98.02%, and the precision and recall values remain above 97%. This very high precision indicates that the model can accurately distinguish mouth conditions, which is a crucial aspect in helping the drowsiness detection system function effectively through yawning.

TABLE VI
CONFUSION MATRIX FOR MOUTH DETECTION MODEL

Actual/Predicted	Yawn	No Yawn
Yawn	1536	31
No Yawn	31	1535

As shown in Table 6 performance evaluation metrics for the mouth model are calculated as follows:

a. Accuracy:
To obtain this value, use Equation (1)

$$\text{Accuracy} = \frac{(1536 + 1535)}{(1536 + 31 + 31 + 1535)} \times 100\%$$

Accuracy = 98.02%

b. Precision (for class 'Yawn'):
To obtain this value, use Equation (2)

$$\text{Precision} = \frac{1536}{(1536 + 31)} \times 100\%$$

Precision = 98.02%

c. Recall (Sensitivity for class 'Yawn'):
To obtain this value, use Equation (3)

$$\text{Recall} = \frac{1536}{(1536 + 31)} \times 100\%$$

Recall = 98.02%

d. F1 Score (for class 'Yawn'):
To obtain this value, use Equation (4)

$$\text{F1 Score} = 2 \times \frac{(0.9802 \times 0.9802)}{(0.9802 + 0.9802)} \times 100\%$$

F1 Score = 98.02%

The calculation results show that the mouth model has a very balanced performance, with identical Precision and Recall values. The Recall of 98.02% indicates that the model is very reliable in finding most actual cases of yawning. Although there are 31 False Negative cases, this number is relatively small compared to the number of correct detections, so the model is considered reliable enough for use in an application.

C. Feature Alarm System Testing

After refining the core detection function, the next step is to test the designed multimodal alarm system to ensure that the driver's response is effective. Based on the test results, the system operates in accordance with the specified requirements.

TABLE VII
IMPLEMENTED APPLICATION FEATURES

Features	Status
Sound Alarm	Active
Vibration	Active
Flashlight	Active
Popup Notification	Active
Screen Always On	Active

As can be seen in Table VII when signs of drowsiness are detected and exceed the specified threshold (drowsinessThreshold), the application will activate various alert methods simultaneously as follows:

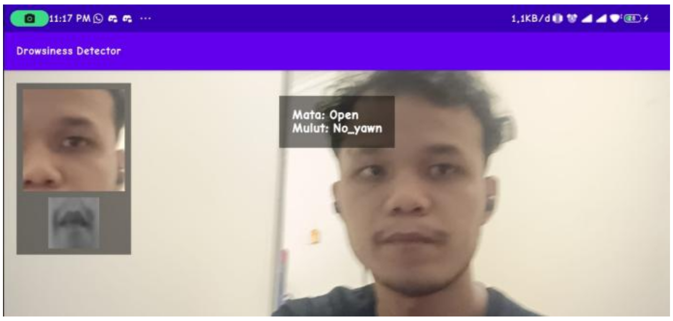
1. Audible warning (siren): The system begins continuous playback of a specific sound from the application's internal source, with the volume automatically increased to maximum to ensure the warning remains effective in very noisy environments.

2. Tactile Alert (Vibration): The device vibrates with a continuous and strong pattern to provide a physical sensation to the driver.
3. Visual Stimulus (Flashlight): The flashlight is used as an aid set to flash regularly as an additional visual stimulus, which is particularly important when driving in low-light conditions or at night.
4. Interface Notification (Pop-up Notification): A high-priority notification display that shows a clear warning message on the device screen.

To ensure the monitoring process runs continuously without interruption, two system stability features are enabled. First, the foreground service feature is activated when the detection process begins. This service ensures the app always runs in the background and prevents the Android operating system from automatically terminating the app, thereby preserving battery life. The app's active status is indicated by a persistent notification in the status bar, making it easy for users to access it. Second, the FLAG_KEEP_SCREEN_ON flag is used in MainActivity to prevent the device from entering sleep mode while the app is on the home screen. Combining these two features significantly enhances the system's reliability for continuous monitoring, whether the app is actively open or running in the background.

D. Testing Application at Normal Light Intensity

This test was done to check how well the app works under normal lighting conditions that users typically experience in their daily lives. In this study, normal lighting is defined as an environment with an average brightness of 300 to 350 lux, which matches the standard lighting level for normal visual activities indoors.



Mata: Open
Mulut: No_yawn

Figure 6. Conditions with light intensity of 300-500 lux

Fig. 6 shows these conditions can come from natural light (sunlight) or artificial light (indoor lights) that is evenly spread, so the user's face can be clearly seen by the device's camera. The test results are shown in Table VIII.

TABLE VIII
TEST RESULTS UNDER NORMAL LIGHTING CONDITIONS (300-350 LUX)

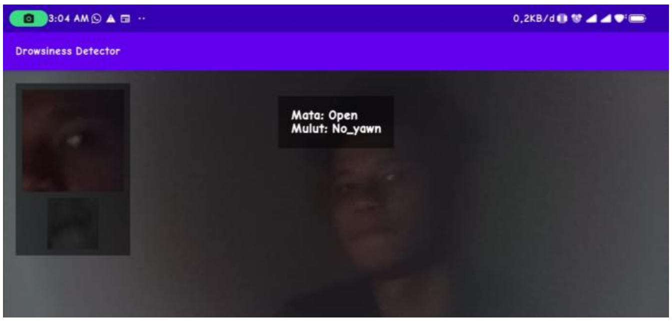
Scenario	Eyes / Time	Mouth / MAR	Detection Result	Accuracy
Normal	Open / >1.7s	Closed / <0.75	No Drowsiness	100%
Drowsiness	Closed / >1.7s	Open / >0.75	Drowsiness	100%
Yawn	Open / <1.7s	Open / >0.75	Drowsiness	100%
Microsleep	Closed / >1.7s	Closed / <0.75	Drowsiness	100%
Talking	Open / <1.7s	Open / <0.75	No Drowsiness	90%
Blink	Closed / <1.7s	Closed / <0.75	No Drowsiness	100%

According to Table 8, the system shows very high accuracy in detecting different driver condition scenarios. Test results demonstrate the system's excellent reliability across various operational scenarios. In normal scenarios, the system is capable of detecting Drowsiness, Yawning, and Microsleep with an accuracy rate of up to 100%. This indicates that the use of a 7-second time threshold and a Mouth Aspect Ratio (MAR) value of 0.75 is highly effective in accurately distinguishing between alert drivers and those exhibiting signs of drowsiness or acute fatigue. Additionally, in the Blinking scenario, the system also achieved 100% accuracy, indicating that the duration used as a filter successfully prevented false alarms caused by normal eye movements. A decrease in accuracy occurred only in the Talking scenario, at 90%, where there was one misidentification out of ten trials. This error occurred due to changes in facial position and the shifting reference point of the mouth as the respondent spoke; however, overall, the system still demonstrated strong capability in detecting driving safety threats.

The average success rate across all scenarios is 98.33%, showing the system works very well under normal lighting conditions. Test results under normal lighting show the system performs very well and consistently, almost perfectly. Errors only happened in the Talking scenario, which is usually affected by changes in facial expressions and mouth angles that look like someone is yawning. There were no errors in the Blink and Normal scenarios, showing that the Time Threshold filter and MAR measurements are accurate. Therefore, this system is reliable for use in normal lighting conditions without needing any extra adjustments.

E. Testing Application at Low Light Intensity

This test was done to check how well the app works in low light conditions, which is similar to driving at night. The test took place in a room with light levels between 10-25 lux, using only the phone's screen for lighting.



Mata: Open
Mulut: No_yawn

Figure 7. Conditions with light intensity of 10-25 lux

Fig.7 shows this situation is important because it's one of the key scenarios in real-world implementation of a drowsiness detection system. The results of the test are shown in Table 9.

TABLE IX.
TEST RESULTS UNDER NORMAL LIGHTING CONDITIONS (10-25 LUX)

Scenario	Eyes / Time	Mouth / MAR	Detection Result	Accuracy
Normal	Open / >1.7s	Closed / <0.75	No Drowsiness	90%
Drowsiness	Closed / >1.7s	Open / >0.75	Drowsiness	80%
Yawn	Open / <1.7s	Open / >0.75	Drowsiness	80%
Microsleep	Closed / >1.7s	Closed / <0.75	Drowsiness	90%
Talking	Open / <1.7s	Open / <0.75	No Drowsiness	90%
Blink	Closed / <1.7s	Closed / <0.75	No Drowsiness	100%

Based on Table IX, the test results show that the application can detect certain conditions with varying levels of accuracy in each scenario. System performance evaluation was conducted using six driving simulation scenarios to comprehensively test the algorithm's reliability. The system achieved perfect results in the Blink scenario with 100% accuracy, demonstrating the effectiveness of the time threshold filter in distinguishing normal blinks from signs of drowsiness. In the Normal, Microsleep, and Talking scenarios, the system demonstrated good consistency with accuracy rates of 90% each. These results indicate that the CNN model and classification method used are capable of maintaining stability in detection even when facial activity changes, such as speaking.

However, accuracy drops to 80% in the Drowsiness and Yawn scenarios. Based on the analysed test results, the system's failure to detect in both scenarios was due to low light intensity, which directly affected the system's ability to locate facial features such as the eyes and mouth. Nevertheless, overall the system still demonstrated fairly good performance with an average accuracy approaching perfection, where environmental lighting conditions remain a critical factor to consider in the system's next development phase. The overall average detection success rate is around 88.33%, indicating that the application's performance is still reliable, although low lighting conditions present a significant challenge.

From the test results, it can be concluded that low lighting causes a small drop in accuracy, especially in scenarios involving mouth detection (Drowsiness and Yawn). This happens because the system has difficulty identifying facial landmarks in dark areas, leading to some detection failures. However, the Blink scenario shows excellent performance, indicating that the time filter works very well. To improve performance, optimization is needed, such as using a face detection algorithm that works better in low light or adding infrared illumination to keep detection stable in dark conditions.

E. Application Testing on External Respondents

System performance was evaluated by testing 10 external participants with varying biometric characteristics. The diversity parameters examined included abnormal facial structures, lip shapes, and skin tone variations to ensure that detection results remained accurate and consistent. This test was conducted to determine how well the CNN model and MAR calculations can be applied to various types of human facial expressions and shapes, thereby ensuring the system's effectiveness as a universally applicable monitoring tool.

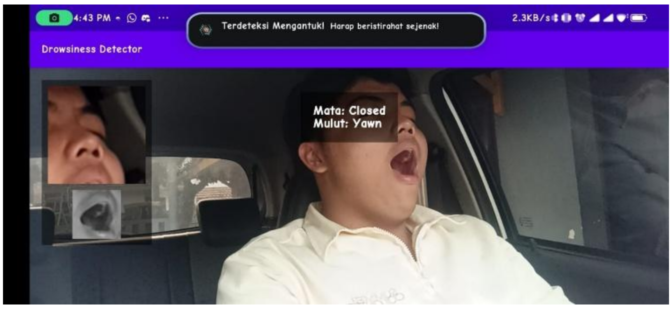


Figure 8. Experiments on Respondents While Driving

Fig.8 shows the system was tested in three critical scenarios yawning, speaking, and blinking to ensure the algorithm is resilient to potential false positives. These scenarios were selected to assess how accurately the system can distinguish genuine signs of drowsiness from ordinary facial movements. The test results, which include observations of eye and mouth conditions as well as the system's response to alarms, are detailed in Table X through Table XIII.

TABLE X.
RESULT EXTERNAL RESPONDENT TESTING YAWN SCENARIO

Respondent	Condition		Detection Result	Active Alarm
	Eyes	Mouth		
R1	Close	Open	Drowsiness	On
R2	Open	Open	Drowsiness	On
R3	Close	Open	Drowsiness	On
R4	Close	Open	Drowsiness	On
R5	Open	Open	Drowsiness	On
R6	Close	Open	Drowsiness	On
R7	Open	Open	Drowsiness	On
R8	Open	Open	Drowsiness	On
R9	Close	Open	Drowsiness	On
R10	Open	Open	Drowsiness	On

Table XI shows the results of the test under the Yawn condition, where participants were asked to yawn naturally. According to the table, all participants (R1–R10) showed varying eye conditions (either open or briefly closed) and had their mouths open with a MAR greater than 0.75, which is the main indicator of yawning. The system detected this condition as Drowsiness in all trials, and the alarm activated 100% of the time.

This success shows that the system has 100% accuracy in detecting the signs of yawning as potential drowsiness. The MAR parameter works effectively in distinguishing yawning from speaking, even though facial expressions varied quite a bit. However, this test did not include additional data such as the average duration of the mouth being open, which could be used to strengthen the analysis of yawning detection.

TABLE XI.
RESULT EXTERNAL RESPONDENT TESTING TALKING SCENARIO

Respondent	Condition		Detection Result	Active Alarm
	Eyes	**Mouth**		
R1	Open	Close	No Drowsiness	Off
R2	Open	Open	No Drowsiness	Off
R3	Open	Open	No Drowsiness	Off
R4	Open	Open	No Drowsiness	Off
R5	Open	Close	No Drowsiness	Off
R6	Open	Close	No Drowsiness	Off
R7	Open	Open	No Drowsiness	Off
R8	Open	Open	No Drowsiness	Off
R9	Open	Close	No Drowsiness	Off
R10	Open	Close	No Drowsiness	Off

Table XI shows the results of the test when respondents were speaking with their eyes open and moving their mouths normally. In this scenario, the MAR stayed below the threshold (0.75), so the system detected all respondents as No Drowsiness, and the alarm did not activate. The success rate reached 100%, proving that the MAR filter works well in avoiding false positives during speaking activities.

TABLE XII.
RESULT EXTERNAL RESPONDENT TESTING BLINK SCENARIO

Respondent	Condition		Detection Result	Active Alarm
	Eyes	**Mouth**		
R1	Open	Close	No Drowsiness	Off
R2	Open	Close	No Drowsiness	Off
R3	Open	Close	No Drowsiness	Off
R4	Open	Close	No Drowsiness	Off
R5	Open	Close	No Drowsiness	Off
R6	Open	Close	No Drowsiness	Off
R7	Open	Close	No Drowsiness	Off
R8	Open	Close	No Drowsiness	Off
R9	Open	Close	No Drowsiness	Off
R10	Open	Close	No Drowsiness	Off

Table XII shows the results of the test when respondents blinked normally with their mouths closed. In all the trials, the system identified this condition as No Drowsiness and the alarm remained inactive. The success rate was 100%, indicating that the Time Threshold filter (1.7 seconds) works well in distinguishing normal blinking from microsleep.

Overall, the test results from 10 external respondents in three main scenarios (Yawn, Talking, Blink) showed 100% accuracy for all cases. This confirms that the combination of the Time Threshold (1.7 seconds) and Mouth Aspect Ratio (MAR) parameters successfully provided consistent detection even when there were variations in faces among the respondents.

F. Discussion

The results show that the hybrid CNN-MAR approach is effective. The high accuracy of the individual models worked well in the real-time app. Using the MAR threshold was important, acting as a check to cut down on false alarms from the mouth CNN during speech, which is a common problem with systems that rely only on vision.

The difficulties faced during mobile development, especially matching the image processing steps between Python (OpenCV) and Android, show how careful software engineering is needed for AI projects. Choosing to keep the app in landscape mode helped fix major issues with mismatched coordinates between the camera and screen, making the facial feature cropping more stable and accurate.

The drop in performance under low light shows one of the limits of vision-based systems and suggests a future improvement, like adding infrared lighting or using better face detection models that work in low light.

IV. CONCLUSION

Based on the results of the research conducted, it can be concluded that the drowsiness detection system using a Convolutional Neural Network (CNN) successfully classified eye and mouth conditions with a very high level of accuracy, reaching 99% for the eye detection model and 99.41% for the mouth detection model. This indicates that even though the communication medium has shifted to digital, the essence of the formal structures of the Haar Cascade and MTCNN remains the primary foundation for real-time facial feature localization. On the other hand, the use of a time threshold of 1.7 seconds is considered the optimal parameter for triggering a warning alarm, thereby minimizing the risk of accidents caused by driver fatigue. This indicates that even though the communication medium has shifted to digital, the formal structure of the CNN model remains the primary foundation for ensuring optimal performance during the preprocessing and data augmentation stages. Consequently, this system exhibits good levels of precision and reliability, making this application suitable for use as a preventive solution to enhance driving safety.

REFERENCES

[1] R. Rahmadiyani and A. widyanti, “Prevalence of drowsy driving and modeling its intention: An Indonesian case study,” *Transp. Res. Interdiscip. Perspect.*, vol. 19, no. March, p. 100824, 2023, doi: 10.1016/j.trip.2023.100824.

[2] H. Triamiyono, “Dalam Proses Belajar Mahasiswa,” *Upaya Mengatasi Rasa Kantuk Di Kelas Dalam Proses Belajar Mahasiswa Taruna Akademi Maritim Djadajat*, vol. 2, pp. 64–69, 2014.

[3] E. Lety Istikhomah Puspita Sari, I. Ketut Agung Enriko, J. Gegerkalong Hilir No, J. DI Panjaitan No, and J. Tengah, “Sistem Peringatan Tersemat untuk Pengemudi Mengantuk Embedded Alert System for Drowsy Drivers,” *Jurnal Riset Rekayasa Elektro*, vol. 5, no. 1, 2023, [Online]. Available: http://jurnalnasional.ump.ac.id/index.php/JRRE

[4] Y. Jebraeily, Y. Sharafi, and M. Teshnehlab, “Driver Drowsiness Detection Based on Convolutional Neural Network Architecture Optimization Using Genetic Algorithm,” *IEEE Access*, vol. 12, no. April, pp. 45709–45726, 2024, doi: 10.1109/ACCESS.2024.3381999.

[5] R. O. Alnashwan, S. A. Chelloug, N. S. Almalki, I. Issaoui, A. Motwakel, and A. Sayed, “Lighting Search Algorithm With Convolutional Neural Network-Based Image Captioning System for Natural Language Processing,” *IEEE Access*, vol. 11, no. November, pp. 142643–142651, 2023, doi: 10.1109/ACCESS.2023.3342703.

[6] X. Li, S. Lai, and X. Qian, “DBCFace: Towards Pure Convolutional Neural Network Face Detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 1792–1804, 2022, doi: 10.1109/TCSVT.2021.3082635.

[7] S. Hossain, S. Umer, R. K. Rout, and H. Al Marzouqi, “A Deep Quantum Convolutional Neural Network Based Facial Expression Recognition For Mental Health Analysis,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 32, pp. 1556–1565, 2024, doi: 10.1109/TNSRE.2024.3385336.

[8] D. Soydaner and J. Wagemans, “Multi-Task Convolutional Neural Network for Image Aesthetic Assessment,” *IEEE Access*, vol. 12, no. January, pp. 4716–4729, 2024, doi: 10.1109/ACCESS.2024.3349961.

[9] Y. He, Y. Zhang, S. Chen, and Y. Hu, “Facial Expression Recognition Using Hierarchical Features With Three-Channel Convolutional Neural Network,” *IEEE Access*, vol. 11, no. July, pp. 84785–84794, 2023, doi: 10.1109/ACCESS.2023.3303402.

[10] B. Fu, F. Boutros, C. T. Lin, and N. Damer, “A Survey on Drowsiness Detection: Modern Applications and Methods,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 11, pp. 7279–7300, 2024, doi: 10.1109/TIV.2024.3395889.

[11] H. Z. Ilmadina, D. Apriliani, and D. S. Wibowo, “Deteksi Pengendara Mengantuk dengan Kombinasi Haar Cascade Classifier dan Support Vector Machine,” *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 7, no. 1, pp. 1–7, 2022, doi: 10.30591/jpit.v7i1.3346.

[12] P. K. Singh, A. Gupta, M. Upadhyay, A. Jain, M. Khari, and P. S. Lamba, “Multimodal Driver Drowsiness Detection From Video Frames,” *Journal of Mobile Multimedia*, vol. 19, no. 2, pp. 567–586, 2023, doi: 10.13052/jmm1550-4646.19210.

[13] T. Sharma *et al.*, “Deep Learning-Based Object Detection and Classification for Autonomous Vehicles in Different Weather Scenarios of Quebec, Canada,” *IEEE Access*, vol. 12, no. September 2023, pp. 13648–13662, 2024, doi: 10.1109/ACCESS.2024.3354076.

[14] “View of Development of Smart Money Classifier & Location Tracker Application Based on Convolutional Neural Network (CNN) for The Visually Impaired.pdf.”

[15] Q. Zhu and X. Zu, “Fully Convolutional Neural Network Structure and Its Loss Function for Image Classification,” *IEEE Access*, vol. 10, pp. 35541–35549, 2022, doi: 10.1109/ACCESS.2022.3163849.

[16] R. Gelar Guntara, “Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55–60, 2023, doi: 10.47233/jteksis.v5i1.750.