An Experimental Study on Driver Drowsiness Detection System using DL

Venkatrajulu P[1], Balakotaiah D[2], Sai keerthana R[3] and Sai madhuharika R[4]
{venkatrajulup@gmail.com[1], dsbalu57@gmail.com[2], keerthanarachagarla77@gmail.com[3], madhuharika7@gmail.com[4] }

Department of CSE, VFSTR Deemed to be University, Guntur, Andhra Pradesh, India[1, 2, 3, 4]

Abstract. Driver drowsiness is one of the largest causes of automobile accidents worldwide, resulting in thousands of fatalities and injuries annually. Conventional methods of monitoring drowsiness using vehicle- mounted observation and biological sensors are constrained by their intrusiveness and environmental sensitivity. The developments in the deep learning technology, specifically Convolutional Neural Networks (CNNs), have allowed for the development of strong, non-invasive eye state moni- tors that can detect drowsiness in real-time. This present paper takes into account some of the most popular CNN models, like VGG-16, ResNet- 50, MobileNetV2, and EfficientNet-B0, to determine the top-performing model for real-time drowsiness detection. EfficientNet-B0 is also known as the best choice with its class-leading accuracy-computation ratio. The proposed system employs live video stream, image processing using OpenCV, and Softmax classification for the identification of excessive eye closure and sending early warnings as part of the initiative towards reducing fatigue-related accidents and road safety. The paper also com- pares the performance of different CNN models with regard to different metrics and describes their implications for field deployment. Finally, low-light detection problems, handling real-time behavior, and facial occlusion are addressed with suggestions towards improvement.

Keywords: Driver Drowsiness Detection, Deep Learning, EfficientNet- B0, VGG-16, ResNet-50, MobileNetV2, Image Preprocessing, Eye State Classification, Real-Time Monitoring, Data Augmentation, Adaptive Thresholding, Alarm System, Multi-Frame Validation, IoT Integration, Edge Computing

1 Introduction

Driver drowsiness has been well recognized as a significant risk factor in road safety, accounting for a large proportion of traffic accidents globally. Driver fatigue, characterized by reduced alertness, delayed reaction times, and reduced decision-making ability, is accountable for over 100,000 re- ported crashes annually, leading to serious injury and death [1]. The rev- elation of such a staggering figure makes the need for an effective driver drowsiness detection system to enhance road safety all the more urgent. Traditional detection systems, including vehicle-mounted observation systems and physiological sensors (e.g., EEG, ECG), as useful as they are, are hampered by their invasiveness, the cost of installation, and sensitivity to the environment [2]. These limitations have required the creation of non-invasive, vision-based driver-monitoring systems utilizing computer vision and deep learning advancements for real-time eye state observation and drowsiness alert [3].

ICITSM-Part I 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2357757

Deep learning and specifically Convolutional Neural Networks (CNNs) have transformed driver monitoring by allowing its real-time and precise examination of eye closures as well as facial moods. Although ResNet-50, MobileNetV2, and VGG-16 are some of the CNN-based models, more ac- curate and power-efficient alternatives were explored to provide balanced recommendations [4]. Objective: The aim of this paper is to minimize fatigue-related accidents by coming up with an efficient deep learning- based drowsiness detector that will be capable of sending timely notifications when it detects prolonged eye closure, hence reducing the prospective dangers involved in driver fatigue.

The organization of the paper is as follows: Section 2 reviews traditional challenges to prediction and recent advances in deep learning. Section 3 describes the approach taken to conduct the research, including preparation of data, pre-processing, and model choice. Section 4 compares the experimental results obtained using various performance metrics. Finally, section 5 summarizes the findings, comments on their implications, and suggests areas of future research.

2 Related work

Vural et al. [1] pointed out limitations in EEG-based driver drowsiness detection systems, more so pointing to the unpractically of installing EEG sensors under real-world operating conditions considering that they call for invasive installation procedures and user-dependent calibration. Authors explained how background conditions, i.e., environmental noise and head movement, as well as drivers' distraction by the external world, even limit the reliability of EEG-based approaches, which would need ideal background conditions to allow reliable detection.

Milan [2] spoke of the drawbacks of drowsiness detection systems based only on yawning or head tilts. Such indicators are not reliable as they may differ in the face structure and may cause erroneous detection. Milan referred to the issues with the detection of eyes, which becomes uncertain with varying levels of light. This affects system performance.

Magan et al. [3] discussed deep learning-based drowsiness detection and highlighted the significance of parameter optimization and rigorous cross-validation methods. They noted that most current systems are trained on small datasets, which impacts their generalizability when used in the conditions of real-world driving.

Parmar and Hiscocks [4] identified limitations in binarization-based detection systems, particularly their ineffectiveness in accurately detecting eye states in individuals with diverse skin tones. This issue arises due to the fixed thresholds used in binarization techniques, which fail to account for variations in lighting and facial features across different populations.

Prasath et al. [5] studied the environmental problems that impact the functioning of camera-based drowsiness detection systems, such as low lighting and glare. These issues tend to cause incorrect eye closures and facial recognition, lowering the system's overall reliability.

Jerith et al. [6] indicated that deep learning models employed in drowsiness detection tend to have difficulty with real-time applicability because they have high computational demands. This characteristic complicates using these models in environments that lack sufficient resources, including the embedded systems of vehicles.

Chirra et al. [7] observed that using eye-state features alone leads to incomplete evaluations since other behavioral cues, including head movements and facial expressions, are not considered. They mentioned that such systems can make inaccurate detections, especially when drivers show subtle drowsiness cues that are not evident in eye behavior.

Tyagi et al. [8] found that abrupt changes in road conditions, e.g., sharp turns and distraction, would influence the performance of drowsiness detection systems. They pointed out the shortcomings of single- static-threshold-based systems that can fail to cope with dynamic driving scenes.

Jahan et al. [9] elaborated on the limitations of SVM-based drowsiness detection systems, such as the difficulty in kernel selection and data limitations. They observed that the incorrect choice of kernel can result in poor model performance and lower detection accuracy.

Faisal et al. [10] pointed out the performance issues in detecting drowsiness at night due to constrained dataset diversity and unavailability of night vision cameras. They indicated that conventional camera-based systems cannot effectively detect facial features under poor lighting, thus affecting the reliability of overall detection.

Shahrudin et al. [11] pointed out user discomfort with intrusive approaches, like ECG-based detection, that involve sensors being strapped to the driver's body. They pointed out that such approaches, though ac- curate, are impractical for extended use because they are inconvenient.

Jabbar et al. [12] have indicated that CNN-based models performed moderately well in drowsiness detection but needed more parameter tuning to make them effective for real-world use. They stated that the performance of such models is typically affected by the small size and limited diversity of the training sets.

Arunasalam et al. [13] found inadequate accuracy in current drowsiness detection systems and pointed out the inadequacy of effective alert mechanisms to notify drivers in real-time. They found that most systems alert too late, which lowers their effectiveness in avoiding drowsiness-related accidents.

Lyu et al. [14] alluded to the problem of dataset diversity and parameter optimization, mentioning that there is a need for careful testing in order to improve the detection performance. They elucidated that most existing models are learned from representative datasets that do not reflect real driving conditions, thus leading to low generalizability.

Adarsh et al. [15] set forth the problems of varying lighting conditions, camera orientation, and face occlusions, which are accountable for most of the job of maintaining the drowsiness detection systems dependable. They recognized that the traditional systems cannot detect small facial expressions if these environmental factors are there.

2.1 Limitations

Detection accuracy is determined by external conditions such as light- ing conditions, weather changes, and driver-specific behaviors, leading to performance inconsistencies Vural et al. [1], Mag´an et al. [3], Jerith et al. [6]

High computational complexity and resource demands make real-time implementation challenging, requiring optimized models and efficient hardware Faisal et al. [10], Jabbar et al. [12]

Dataset limitations, including lack of diversity and insufficient real- world testing, reduce model robustness and generalizability across different driving scenarios. Lyu et al. [14], Adarsh et al. [15].

3 Methodology

The project includes real-time eye state detection using the application of deep learning methods. Actions include data recording, preprocessing, model training, testing, and real-time execution. The data set includes open and closed eyes and are labeled and utilized as training for deep learning classifier models. Performance is evaluated with accuracy, precision, recall, F1-score, and confusion matrix to identify the best per- forming model to be used in real-time detection and drowsiness monitoring. Furthermore, the system incorporates an alerting mechanism that produces a warning when drowsiness is identified for successive frames. Fig 1 shows the proposed flowchart.

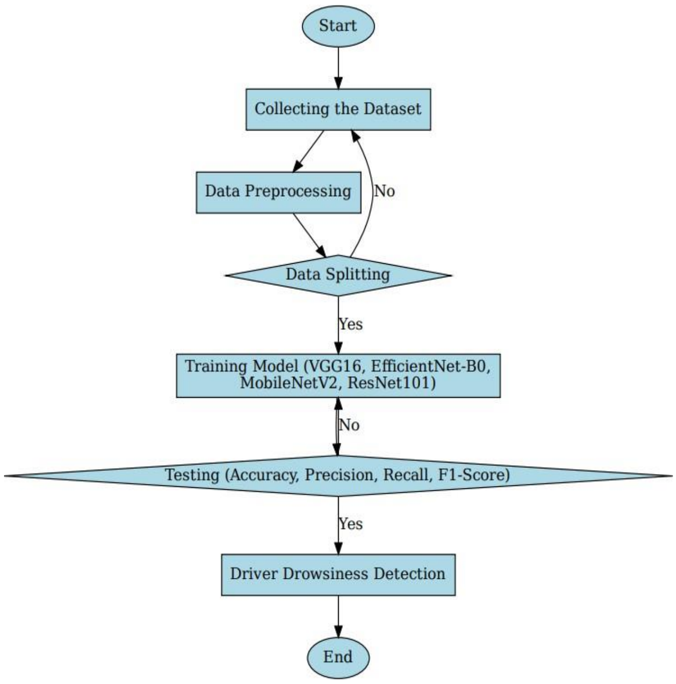

Fig. 1. Proposed workflow for driver drowsiness detection.

3.1 Data Collection:

A human eye image database is used, which is divided into two classes:

- Open eyes
- Closed eyes

It is derived from the public domain and augmented with further samples for added solidity.

3.2 Pre-processing Data:

Image Preprocessing to ensure high-quality inputs to the deep learning models, the following preprocessing methods are applied:

- Resize: Each of the images is resized uniformly to 32×32 pixels.
- Normalize: Values of the pixel range from [0,1] to speed up convergence:
- Data Augmentation: ±15° rotation compensates for head movement. Brightness controls to best fit varied lighting. Horizontal flipping is adapted for extra data set variability. Hot Encoding for categorical labeling of classes.

3.3 Data Splitting:

After preprocessing, the dataset is fragmented into training and testing subsets. Therefore, the subsets. Therefore, the model tested on data that has not been observed yet gives a fair performance rating. It is common to train at 80 percent and set aside the remaining 20 percent for testing.

3.4 Training:

Deep learning models namely EFFICIENTNETB0, RES- NET101, MOBILENETV2 and VGG16 are employed in training the net- work. They learn from the training data to establish models that may predict energy output trends based on historical trends.

3.5 Testing:

The test data set to gauge the important performance metric. They all are accuracy, precision, recall, and F1 score. Metrics like Mean Absolute Error, Mean Squared Error and Root Mean Squared Error estimate how well the models generalize to novel test data.

- **Mean Absolute Error (MAE):**
  Average magnitude of errors, ignoring the direction, whether it is negative or positive. It is an easy measure that provides an intuitive view of how far the predictions are from the truth's corresponding values.
  $$MAE = \frac{1}{n}\sum_{l=1}^{m}|u_l - \widehat{u_l}| \tag{1}$$
- **Mean Squared Error (MSE):**
  MSE estimates the mean of the squared differences between actual and predicted values. Since it squares the errors, it generally biases towards larger errors, which is a good measure for using significant deviation penalties.
  $$MSE = \frac{1}{k}\sum_{j=1}^{k}\left(y_j - \widehat{y_j}\right)^2 \tag{2}$$
- **Root Mean Squared Error (RMSE):**
  RMSE is the square root, giving an error measure in the same units as the predicted output. It penalizes large errors greatly and gives a far more sensitive measure than MAE if larger deviations are a concern.
  $$RMSE = \sqrt{\frac{1}{k}\sum_{j=1}^{k}\left(y_j - \widehat{y_j}\right)^2} \tag{3}$$

3.6 Driver drowsiness detection:

Drowsiness of drivers is forecast by the model through eye state classification. The data is critical for road safety because it aids in preventing accidents through real-time warning of drivers, thus reducing the risk of drowsiness crashes.

3.7 Architecture of VGG16

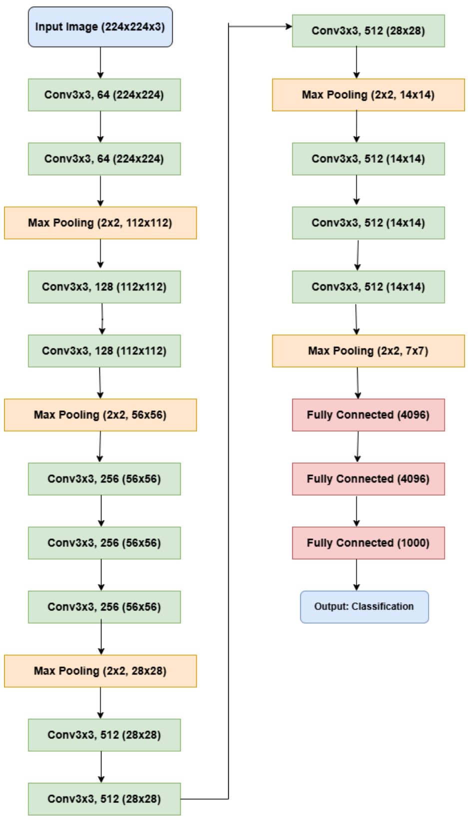

Fig. 2. Architecture of VGG16 model.

VGG16 is a 16-layer deep convolutional neural network (CNN) model as shown in fig 2 put forward by the Visual Geometry Group (VGG) of the University of Oxford. It is extensively utilized for image classification and feature ex- traction. The model is specifically renowned to employ small-sized 3×3 filters in the convolutional layers, hence the model is capable of recognizing detailed patterns within images and yet has a basic structure.

Convolutional Layer:

$$\boldsymbol{C_O = \frac{(C_I - C_K + 2C_P)}{C_S} + 1} \tag{4}$$

where:

CO = Output size (height/width)

CI = Input size (height/width)

CK = Kernel (filter) size

CP = Padding

CS = Stride

Max-Pooling Layer Formula

$$O = \frac{(J-L)}{T} + 1 \qquad (5)$$

where:

O = Output size

J = Input size

L = Pooling window size

T = Stride

Fully Connected Layer Formula

$$O = W \cdot X + B \qquad (6)$$

where:

CO = Output vector

CW = Weight matrix

CX = Input vector

CB = Bias vector

RELU:

$$f(x) = \max(0, x) \qquad (7)$$

SOFTMAX:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \qquad (8)$$

zi is the input of the Softmax function for class i.

Alarm Trigger Condition for Drowsiness Detection

$$\sum_{t=1}^{T}(Prediction_t = Closed) \geq Threshold \qquad (9)$$

Where:

Prediction t is the model's output at frame t.

T is the consecutive time frames checked.

3.8 Architecture of MobileNetV2

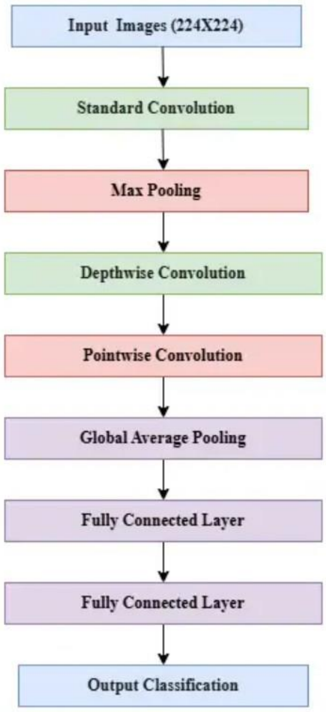

Fig. 3. Architecture of mobileNetV2.

MobileNetV2 is a resource-efficient deep CNN optimized for mobile and embedded systems to execute. It is commonly used for image classification, object detection, and real-time video processing because it is efficient. Depth wise separable convolutions are employed in MobileNetV2, which reduces the computational expense and increases speed without any fall in accuracy. Fig 3 gives the architecture of mobilenetv2.

Depth wise Separable Convolution Formula:

$$C_O = \frac{(C_I - C_K + 2C_P)}{C_S} + 1 \tag{10}$$

Where:

CO: Output size, height/width

CI: Input size, height/width

CK: Kernel size, filter

CP: Padding

CS: Stride

Max-Pooling Layer Formula The equation to calculate the output size in a max-pooling layer is:

$$C_O = \frac{(C_I - C_K)}{C_S} + 1 \tag{11}$$

where:

CO: Output size, height/width

CI: Input size, height/width

CK: Pooling window size

CS: Stride

Softmax Function The softmax function for classifying probabilities

$$\boldsymbol{\sigma}(\boldsymbol{z_i}) = \frac{\boldsymbol{e^{z_i}}}{\sum_j \boldsymbol{e^{z_j}}} \tag{12}$$

Alarm Trigger Condition for Drowsiness Detection To trigger an alarm when drowsiness is detected, the following condition is used:

$$\sum_{t=1}^{T} (Prediction_t = Closed) \geq Threshold \tag{13}$$

3.9 Architecture of Efficient-netB0

EfficientNet-B0 is a member of the EfficientNet model series, which is best known for its scalability and resource-efficient use. EfficientNet- B0 is highly accurate with fewer parameters since the compound scaling method is employed for depth, width, and resolution optimization. EfficientNet-B0 applies a variety of applications like object detection in real-time, face recognition, and image classification. Fig 4 gives the architecture of efficient netB0.

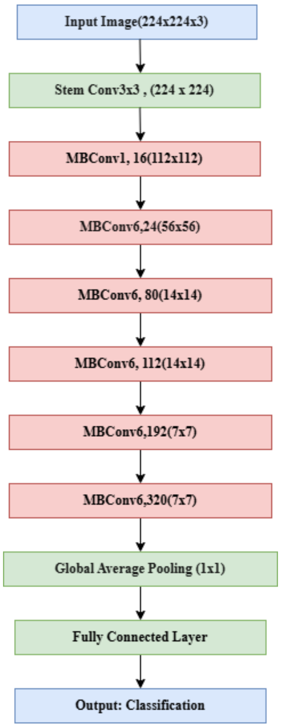

Fig. 4. Architecture of efficient-netB0.

Compound Scaling Formula: With a compound coefficient, EfficientNet-B0 deepens the network's depth, width, and resolution. ϕ:

$$CA = \alpha\phi, Cq = \beta\phi, Cl = \gamma\phi \qquad (14)$$

Where:

CA: Network depth—number of layers

Cq: Network width—number of channels per layer

Cl: Input resolution

α, β, γ: Scaling coefficients

ϕ: Compound scaling factor

Depthwise Separable Convolution Formula: EfficientNet-B0 uses depthwise separable convolutions, calculated as:

$$\boldsymbol{C_O} = \frac{(\boldsymbol{C_I} - \boldsymbol{C_K} + \boldsymbol{2C_P})}{\boldsymbol{C_S}} + \boldsymbol{1} \qquad (15)$$

Where:

CO: Output size, height/width

CI: Input size, height/width

CK: Kernel size, filter

CP: Padding

CS: Stride

Squeeze-and-Excitation Formula: EfficientNet-B0 uses squeeze- and-excitation (SE) layers to improve feature extraction:

$$X_{out} = \sigma(W_2\delta(W_1X_{in})) \cdot X_{in} \tag{16}$$

Where:

Xin: input vector

W1, W2: weight matrices

δ: ReLU activation

σ: sigmoid activation

Activation Function (Swish): EfficientNet-B0 uses the Swish activation function:

$$f(x) = x \cdot \sigma(x) \tag{17}$$

Where:

σ(x): Sigmoid activation, σ (x) = $-\frac{1}{1+e-cz}$

Softmax Function: The softmax function for classifying probabilities:

$$\boldsymbol{\sigma}(\boldsymbol{z_i}) = \frac{\boldsymbol{e^{z_i}}}{\sum_j \boldsymbol{e^{z_j}}} \tag{18}$$

Alarm Trigger Condition for Drowsiness Detection: An alarm will trigger when the number of consecutive" Closed" frame classifications reach the defined threshold:

$$\sum_{t=1}^{T}(Prediction_t = Closed) \geq Threshold \tag{19}$$

3.10 Architecture of Resnet50

ResNet-50 represents a deep convolutional neural network consisting of 50 layers known for implementing residual connections to overcome training limitations in deep architectures. the vanishing gradient issue in deep networks. The application scope of ResNet-50 extends to regular image classifying as well as detecting objects and recognizing faces. The model benefits from residual connections since they enable learning of identity mappings which simplifies training deep networks.

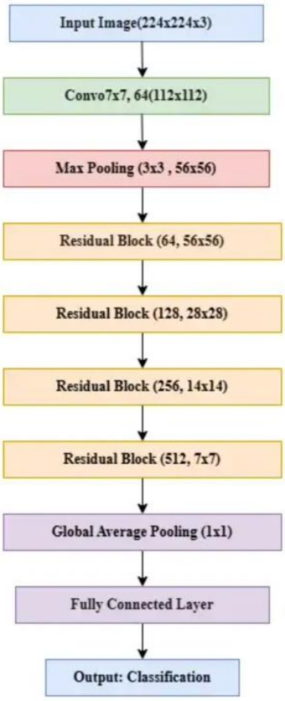

Fig. 5. Architecture of Resnet-50.

Fig 5 ResNet-50 architecture has stacked residual blocks, two or three layers of convolution with a skip connection each. These allow the network to bypass some of the layers without breaking gradient flow.

Residual Block Formula: ResNet-50 uses residual connections to learn identity mappings, which help preserve gradient flow:

$$Xout = Xin + F\,(Xin) \tag{20}$$

Where:

Xout: Output vector

Xin: Input vector

F (Xin): Residual mapping (the learned transformation)

Convolutional Layer formula: The formula to calculate the output size of a convolutional layer is:

$$\boldsymbol{C_O = \frac{(C_I - C_K + 2C_P)}{C_S} + 1} \tag{21}$$

Where:

CO: Output size (height/width)

CI: Input size (height/width)

CK: Kernel (filter) size

CP: Padding

CS: Stride

Max-Pooling Layer Formula: The output size of a max-pooling layer is calculated as:

$$C_O = \frac{(C_I - C_K)}{C_S} + 1 \tag{22}$$

Where:

CO: Output size (height/width)

CI: Input size (height/width)

CK: Pooling window size

CS: Stride

Fully Connected Layer Formula: The output vector in a fully connected dense layer is computed as :

$$CO = CW \cdot CX + CBs \tag{23}$$

Where:

CO: Output vector

CW: Weight matrix

CX: Input vector

CBs: Bias vector

Softmax Function: The Softmax function for classifying probabilities is:

$$\boldsymbol{\sigma}(\boldsymbol{z_i}) = \frac{\boldsymbol{e^{z_i}}}{\sum_j \boldsymbol{e^{z_j}}} \tag{24}$$

Alarm Trigger Condition for Drowsiness Detection: An alarm will trigger when the number of consecutive" Closed" frame classifications reaches the defined threshold

$$\sum_{t=1}^{T}(Prediction_t = Closed) \geq Threshold \tag{25}$$

4 Experimental Results and Discussion

4.1 About Dataset

The data set includes images of eyes (open or close) that were captured for driver drowsiness detection taking into account the state transition of the eyes for measuring the alertness levels. The data set includes varied samples captured under various light, angle, and orientation of the face to enhance generalizability in the model. The data set exists as labeled images and therefore, supervised learning and proper classification of pat- terns of drowsiness is possible. The dataset is also metadata-rich, including timestamps and ambient conditions, and can be helpful for secondary analysis. The dataset serves as an essential component which guides deep learning models particularly CNNs toward detecting real-time drowsiness as part of their training process. Analysis of this dataset supports safe driving through smart vehicles thus helping to avoid accidents stemming from driver fatigue.

4.2 Evaluation Metrics

The performance analysis requires accuracy, precision, recall, and F1- score metrics for assessment of our driver drowsiness detection system. Accuracy is the ratio of correctly classified eye states (open or closed) that determine the model's overall effectiveness. Precision measures the ratio of true drowsiness detections among all predicted drowsy states so that the model maintains minimal false alarms. Recall is the degree to which the system identifies all actual cases of drowsiness, keeping the chances of missing critical cases to a minimum. The F1-score is a trade-off between precision and recall, providing a general estimate of the reliability of the model in day-to-day life applications, i.e., monitoring drivers to enhance road safety.

- Precision: A model's precise measurement appears as the ratio of its correct positive predictions to all total positive predictions. positives to the total number of positive predictions of the model. The ratio demonstrates which proportion of predicted positive outcomes turns out to be accurate. High levels of precision matter in scenarios where wrong positive identifications produce substantial expenses.

$$\text{Precision} = \frac{C_T ruePositive}{C_T ruePositive + C_F alsePositive} \tag{26}$$

Recall: Recall, or sensitivity, is the ratio of true positive receiver classifies all actual positives as such. This would imply that the model can return all positive class instances.

$$\text{Recall} = \frac{C_T ruePositive}{C_T ruePositive + C_F alseNegative} \tag{27}$$

- F1-Score: It is the harmonic mean of precision and recall so that it could balance the two measures. It can be especially useful in imbalanced datasets wherein the essential levels of precision and recall are unlike each other. A higher F1 score often refers to a better general performance in terms of both precision and recall.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{28}$$

- Accuracy: Accuracy measures how well the model correctly predicted what was true and not true, true positives and true negatives about all predictions. It is widely applied to classification tasks as an estimate of overall performance. It will not help to identify imbalanced datasets.

$$Accuracy = \frac{C\,TP + C\,TN}{C\,TP + C\,TN + C\,FP + C\,FN} \tag{29}$$

4.3 Model Performance Summary

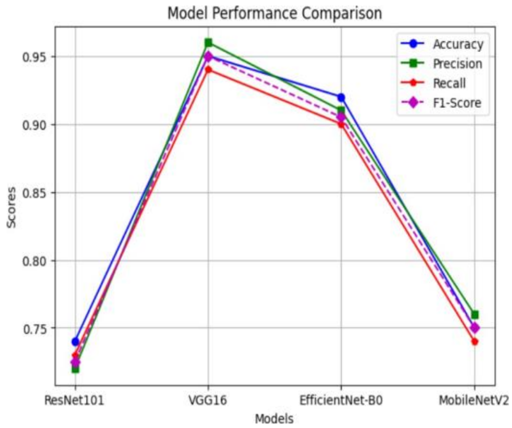

Fig. 6. Evaluation metrics Comparison of Models.

Fig. 6 illustrates the performance of ResNet101, VGG16, EfficientNet- B0, and MobileNetV2 based on accuracy, precision, recall, and F1-score. The best performance is achieved by VGG16, followed by EfficientNet-B0. ResNet101 and MobileNetV2 are less effective, so the most appropriate model is VGG16.

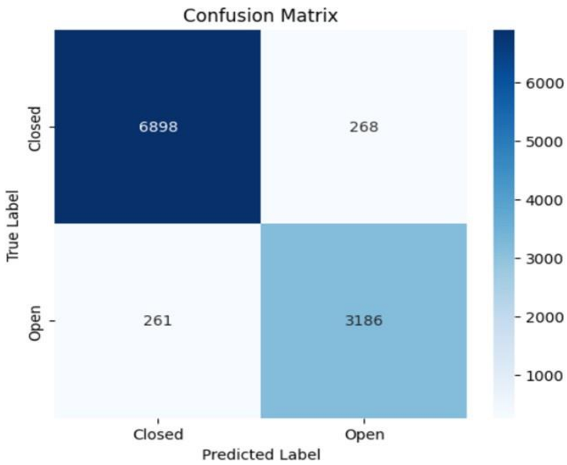

Fig. 7. Classification report for VGG16 model.

Fig. 7 The model demonstrates strong performance in distinguishing between open and closed eyes based on data in the confusion matrix. The model accurately labels 6,898 closed-eye images and 3,186 open-eye images with few misclassifications of 268 and 261, respectively, showing excellent performance.

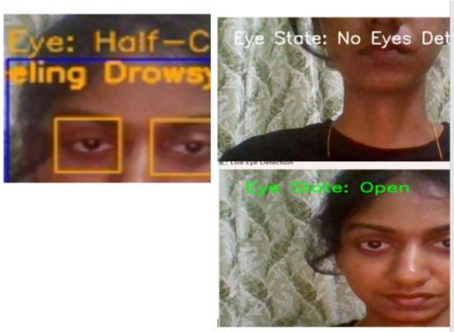

Fig. 8. Driver drowsiness detection system in action.

Fig. 8 This figure illustrates the driver drowsiness detection system in action, showing different eye states captured in real time. The images rep- resent three conditions: half-closed eyes (drowsy state), no eyes detected, and fully open eyes (awake state).

4.4 Discussion

Accuracy for real-time driving drowsiness detection was verified for VGG16, EfficientNet-B0, MobileNetV2, and ResNet-50. The best was obtained for VGG16 as 95% strongly signified its excellent feature extraction capability as well as effective eye state classifying capacity. EfficientNet-B0 ranked as second-best at 92% accuracy with appropriate performance-compute balance. MobileNetV2 (75%) was observed to be perfect for edge devices and smartphones due to its lightness, whereas ResNet-50 (74%) performed reasonably but consumed more processing power and is not suitable for day-to-day activities.

VGG16 achieved the superior precision-recall ratio according to the confusion matrix and evaluation metrics, which made it the optimal model to detect live drowsiness. The high accuracy results of EfficientNet-B0 occurred alongside minimal resource requirements. MobileNetV2 offers an exceptional capability for embedded systems that need low-power performance even if its accuracy level is lower than some models. The deep architecture of ResNet-50 produced signs of overfitting because of its excessive depth. Eye state classification succeeds exceptionally through deep learning approaches, but the most suitable choice depends on attaining either better computational performance or higher accuracy that fits real- time application needs.

5 Conclusion and Future Work

The research project implemented a real-time system for driver drowsiness detection that used deep learning models VGG16, EfficientNetB0, MobileNetV2, and ResNet-50. The testing of these models validated that VGG16 (95%) yielded the highest accuracy and reliability, while Effi- cientNetB0 (92%), MobileNetV2 (75%), and ResNet-50 (74%) were good but needed optimization for real-time use. The system efficiently detects drowsiness by constantly monitoring eye states and providing reminders when it detects extended eye closure. The combination of image preprocessing, feature extraction, and deep learning-based classification ensures robust prediction accuracy and reliability for the prevention of drowsy driving-caused accidents.

Future work: Future work will emphasize the combination of Transformerbased models like Vision Transformers (ViTs) to ensure improved classification efficiency and accuracy. Real-time processing optimization techniques will be considered to reduce computational latency and enhance performance on embedded automotive platforms. In addition, the data set will be enriched for multi-angle facial tracking, low-light environments, and occlusion management to determine strong generalizability in many real-world environments. Deployment of the system within car systems using Edge AI solutions will allow real-time identification and processing and support improved road safety and convenient real-world functionality.

References

[1] Vural, Esra, et al. ”Drowsy driver detection through facial movement analysis.” in- ternational workshop on human-computer interaction. Berlin, Heidelberg: Springer Berlin Heidelberg,2007.

[2] Milan, A. ”DRIVER’S DROWSINESS DETECTION.”(2022).

[3] Maga´n, Elena, et al. ”Driver drowsiness detection by applying deep learning tech- niques to sequences of images.” Applied Sciences 12.3(2022):1145.

[4] Parmar, Neeta, and P. Hiscocks. ”Drowsy driver detection system.” Engineering Design Project Thesis, Ryerson University (2002).

[5] Prasath, N., J. Sreemathy, and P. Vigneshwaran. ”Driver drowsiness detection using machine learning algorithm.” 2022 8th International conference on advanced computing and communication systems (ICACCS). Vol. 1. IEEE, 2022.

[6] Jerith, G. Gifta, et al. ”Driver Drowsiness Detection Using Deep Learning.”

[7] Chirra, Venkata Rami Reddy, Srinivasulu Reddy Uyyala, and Venkata Krishna Kishore Kolli. ”Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State.” Rev. d’Intelligence Artif. 33.6 (2019): 461-466.

[8] Tyagi, Divyanshu, et al. ”Real time ‘driver drowsiness’ & monitoring & detec- tion techniques.” International Journal of Innovative Technology and Exploring Engineering 9.8 (2020): 280-284.

[9] Jahan, I., et al. ”4D: A Real-Time Driver Drowsiness Detector Using Deep Learn- ing. Electronics 2023, 12, 235.” 2023,

[10] Faisal, Tarig, et al. ”Systematic development of real-time driver drowsiness detec- tion system using deep learning.” IAES International Journal of Artificial Intelli- gence 11.1 (2022): 148.

[11] Shahrudin, NS Nor, and K. A. Sidek. ”Driver drowsiness detection using different classification algorithms.” Journal of Physics: Conference Series. Vol. 1502. No. 1. IOP Publishing,2020.

[12] Jabbar, Rateb, et al. ”Driver drowsiness detection model using convolutional neural networks techniques for android application.” 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT).IEEE,2020.

[13] Arunasalam, M., et al. ”Real-time drowsiness detection system for driver monitor- ing.” IOP Conference Series: Materials Science and Engineering. Vol. 767. No. 1. IOP Publishing, 2020.

[14] Lyu, Jie, Zejian Yuan, and Dapeng Chen. ”Long-term multi-granularity deep framework for driver drowsiness detection.” arXiv preprint arXiv:1801.02325 (2018).

[15] Adarsh, Gauri, et al. ”Drowsiness detection system in real time based on behavioral characteristics of driver using machine learning approach.” Journal of Informatics Electrical and Electronics Engineering (JIEEE) 4.1 (2023):1-10.