Evaluation of Convolutional Neural Network Architectures for Detecting Drowsiness in Drivers

Mario Aquino Cruz, Bryan Hurtado Delgado, Marycielo Xiomara Oscco Guillen
Departamento Académico de Informática y Sistemas, Universidad Nacional Micaela Bastidas de Apurímac, Abancay, Perú

Abstract—**Drowsiness in drivers is a condition that can manifest itself at any time, representing a constant challenge for road safety, especially in a context where artificial intelligence technologies are increasingly present in driver assistance systems. This paper presents a comparative evaluation of convolutional neural network (CNN) architectures for drowsiness detection, focusing on the identification of signals such as eye state and yawning. The research was of an applied type with a descriptive level, comparing the performance of LeNet, DenseNet121, InceptionV3 and MobileNet under challenging conditions, such as lighting and motion variations. A non-experimental design was used, with two datasets: a public dataset from Kaggle that included images classified into two categories (yawn and no yawn) and another created specifically for this study, which included images classified into three main categories (eyes open, eyes closed and undetected). The results indicated that, although all architectures performed well in controlled conditions, MobileNet stood out as the most accurate and consistent in challenging scenarios. DenseNet121 also showed good performance, while LeNet was effective in eye-state detection. This study provided a comprehensive assessment of the capabilities and limitations of CNNs for applications in drowsiness monitoring systems, and suggested future directions for improving accuracy in more challenging environments.**

Keywords—Architectures; detection; drowsiness; neural networks

I. Introduction

Drowsiness at the wheel is a common problem that negatively impacts road safety worldwide. It is defined as the biological need for sleep, which can be caused by fatigue, sleep deprivation or medical conditions, affecting the driver's concentration and reaction time [1]. This condition poses a considerable danger to drivers, as it severely limits their ability to respond, increasing the risk of accidents, especially during long or night-time journeys. Moreover, it compromises not only the safety of the driver, but also that of passengers and other road users [2]. This problem is more serious in situations such as long working hours or lack of sleep.

According to the World Health Organization (WHO), road traffic accidents are responsible for approximately 1.19 million deaths each year, with men being three times more likely than women to lose their lives in these incidents [3]. An analysis of 208,727 passenger vehicle records involved in fatal crashes in the United States between 2017 and 2021 revealed that 17.6% of the cases were related to drowsy drivers [4]. In Spain, the Dirección General de Tráfico (DGT) stated that between 15% and 30% of vehicle accidents are directly or indirectly caused by drowsiness [5]. This problem is not only evident in developed countries, but also generates concern in Latin American countries such as Peru, which according to the National Institute of Statistics and Informatics (INEI) of Peru, 116,659 traffic accidents were recorded in 2016, of which 0.97% (1,131 accidents) were attributed to driver tiredness or fatigue, with Lima being the most affected region with 813 accidents, followed by Puno with 38 and Arequipa with 34 [6]. These data highlight the need to implement preventive measures, especially in areas with high accident rates due to fatigue.

To address this problem, recent advances in artificial intelligence have enabled the development of automated driver monitoring systems, Convolutional Neural Networks (CNNs) are advanced artificial intelligence models widely used in tasks such as image classification, segmentation, object detection and video processing. Such models are composed of several layers, such as the convolution layer, the clustering layer and the fully connected layer, which allow extracting essential features from the input data [7], these deep learning networks have gained recognition for their ability to process images with high efficiency and detect complex patterns [8], which makes them particularly suitable for drowsiness detection applications.

The objective of this research project was to evaluate and compare different CNN architectures for drowsiness detection in drivers. In particular, we sought to determine which of these architectures offered the best performance in the identification of drowsiness cues, such as the state of the eyes (open, closed or undetected) and the presence of yawning. For the yawning state, images of drivers in different states were used according to the dataset in the Kaggle database [9], and a new dataset specific to the state of the eyes was constructed, which included images in a wide variety of situations. The training was performed with 4 convolutional neural network architectures: LeNet [10], DenseNet121 [11], InceptionV3 [12], MobileNet [13]. The architectures were selected for their diversity, ranging from simple and efficient models to more advanced and specialized ones.

All research has its limits, and this study is no exception. Although the models evaluated have demonstrated high performance in drowsiness detection in stable conditions, their effectiveness can be affected by external factors, such as abrupt changes in lighting and extreme variations in the driver's posture, which can reduce their accuracy in more challenging scenarios. In addition, the availability of computational resources was a determining factor in the training process, as the use of a Google Colab Pro account was practically indispensable for reasonable processing times.

The main contributions of this study consist of the comparative evaluation of four CNN architectures to determine their strengths and limitations in drowsiness detection. In addition, two datasets have been worked with the purpose of improving generalization and robustness in various driving conditions. Finally, real-time tests were carried out to analyze the practical applicability of the models in real-world scenarios.

The remainder of this paper is organized as follows: Section II discusses related work on drowsiness detection using CNN and other artificial intelligence techniques. Section III describes the methodology, including details of the datasets, preprocessing steps, and training setup. Section IV presents the experimental results and performance comparisons. Section V discusses the findings and their implications. Finally, Section VI concludes the study and raises possible future directions for improving drowsiness detection systems.

II. Related Works

Recent studies have extensively explored the use of artificial intelligence for traffic accident prevention. For example, Ma, Chau, and Yap [14] focused on developing a fatigue detection system for nighttime driving conditions using in-depth video sequences, overcoming the limitations of RGB video-based systems in low-light environments. Their goal was to leverage Kinect sensor data to detect signs of fatigue such as yawning and posture changes. They used a Two-stream CNN architecture that includes a spatial stream to capture static features, such as driver posture, and a temporal stream to analyze changes between frames, using motion vectors instead of dense optical flow. The results of both flows were combined using an SVM classifier. The system achieved an accuracy of 91.57%, significantly outperforming systems using only RGB video in daylight environments.

Zhao et al. [15] designed a CNN-based algorithm to detect driver fatigue by analyzing the state of the eyes and mouth using the Eye and Mouth CNN (EM-CNN) network. The methodology included face and facial point detection using Multitask Cascaded Convolutional Network (MTCNN) to extract the regions of interest (ROI) of eyes and mouth. Subsequently, EM-CNN classified whether the eyes and mouth were open or closed, and the indicators percent eye closure time (PERCLOS) and percent mouth opening (POM) were calculated to assess fatigue. The results showed an accuracy of 93.623%, with a sensitivity of 93.643% and a specificity of 60.882%, demonstrating high effectiveness in detecting fatigue in a real driving environment.

Li, Gao and Suganthan [16] developed an advanced system for driver fatigue recognition using electroencephalography (EEG) signals, which reflect brain activity and exhibit high inter-subject variability, complicating cross-recognition tasks. Their goal was to improve the ability of the models to extract more distinguishable features from the decomposed EEG signals. The proposed methodology included decomposing the signals into components of different frequency bands using techniques such as discrete wavelet transform (DWT), empirical wavelet (EWT), empirical mode decomposition (EMD) and variational mode decomposition (VMD). These components were processed by independent CNNs with a Component-Specific Batch Normalization (CSBN) layer for each component to reduce inter-individual variability. The model, a hybrid ensemble of CNNs, was evaluated on cross-recognition tasks, achieving an average accuracy of 83.48%, with the DWT-based model being the most effective, outperforming existing approaches by more than 5%.

Chirra, Uyyala y Kolli [17] designed a system based on Deep CNN with the aim of detecting drowsiness in drivers based on the analysis of the state of the eyes. To achieve this, they used the Viola-Jones algorithm to detect the face and extract the eye region as the ROI. Subsequently, they applied a stacked CNN architecture including four convolution layers, each followed by normalization, ReLU activation and MaxPooling, which allowed them to extract relevant features from the images. Finally, a SoftMax layer classified the driver as drowsy or non-drowsy. The model was trained with 1200 images and evaluated with 1150, reaching an accuracy of 96.42%. This approach overcame the limitations of traditional CNN methods, which presented difficulties in pose accuracy during regression, thus demonstrating high effectiveness in accurately detecting drowsiness in drivers.

Flórez [18] designed a real-time drowsiness detection system using computer vision, using CNN to analyze visual features such as eye closure and yawning. The goal of his research was to develop an efficient model for drowsiness detection in drivers. He evaluated several CNN architectures, such as InceptionV3, VGG16 and ResNet50V2, as well as two custom models, DD-AI and DD-AI-G, adapted for implementation on an NVIDIA Jetson Nano device. The results obtained in simulated and real environments showed an accuracy of 91.48% in simulations and 86.28% in real driving conditions, with the DD-AI-G model standing out for its superior performance.

III. Methodology

A. Type and Level

This research was of an applied type, as it focused on implementing and evaluating CNN architectures for drowsiness detection in drivers. It was descriptive, since it evaluated and compared the performance of different CNN architectures without developing new theories, using quantitative metrics for its evaluation Study design.

B. Study Design

This study employed a non-experimental design, since the independent variables were not manipulated, but observed in their performance under controlled conditions. Likewise, a cross-sectional design was used, collecting and analyzing the data at a specific moment in time. A quantitative approach was used, measuring the performance of the models through metrics such as accuracy, recall and F1-score.

C. Sample

Two datasets were used for the research. The first was a public dataset from the Kaggle platform, which contains images classified in the categories of yawning ('yawning') and non-yawning ('no_yawning') drivers. The second dataset was created specifically for this research, with images classified into three main categories: eyes open ('open'), eyes closed ('closed') and images where the eyes were not detected ('no_detected').

The images collected for the second dataset include a variety of features, such as eyes with makeup, eyes of different ethnicities (Caucasian, Asian, Afro-descendant and Latin American), with lenses (cool, warm or neutral colors) and without lenses, as well as different eye colors (very dark, medium dark, warm dark, warm light, and cool light). In addition, variations such as irritated eyes, aged eyes and eyes looking away from the eye were included.

On the other hand, images classified as 'non-detected' comprise cases in which the eyes were not detected due to factors such as dark or reflective lenses, obstructions by hair, hats, caps or hands, inadequate lighting (very bright or dark), blurring (total or slight), or artistic make-up that confuses the processing.

The inclusion criteria for both datasets were images of acceptable quality for neural network processing.

The sample size includes two datasets: a public dataset from Kaggle, consisting of 2892 images distributed in 2083 for training, 519 for validation and 290 for testing; and a proprietary dataset, with a total of 4593 images, of which 3307 are for training, 825 for validation and 461 for the test set.

D. *Procedure*

This study started with data preprocessing, where the original dataset of the yawning state, represented in Fig. 1, was modified by duplicating the images by flipping them vertically and converting them to black and white. The conversion to black and white helped to improve performance on nighttime images by removing the color ranges present during the day and allowing CNNs to perform better predictions of both daytime and nighttime images. In addition, a green facial mesh was applied to the drivers' faces using the MediaPipe library and its FaceMesh function, which allowed the image to be cropped and focused exclusively on the facial region. These modifications were implemented after evaluating that this format offered better results in previous tests.

Fig. 1. Images of the original yawning state dataset: yawn and no_yawn.

General preprocessing was performed for the eye status dataset and the yawning status dataset as shown in Fig. 2 and Fig. 3. This process included normalizing the pixel values, scaling them from 0 to 1, and resizing the images to 224x224 pixels. In addition, the data were organized into batches of 64 and sorted. For eye status (“open”, “closed” and “no_detected”), categorical classification was used, while for yawning status (“yawn” and “no_yawn”), binary classification was employed.

For eye and yawning status, approximately 70.0% of the images were separated for training, 20.0% for validation, and 10.0% of the images for testing.

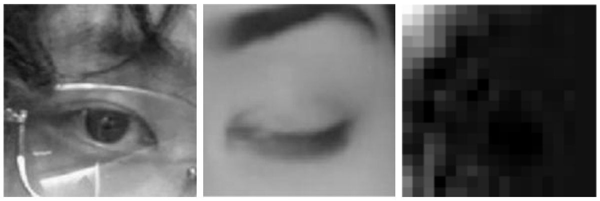


Fig. 2. Images of the preprocessed dataset of the eye status: open, closed and no-detected.

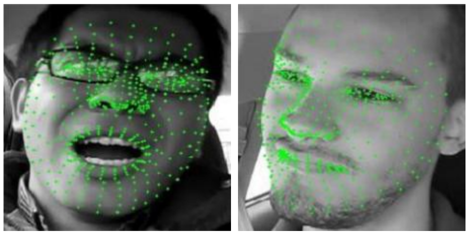


Fig. 3. Images of the preprocessed yawning state dataset: yawn and no_yawn.

For the configuration of architectures, we used the optimizer Adam, using a learning rate of 0.001, and the loss function categorical_crossentropy for the eye state and binary_crossentropy for the yawning state. For evaluation, the accuracy metric was used to monitor performance in each epoch. In addition, the L2 regularization technique (kernel_regularizer=l2(0.01)) was applied to the fully connected layers and a 50% dropout layer was added to prevent over-fitting.

The modifications implemented in each architecture are illustrated in Fig. 4, Fig. 5, Fig. 6 and Fig. 7, showing the structure of LeNet, DenseNet121, InceptionV3 and MobileNet with the optimization techniques applied.

In order to optimize the training process, the Early Stopping technique was applied, which ended the training if the loss in the validation set did not improve after five consecutive epochs. And in case the training was terminated, the weights corresponding to the best performance were restored.

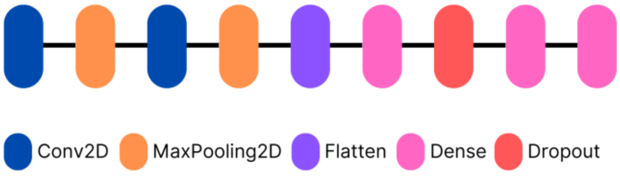


Fig. 4. Structure of the LeNet architecture with optimization techniques.

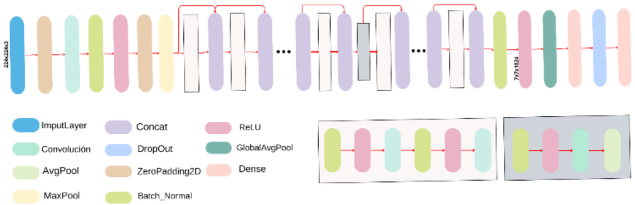


Fig. 5. Structure of the DenseNet121 architecture with optimization techniques.

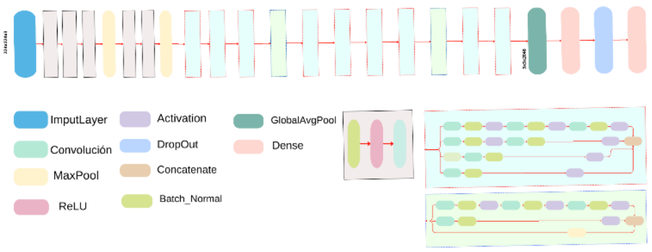


Fig. 6. Structure of the InceptionV3 architecture using optimization techniques.

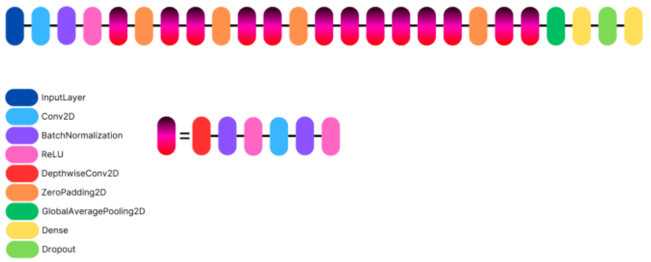


Fig. 7. Structure of the MobileNet architecture with optimization techniques.

Next, each convolutional neural network architecture was trained: LeNet, DenseNet121, InceptionV3 and MobileNet. In the case of DenseNet121, InceptionV3 and MobileNet, pre-trained models were used with the weights of ImageNet.

The training was carried out on Google Colab Pro using an A100 GPU, which allowed processing time to be optimized. Without this configuration, training the heavier architectures would have taken 4 to 5 days, requiring continuous connection. The trained models were stored in .h5 and tflite formats for future evaluation.

Subsequently, the trained models were stored in .h5 and.tflite formats to facilitate their evaluation and further use. A prototype was developed in Google Colab to perform the corresponding evaluations, which activated the camera and processed the video in real time. Each captured frame was preprocessed and sent to each trained model, allowing the driver's status to be displayed immediately.

IV. Results

This section presents the results obtained from the evaluation of the LeNet, DenseNet121, InceptionV3 and MobileNet architectures in drowsiness detection. The results are structured in the following analyses:

A. *Training and Validation Curves*

Fig. 8 shows the accuracy and loss curves for yawning detection. MobileNet and InceptionV3 achieved fast convergence and maintained stability during training. DenseNet121 completed its training in fewer epochs with competitive performance. In contrast, LeNet exhibited greater variability in accuracy and difficulties in generalization.

Fig. 9 illustrates the performance in eye state detection. MobileNet and DenseNet121 demonstrated rapid convergence, with a steep reduction in loss from the earliest epochs. InceptionV3, although progressively improving, showed a less pronounced decrease in loss. LeNet exhibited fluctuations in both accuracy and loss, evidencing instability in training.

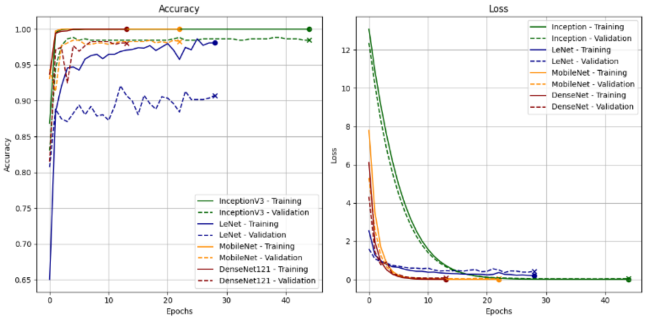


Fig. 8. Accuracy and loss curves during training for yawning state detection using LeNet, DenseNet121, InceptionV3 and MobileNet architectures.

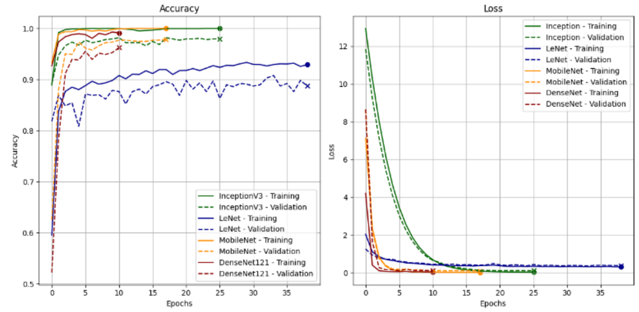


Fig. 9. Accuracy and loss curves during MobileNet, InceptionV3, DenseNet121 and LeNet training for the eye state.

B. *Accuracy in Validation, Training and Testing*

Fig. 10 presented the performance of the models in each phase of the training, validation and testing process. DenseNet121 and MobileNet achieved accuracies of 99.66% and 99.31%, respectively, in the testing phase for yawning detection, with minimal variations with respect to validation. In eye state detection, DenseNet121 achieved an accuracy of 96.53%, while MobileNet recorded 93.28% in the test phase.

InceptionV3 showed an accuracy of 100% in the training phase for eye state detection, with a reduction to 59.87% in the test phase. LeNet obtained an accuracy of 97.24% in yawning detection in the test phase, with lower values than those obtained by DenseNet121 and MobileNet.

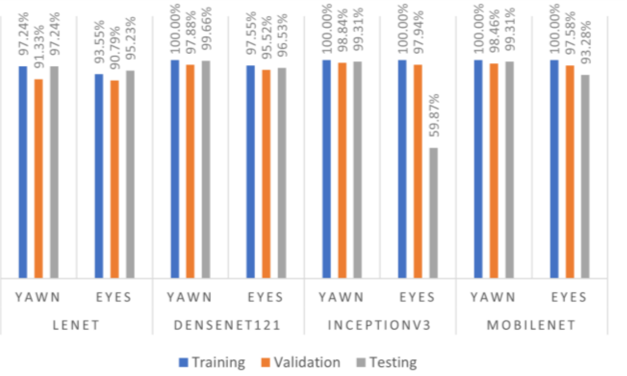


Fig. 10. Comparative performance of CNN architectures in the training, validation and testing phases for yawning and eye states.

C. *Confusion Matrix*

Fig. 11 shows the confusion matrices of LeNet, DenseNet121, InceptionV3 and MobileNet in the classification of 'Yawn' and 'No Yawn' states. It was observed that DenseNet121 performed the best, with high accuracy and minimal error incidence. InceptionV3 and MobileNet showed balanced performance, with similar error rates. In contrast, LeNet presented greater difficulty in detecting 'Yawn', registering more false negatives compared to the other architectures.

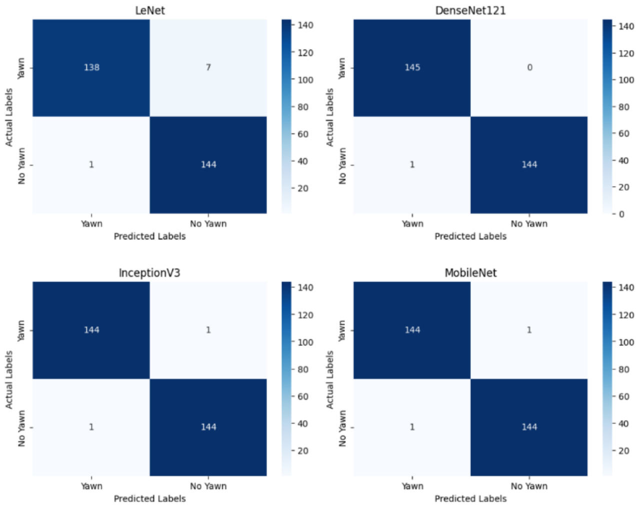


Fig. 11. Confusion matrices of LeNet, DenseNet121, InceptionV3 and MobileNet architectures in yawning state classification (Yawn and No-Yawn).

The confusion matrices of the LeNet, DenseNet121, InceptionV3 and MobileNet architectures for the classification of eye states ('Open', 'Closed' and 'No-Detected') are presented in Fig. 12.

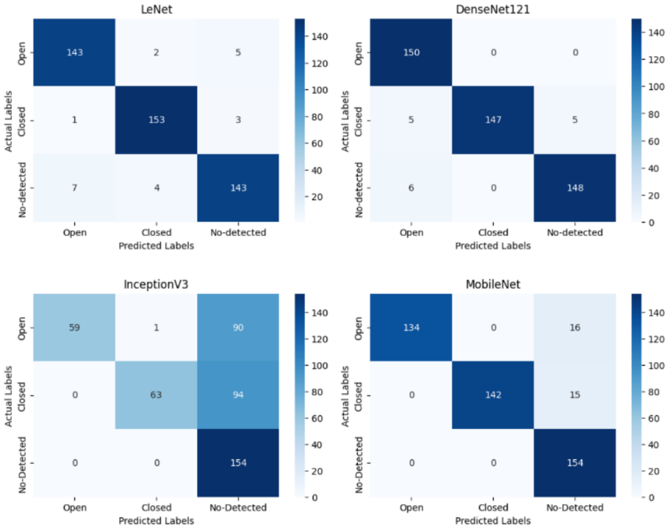


Fig. 12. Confusion matrices for the LeNet, DenseNet121, InceptionV3 and MobileNet architectures in the classification of eye states (Open, Closed and No-Detected).

DenseNet121 performed best, with high true positive values and minimal errors, with no false positives in the 'No-Detected' class. LeNet showed adequate performance, although it recorded false negatives in 'Open' and 'Closed'. MobileNet presented a balance between accuracy and sensitivity, with moderate confusions between 'Open' and 'No-Detected'. In contrast, InceptionV3 had the highest number of false negatives in 'Open' and 'Closed', reflecting a lower ability to correctly classify these categories.

D. *Ranking Metrics*

Table I presents the classification metrics obtained for each architecture in yawning state detection. DenseNet121 achieved an F1-Score of 1.00 in both classes, with precision of 0.99 and recall of 1.00 in "Yawn". InceptionV3 and MobileNet recorded an F1-Score of 0.99, with balanced accuracy and recall in both categories. LeNet obtained an F1-Score of 0.97, with an accuracy of 0.99 in "Yawn" and a recall of 0.95 in the same class. In the yawning state.

TABLE I. YAWNING STATE CLASSIFICATION REPORTS

Architecture	Class	Accuracy	Recall	F1-Score
LeNet	Yawn	0.99	0.95	0.97
	No Yawn	0.95	0.99	0.97
	Macro AVG	0.97	0.97	0.97
DenseNet121	Yawn	0.99	1	1
	No Yawn	1	0.99	1
	Macro AVG	1	1	1
InceptionV3	Yawn	0.99	0.99	0.99
	No Yawn	0.99	0.99	0.99
	Macro AVG	0.99	0.99	0.99
MobileNet	Yawn	0.99	0.99	0.99
	No Yawn	0.99	0.99	0.99
	Macro AVG	0.99	0.99	0.99

TABLE II. EYE CONDITION CLASSIFICATION REPORTS

Architecture	Class	Accuracy	Recall	F1-Score
LeNet	Closed	0.95	0.95	0.95
	Open	0.96	0.97	0.97
	No-detected	0.95	0.93	0.94
	Macro AVG	0.95	0.95	0.95
DenseNet121	Closed	0.93	1	0.96
	Open	1	0.94	0.97
	No-detected	0.97	0.96	0.96
	Macro AVG	0.97	0.97	0.97
InceptionV3	Closed	0.98	0.4	0.57
	Open	1	0.39	0.56
	No-detected	0.46	1	0.63
	Macro AVG	0.81	0.6	0.59
MobileNet	Closed	1	0.9	0.95
	Open	1	0.89	0.94
	No-detected	0.83	1	0.91
	Macro AVG	0.94	0.93	0.93

Table II presents the ranking metrics for each architecture. DenseNet121 obtained the best performance with an average F1-Score of 0.97. MobileNet showed a balanced performance with an F1-Score of 0.93. LeNet showed consistent values across all classes with an F1-Score of 0.95. InceptionV3 recorded the lowest performance, with a noticeable reduction in recall for "Closed" and "Open".

E. Real-Time Testing

The prototype created at Google Colab prepared each frame to meet the input requirements of the architectures, such as image size and format. Subsequently, each model generated its prediction and the level of certainty expressed as a percentage. This level of certainty indicates the model's confidence in its prediction:

- Close to 100%: High prediction confidence.
- Near 50% or less: Low confidence, suggesting doubt or ambiguity in the prediction.

The tests were carried out under real conditions, as shown in Fig. 13, where the models faced different situations such as subject movements, light variations and facial expressions.

1) Yawning state predictions:

a) LeNet: Its level of certainty varies between 56% and 100%, remaining at 100% in stable conditions (with little light variation and minimal user movements). However, in extreme scenarios, such as constant user movements or environments with low illumination, its transparency decreases significantly, leading to incorrect predictions on several occasions.

b) DenseNet121: Its accuracy level ranges between 90% and 100%, remaining at 99% in stable conditions. In extreme scenarios it maintains excellent accuracy, showing great robustness to environmental variations.

c) InceptionV3: Its accuracy fluctuates between 60% and 100%, stabilizing around 91% under normal conditions. It is the model with the greatest variability in its level of certainty, but even so it adapts well to changes in illumination and extreme conditions, offering reliable predictions.

d) MobileNet: With a range of certainty between 95% and 100%, it remains practically 99% in optimal conditions. In extreme scenarios, it shows an outstanding performance, standing out for its accuracy and high reliability.

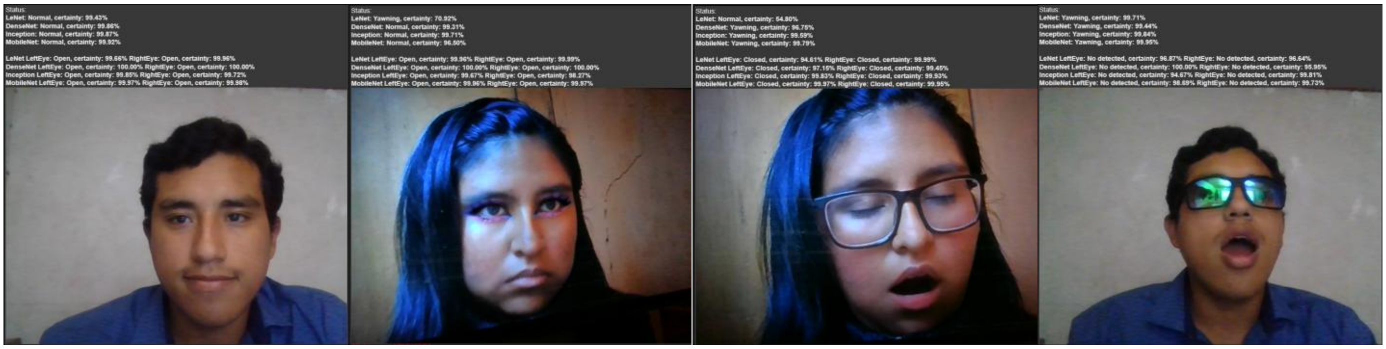
2) Eye condition predictions:

a) LeNet: Although its performance in yawn detection was the lowest, it surprises with decent results in eye status, even in low light environments. Its performance is almost comparable to MobileNet and even becomes better in some cases, showing a very solid behavior in stable conditions. It could be considered as the second best model for this condition.

b) DenseNet121: In challenging environments, it shows greater instability, with notable fluctuations in its predictions. However, in standard conditions, it achieves a satisfactory performance, although it lags behind other more consistent models.

c) InceptionV3: It shares similar characteristics with DenseNet121 in terms of instability in difficult scenarios. Although it achieves very good results in controlled environments, its variability also places it among the least reliable models in this test.

d) MobileNet: It stands out as the most robust and consistent model for the eye condition, maintaining outstanding performance even in difficult or low-light environments. Its ability to adapt to adverse conditions clearly positions it as the best model for this condition.



```
Status:
LeNet: Normal, certainty: 99.43%
DenseNet: Normal, certainty: 99.86%
Inception: Normal, certainty: 99.87%
MobileNet: Normal, certainty: 99.92%

LeNet LeftEye: Open, certainty: 99.66% RightEye: Open, certainty: 99.96%
DenseNet LeftEye: Open, certainty: 100.00% RightEye: Open, certainty: 100.00%
Inception LeftEye: Open, certainty: 99.85% RightEye: Open, certainty: 99.72%
MobileNet LeftEye: Open, certainty: 99.97% RightEye: Open, certainty: 99.98%
```

```
Status:
LeNet: Yawning, certainty: 70.92%
DenseNet: Normal, certainty: 99.31%
Inception: Normal, certainty: 99.71%
MobileNet: Normal, certainty: 96.50%

LeNet LeftEye: Open, certainty: 99.96% RightEye: Open, certainty: 99.99%
DenseNet LeftEye: Open, certainty: 100.00% RightEye: Open, certainty: 100.00%
Inception LeftEye: Open, certainty: 99.67% RightEye: Open, certainty: 98.27%
MobileNet LeftEye: Open, certainty: 99.96% RightEye: Open, certainty: 99.97%
```

```
Status:
LeNet: Normal, certainty: 54.80%
DenseNet: Yawning, certainty: 96.75%
Inception: Yawning, certainty: 99.59%
MobileNet: Yawning, certainty: 99.79%

LeNet LeftEye: Closed, certainty: 94.61% RightEye: Closed, certainty: 99.99%
DenseNet LeftEye: Closed, certainty: 97.15% RightEye: Closed, certainty: 99.45%
Inception LeftEye: Closed, certainty: 99.83% RightEye: Closed, certainty: 99.93%
MobileNet LeftEye: Closed, certainty: 99.97% RightEye: Closed, certainty: 99.95%
```

```
Status:
LeNet: Yawning, certainty: 99.71%
DenseNet: Yawning, certainty: 99.44%
Inception: Yawning, certainty: 99.84%
MobileNet: Yawning, certainty: 99.95%

LeNet LeftEye: No detected, certainty: 96.87% RightEye: No detected, certainty: 96.64%
DenseNet LeftEye: No detected, certainty: 100.00% RightEye: No detected, certainty: 95.95%
Inception LeftEye: No detected, certainty: 94.67% RightEye: No detected, certainty: 99.81%
MobileNet LeftEye: No detected, certainty: 98.69% RightEye: No detected, certainty: 99.73%
```

Fig. 13. Images of the prototype executed in real time.

V. DISCUSSIONS

The results obtained in this study showed that the DenseNet121 and MobileNet architectures significantly outperformed the models used in previous research, achieving accuracies of over 99% in yawn detection and 93% in eye detection.

Unlike the model of Zhao et al [15], which used EM-CNN and MTC-CNN for face detection, this evaluation identified MobileNet and DenseNet as the best performing architectures. MobileNet achieved 99.31% test accuracy for yawn detection and 93.28% for eye classification, while DenseNet obtained 99.66% and 96.53%, respectively. These results exceed the 93.62% reported in their study, evidencing the stability of MobileNet and DenseNet in diverse scenarios. Furthermore, unlike PERCLOS and POM-based approaches, the landmark-based classification enabled a more accurate segmentation of facial regions, optimizing the detection of fatigue-relevant facial states.

The study by Chirra, Uyyala, and Kolli [17] used a stacked CNN together with the Viola-Jones algorithm, obtaining an accuracy of 96.42% in eye state detection. In the present work, DenseNet121 achieved an accuracy of 96.53% without the need for additional face detection algorithms, demonstrating its ability to extract relevant features efficiently. The improved performance can be attributed to the preprocessing techniques applied, such as normalization, grayscale conversion and face segmentation with MediaPipe, in addition to the implementation of L2 regularization and dropout, which contributed to reduce overfitting and improve the generalization capability of the model.

Also, unlike the study by Ma, Chau, and Yap [14], which employed a Two-stream CNN with Kinect sensors to improve fatigue detection in nighttime conditions, this study evaluated real-time CNN architectures without the need for additional sensors. While the video stream-based approach achieved 91.57% accuracy, MobileNet and DenseNet121 achieved up to 99% in yawning and eye state detection, while maintaining stability in the face of illumination and motion variations. These results suggest that optimized models can be an efficient alternative for drowsiness detection without requiring specialized hardware.

On the other hand, in the study by Li, Gao and Suganthan [16] they achieved an accuracy of 83.48% when combining CNN with EEG signals, a methodology that, although useful, is more complex and less accurate than the visual feature analysis performed in this study. The results obtained with LeNet, DenseNet121 and MobileNet in both states evaluated reflect that facial image-based techniques are more accurate and practical for vehicular implementations.

Finally, the study by Florez [18] used CNN in a real-time system with InceptionV3, VGG16, ResNet50V2, obtaining an accuracy of 91.48% in simulations and 86.28% in real driving. In this study, InceptionV3 showed 99.31% in test, but its real-time certainty ranged from 60% to 100%, with a drop to 59.87% in eye-state detection. MobileNet and DenseNet121 were more stable, with 99% real-time certainty. This confirms that, although the models achieve high accuracy in controlled tests, their performance in real environments can be affected, with MobileNet standing out as the most robust.

VI. Conclusions and Recommendations

In this study, different convolutional neural network architectures were evaluated and compared for drowsiness detection in drivers, focusing on two key aspects: eye state and yawning state. For yawning state, the architectures that topped the list in accuracy upon training were InceptionV3 with a validation accuracy of 98.84%, followed by MobileNet with 98.46%, DenseNet121 with 97.88% and finally LeNet with 91.33%. For eye status, InceptionV3 stood out with a validation accuracy of 97.94%, followed by MobileNet with 97.58%, then InceptionV3 with 95.52%, and finally LeNet with 90.79%.

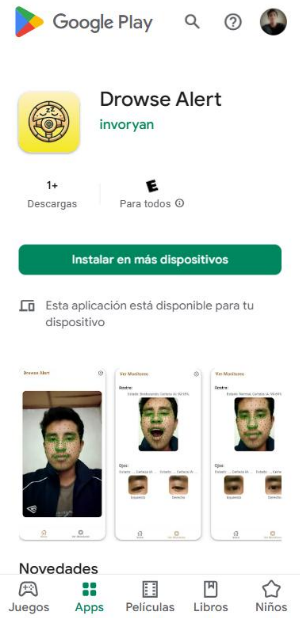
The results of the confusion matrices and the classification report clearly reflect this superior performance. LeNet, DenseNet121, InceptionV3 and MobileNet achieved outstanding classifications in the yawning and eyes state having very few errors with the exception of InceptionV3 which showed less consistent performance in the eyes state, especially in the 'Open' class, where a significant number of errors were observed.

Finally, in real-time testing with the prototype, MobileNet proved to be the most robust and reliable architecture for both yawning and eye detection. Its outstanding accuracy remained consistent even under challenging conditions. LeNet, although it came in last place in yawning state detection, surprised by showing solid performance in eye state detection, obtaining comparable or even better results to MobileNet in some scenarios. DenseNet121 showed solid and consistent performance in yawning state detection, positioning itself as a reliable alternative to MobileNet. However, its performance in eye state detection was more unstable in challenging environments. For its part, InceptionV3, while achieving acceptable results in stable conditions, presented the greatest variability between architectures in both tasks. Its performance was less consistent in challenging environments, especially in eye-state detection, where it showed a higher number of errors compared to the other models.

In conclusion, when comparing the architectures, MobileNet stood out as the best choice for its consistency and accuracy, even in challenging conditions such as low illumination or constant user movements. Although DenseNet121 and InceptionV3 also offered good performance in stable environments, MobileNet stood out for its adaptability and comprehensive performance. On the other hand, LeNet, although the simplest architecture in this study, showed surprisingly good performance in eye detection, despite its lack of optimization compared to more modern architectures. It is important to note that the architectures were not modified, as the goal was to evaluate them as they are, respecting their internal structure. Although it is possible to improve the architectures by adding layers, in this study we chose to add the same layers at the beginning and at the end of all the architectures, maintaining their original shape and respecting the design with which they were built.

For future research, the integration of other types of data, such as heart rate or electromyographic activity, could be explored to improve accuracy and robustness in the detection of physiological states. It would be interesting to develop a prototype that combines these new instruments with the data obtained in this study, offering a more complete analysis. This approach could open new possibilities for more personalized and effective health monitoring systems.

Finally, based on the evaluations, an application called 'Drowse Alert' was developed using the MobileNet model, which obtained the best performance. The app is currently in closed testing on Google Play, available only to selected users. The link to access the application is as follows: https://play.google.com/store/apps/details?id=com.invoryan.drowse, as shown in Fig. 14.



Esta aplicación está disponible para tu dispositivo

Fig. 14. Screenshot of the “Drowse Alert” application in Google Play.

References

[1] Y. Albadawi, M. Takruri, and M. Awad, “A review of recent developments in driver drowsiness detection systems,” Sensors, vol. 22, no. 5, p. 2069, 2022.

[2] K. Peña Prado, “Somnolencia en conductores de transporte público regular de pasajeros de Lima Metropolitana – Perú. 2016,” Master’s thesis, Universidad Peruana Cayetano Heredia, Lima, Peru, 2017.

[3] World Health Organization, “Road traffic injuries,” Who.int, Dec. 13, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. [Accessed: Sep. 23, 2024].

[4] B. C. Tefft, “Drowsy driving in fatal crashes, United States, 2017–2021 (Research Brief),” AAA Foundation for Traffic Safety, Washington, D.C., 2024.

[5] Dirección General de Tráfico, “Conducir con sueño o cansancio,” Www.dgt.es, 2024. [Online]. Available: https://www.dgt.es/muevete-con-seguridad/evita-conductas-de-riesgo/Conducir-con-sueno-o-cansancio. [Accessed: Sep. 24, 2024].

[6] Instituto Nacional de Estadística e Informática, “Análisis de los accidentes de tránsito ocurridos en el año 2016,” Plataforma del Estado Peruano, 2017. [Online]. Available: https://www.inei.gob.pe/Est/Lib1528/cap03. [Accessed: Sep. 24, 2024].

[7] P. Purwono et al., “Understanding of convolutional neural network (CNN): A review,” Int. J. Robotics Control Syst., vol. 2, no. 4, pp. 739–748, 2022.

[8] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, “Fundamental concepts of convolutional neural network,” in Recent Trends and Advances in Artificial Intelligence and Internet of Things, V. Balas, R. Kumar, and R. Srivastava, Eds., Intelligent Systems Reference Library. Springer, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-32644-9_36.

[9] D. Perumandla, “Drowsiness dataset,” Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/dheerajperumandla/drowsiness-dataset. [Accessed: Sep. 20, 2024].

[10] M. V. Vijaya Saradhi, P. Venkateswara Rao, V. Gokula Krishnan, K. Sathyamoorthy, and V. Vijayaraja, “Prediction of Alzheimer’s disease using LeNet-CNN model with optimal adaptive bilateral filtering,” Int. J. Commun. Netw. Inf. Secur., vol. 15, no. 1, pp. 52–58, 2023.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit, 2017, pp. 4700–4708.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2818–2826.

[13] A. G. Howard, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017. [Online]. Available: https://arxiv.org/abs/1704.04861.

[14] X. Ma, L.-P. Chau, and K.-H. Yap, “Depth video-based two-stream convolutional neural networks for driver fatigue detection,” in 2017 International Conference on Orange Technologies (ICOT), Singapore, pp. 155–158, 2017. [Online]. Available: https://doi.org/10.1109/ICOT.2017.8336111.

[15] Z. Zhao, N. Zhou, L. Zhang, H. Yan, Y. Xu, and Z. Zhang, “Driver fatigue detection based on convolutional neural networks using EM-CNN,” Computational Intelligence and Neuroscience, vol. 2020, Art. no. 7251280, 11 pages, 2020. [Online]. Available: https://doi.org/10.1155/2020/7251280.

[16] R. Li, R. Gao, and P. N. Suganthan, “A decomposition-based hybrid ensemble CNN framework for driver fatigue recognition,” Information Sciences, vol. 624, pp. 833–848, 2023. [Online]. Available: https://doi.org/10.1016/j.ins.2022.12.088.

[17] V. R. R. Chirra, S. R. Uyyala, and V. K. K. Kolli, “Deep CNN: A machine learning approach for driver drowsiness detection based on eye state,” Revue d'Intelligence Artificielle, vol. 33, no. 6, pp. 461–466, Dec. 2019. [Online]. Available: https://doi.org/10.18280/ria.330609.

[18] R. D. Florez Zela, “Diseño e implementación de un sistema detector de somnolencia en tiempo real mediante visión computacional usando redes neuronales convolucionales aplicado a conductores,” Undergraduate Thesis, Universidad Nacional de San Antonio Abad del Cusco, 2024. [Online]. Available: http://hdl.handle.net/20.500.12918/8298.