applied sciences

MDPI

Article

A CNN-Based Approach for Driver Drowsiness Detection by Real-Time Eye State Identification

Ruben Florez [1,*], Facundo Palomino-Quispe [1], Roger Jesus Coaquira-Castillo [1], Julio Cesar Herrera-Levano [1], Thuanne Paixão [2] and Ana Beatriz Alvarez [2]

[1] LIECAR Laboratory, University of San Antonio Abad del Cusco (UNSAAC), Cuzco 08000, Peru; facundo.palomino@unsaac.edu.pe (F.P.-Q.); roger.coaquira@unsaac.edu.pe (R.J.C.-C.); julio.herrera@unsaac.edu.pe (J.C.H.-L.)

[2] PAVIC Laboratory, University of Acre (UFAC), Rio Branco 69915-900, Brazil; thuannepaixao@gmail.com (T.P.); ana.alvarez@ufac.br (A.B.A.)

* Correspondence: rubendfz2206@gmail.com

Abstract: Drowsiness detection is an important task in road safety and other areas that require sustained attention. In this article, an approach to detect drowsiness in drivers is presented, focusing on the eye region, since eye fatigue is one of the first symptoms of drowsiness. The method used for the extraction of the eye region is Mediapipe, chosen for its high accuracy and robustness. Three neural networks were analyzed based on InceptionV3, VGG16 and ResNet50V2, which implement deep learning. The database used is NITYMED, which contains videos of drivers with different levels of drowsiness. The three networks were evaluated in terms of accuracy, precision and recall in detecting drowsiness in the eye region. The results of the study show that all three convolutional neural networks have high accuracy in detecting drowsiness in the eye region. In particular, the Resnet50V2 network achieved the highest accuracy, with a rate of 99.71% on average. For better visualization of the data, the Grad-CAM technique is used, with which we obtain a better understanding of the performance of the algorithms in the classification process.

Keywords: driver monitoring system; drowsiness detection; convolutional neural network; Grad-CAM visualization

Citation: Florez, R.; Palomino-Quispe, F.; Coaquira-Castillo, R.J.; Herrera-Levano, J.C.; Paixão, T.; Alvarez, A.B. A CNN-Based Approach for Driver Drowsiness Detection by Real-Time Eye State Identification. *Appl. Sci.* **2023**, *13*, 7849. https://doi.org/10.3390/app13137849

Academic Editor: Hui Yuan

Received: 1 June 2023
Revised: 29 June 2023
Accepted: 30 June 2023
Published: 4 July 2023

1. Introduction

According to the Pan American Health Organization (PAHO), 1.35 million people die from road traffic crashes, and millions of people are injured worldwide. In middle- and low-income countries, 90 percent of deaths are caused by traffic accidents, accounting for approximately 3 percent of PBI [1]. In the case of Peru, in the first seven months of 2022, more than 47,600 traffic accidents were reported, causing the death of 1853 people, which is a monthly average of 265 victims in traffic accidents. In 2021, there were more than 74,620 traffic accidents causing the death of 30,032 people [2], which is alarming. Of all the possible causes, the human factor is one of the main factors, representing 93.9% (27,396) of road accidents caused by drivers in 2022 [3]. The majority of vehicle accidents are caused by driver drowsiness while driving.

In order to reduce accidents due to drowsiness, there are current studies that provide different methods to detect driver drowsiness in time to avoid an accident. According to the study by Albadawi et al. [4], four measures are determined for the detection of drowsiness, one of them is based on the vehicle, taking the angle of the steering wheel and the deviation from the highway lane; another is based on bio-signals such as electrocardiography (ECG), electroencephalography (EEG), electrooculogram (EOG), etc. Biosignal measures are very accurate but invasive for drivers. The other measure is based on image analysis, specifically focusing on the eyes, mouth and head position; this measure is widely used because it is non-invasive and does not cause discomfort to the driver. Furthermore, most drowsiness signals

are presented in facial features, so it is easier to determine when a driver shows symptoms of drowsiness. The last method is a combination of the three measures mentioned above.

In the research by Vikranth et al. [5], a compilation of 24 studies with various methods for drowsiness detection with deep learning is presented. The author presents the use of CNNs based on ResNet50, MobileNetV2, VGG16, Inception and GoogleLeNet as the best performers for drowsiness classification in images.

Currently, there are several databases for drowsiness detection, among them we have NTHU-DDD [6], YawDDD [7], MRL Eye [8], UTA-RLDD [9], NITYMED [10], etc. The latter database, NITYMED, presents videos of people driving in a real uncontrolled environment with different light conditions, manifesting drowsiness symptoms through the eyes and mouth. The other databases are mostly in a controlled environment, where people simulate drowsiness symptoms. Therefore, in comparison with the aforementioned databases, NITYMED presents a more realistic approach to the presence of drowsiness in drivers, which makes it suitable for use in this research.

This paper presents an approach to determining driver drowsiness by digital image analysis, exploring the state of the eyes (open or closed) using methods that implement deep learning such as convolutional neural networks (CNNs). For the selection of the region of interest, an approach for the correction of points near the eyes is proposed. Based on Vikranth et al. [5], three CNN architectures will be used as a basis: InceptionV3 [11], VGG16 [12] and ResNet50V2 [13], which use transfer learning [14] and MediaPipe [15] for facial point detection and region of interest (ROI) extraction. The authors also adapted the fully connected network for the binary classification of drowsiness. For the identification of drowsiness in drivers in a real environment, the probability of the ROI belonging to the drowsiness class is evaluated and subsequently used as the proposal presented by [16], which consists of counting the time of a normal blinking eye from 100 to 300 ms; when the eyes are closed for more than 300 ms it is considered a drowsy state.

The paper is organized as follows: Section 2 describes the literature related to drowsiness detection; Section 3 describes the materials and methods used; Section 4 shows the results obtained and their analysis for each CNN architecture, obtaining a model for each. Finally, Section 5 presents the conclusions and future research.

2. Related Work

In the research of Park et al. [17], an architecture called deep drowsiness detection (DDD) is proposed, which processes RGB videos that focus on the driver's entire face. The DDD architecture makes use of three architectures: AlexNet, VGG-FaceNet and FlowImageNet, where the output of the three networks are unified in order to classify the drowsiness in frames of the input videos. To test the proposed model, the authors use the NTHU drowsy driver detection (NTHU-DDD) database, achieving an average accuracy of 73.06% during their experimental results.

Chirra et al. [18] proposed an architecture that specifically uses the eye region. For the extraction of the eye region, the Haar Cascade technique proposed by Viola Jones was used. To detect the face and eyes, the ROI of the eyes becomes the input of their CNN where they used a database collected for the training of their network, obtaining an accuracy of 98% in training, 97% in validation and 96.42% in the final test.

In the approach of Zhao et al. [19], the authors used facial characteristic points for drowsiness detection and classification. They made use of an MTCNN (multi-task cascaded convolutional network) for face detection and characteristic point location, extracting ROIs from the eyes and mouth that pass to their network called EM-CNN, where they make a classification of four classes, two for the eyes state and two for the mouth state. Their tests were performed on a database provided by the company Biteda, where they obtained 93.623% accuracy compared to other types of architectures.

In the proposal by Phan et al. [20], two methods were proposed for drowsiness detection, the first one uses characteristic points of the face focusing on the eyes and mouth using the Dlib library, applying thresholds to determine if it is yawning or blinking. The

second method uses MobileNet-V2 and ResNet-50V2 networks using transfer learning. For the training of CNNs, the authors collected images from various sources to generate their dataset, obtaining an average result of 97% accuracy.

In the system presented by Rajkar et al. [21], Haar Cascade was used to extract the eyes. The ROI extraction was performed for each eye separately after detecting the face, then the proposed architecture was used for training, using two databases: YawDDD and Closed Eyes In The Wild. The authors achieved an accuracy of 96.82%.

In the research presented by Hashemi et al. [22], a drowsiness detection system was proposed by training three CNNs, one designed by the authors and the others by transfer learning using VGG16 and VGG19. The face detection was performed by Haar Cascade, and then Dlib was used to detect the eye points and thus delimit the region of interest for the training of the three networks using the ZJU Eyeblink database. Their results showed an accuracy of 98.15% with the proposed network.

In the proposal presented by Alameen and Alhothali [23], a 3D-CNN was implemented for the extraction of spatio-temporal features; these learned features were used as inputs for a long-term short-term memory (LSTM). The authors propose two 3D-CNN+LSTM models (A and B) tested on two datasets (3MDAD and YawDDD), obtaining a 96% accuracy on YawDDD.

Gomaa et al. [24] proposed several architectures combined with CNN and LSTM, the CNNs were based on ResNet, VGG-16, GoogleNet and MobileNet. The authors also propose their own architecture called "CNN-LSTM", training and testing the architectures on the "NTHU" dataset. Their network "CNN-LSTM" obtained an accuracy of 98.3% in training and 97.31% in testing, outperforming the other four networks.

Singh et al. [25] presented a system to detect drowsiness based on eye aspect ratio (EAR) combined with PERCLOS, which calculates the percentage of eye closure for a period of time. They used Dlib for eye point extraction, which is necessary for EAR. Their system had an 80% accuracy for their presented method.

Finally, in the research of Tibrewal et al. [26], they proposed a CNN architecture. For learning and testing, the MRL eye database was used, which provides images of a single eye. For eye ROI extraction, the Dlib library was used. The authors obtained 94% average accuracy in drowsiness detection by focusing on the eye state.

3. Proposed Method

The proposed approach uses the methodology represented by the flowchart in Figure 1, which consists of six stages: acquisition of the data (video), pre-processing of the images captured from the videos, creation of the dataset, training of the CNN architectures, testing of the trained models and subsequent prediction of driver drowsiness. Firstly, the acquisition of videos showing driver drowsiness is performed. After obtaining the data, pre-processing is conducted to extract the frames where the 468 facial points are detected by MediaPipe. For a better selection of the ROI, a methodology is proposed that uses 4 points around the eyes and—with the help of an intermediate point between the eyes—calculates the distances from the extreme point of the right eye to the extreme point of the left eye and the upper and lower extreme points of the right and left eyes, comparing them and thus selecting the most significant distance to create the ROI. This method guarantees the ROI of the eyes without losing information when the driver makes head movements looking up, down, right and left. This proposed method is described in detail in Section 3.2.3 ROI selection. After having the ROI selected, the frames are extracted to create the dataset. All the images of the dataset go through a processing that resizes them to an image of 112×112 pixels and then normalizes them by dividing each pixel by 255. At this stage, data augmentation is also applied to the training set in order to avoid overfitting. Then, the processed images are used for the training of the 3 CNN architectures generating their respective accuracy and loss graphs. To run and evaluate the performance of the networks, a test is performed with the set of test images resulting in the selection of the

best performing model. Finally, with the selected model, the driver drowsiness prediction test is performed.

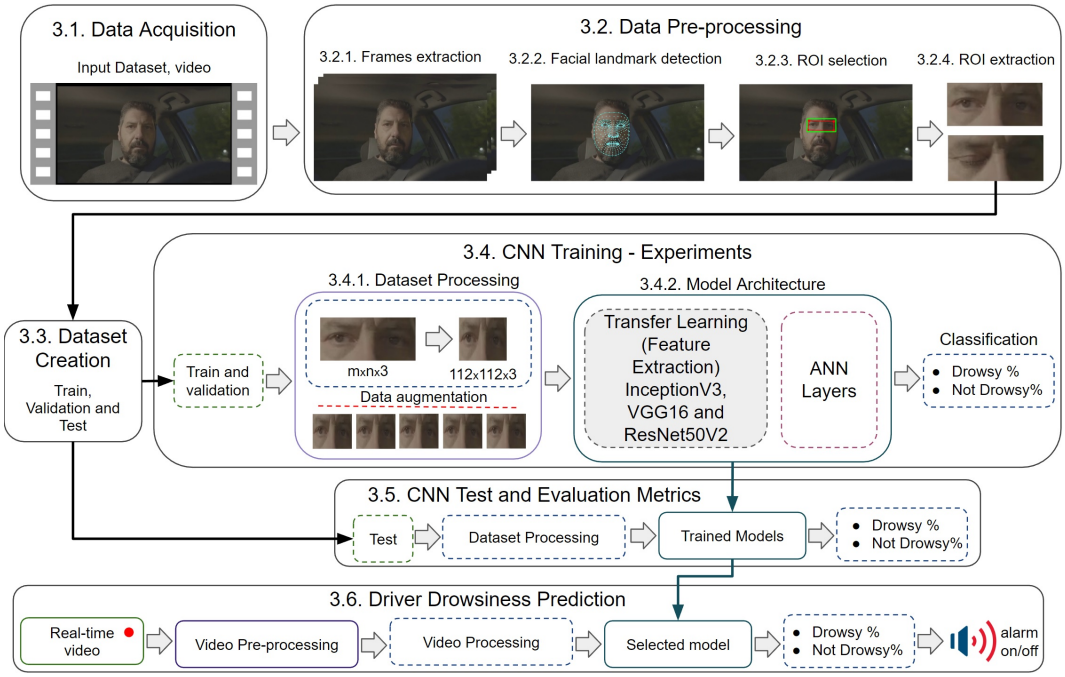


Figure 1. Methodology for the detection of driver drowsiness.

3.1. Data Acquisition

Currently there are some databases that can be used for sleepiness detection such as NTHU-DDD, YawDDD, MRL Eye, UTA-RLDD, etc. This proposal uses the night-time yawning–microsleep–eyeblink–driver distraction (NITYMED) database, a database recently made available containing videos of males and females in a real night driving environment manifesting drowsiness symptoms through their eyes and mouth. NITYMED consists of 130 videos in mp4 format at 25 fps in 1080p (FullHD) and 720p (HD) resolutions. Compared to the other databases, NITYMED is more realistic for the purpose of this work, which justifies its use.

3.2. Data Pre-Processing

This subsection includes 4 steps, where the proposed ROI correction is shown in Section 3.2.3. These steps were also used to create the training, validation and test dataset.

3.2.1. Frames Extraction

From the video database, consecutive frames were extracted using a counter $f(n) = f_1, f_2, f_3, ..., f_n$, according to the 25 fps and duration of each video. Considering the duration of the videos (30 s to 120 s), the fps was constant $k = 25$, and the frames of each video were obtained using Equation (1).

$$f(n) = k * A_v \tag{1}$$

where the number of frames depends on the duration of each video in the dataset. An example is given with the time of 30 s and 120 s using Equation (1).

$$f(n_1) = 25 * 30 = 750\ frames, f(n_2) = 25 * 120 = 3000\ frames$$

3.2.2. Facial Landmark Detection

In this step, use is made of MediaPipe Face Mesh [27], which estimates 468 facial reference points of the 3D face in real time, thus detecting the face in each image. MediaPipe

Face Mesh works for different head positions, where the face can be detected at different head rotation angles by employing machine learning (ML) to infer the 3D facial surface.

3.2.3. ROI Selection

From the 468 points estimated in the previous step, only 4 points are needed to select the area of the region of interest (ROI). The points chosen within MediaPipe Face Mesh were: 63, 117, 293 and 346, where joining them to create the ROI forms an irregular rectangle as shown in Figure 2a. From most of the existing ROI extraction algorithms [28–31], Figure 2b shows the proposed method for ROI correction, where a point correction was performed. It was proposed to consider as initial point for the x and y components of point 63 $(P_{xi,yi})$, and as final point, the x and y components of point 346 $(P_{xf,yf})$. Then, we find the corresponding distances to each point, d1, d2, d3 and d4, with a point in the middle of both eyes, which is point 9. Then, we made a comparison of extreme points at different head movements stored in the variables $start\_px, end\_px, start\_py$ and end_py. The pseudocode used for ROI correction is shown in Algorithm 1.

(**a**) (**b**)

Figure 2. ROI correction. (**a**) ROI with the 4 points. (**b**) ROI correction method.

Algorithm 1 ROI Correction

Input: Points: $[63, 117, 293, 346, 9]$ ▷ eye region points
Output: ROI
$x_i, y_i = P_{63}[x], P_{63}[y]$ ▷ "x" and "y" components of the upper right extreme points
$x_f, y_f = P_{346}[x], P_{346}[y]$
$d_1 = distance(P_{63}, P_9)$ ▷ distances of the extreme points and superiors
$d_2 = distance(P_9, P_{293})$
$d_3 = distance(P_{293}, P_{346})$
$d_4 = distance(P_{63}, P_{117})$
if $x_i > x_f$ **then** ▷ distance comparison in "x" components
  $start\_px, end\_px = x_f, (x_i + d_1)$
else
  $start\_px, end\_px = x_i, (x_f + d_2)$
end if
if $y_i > y_f$ **then** ▷ distance comparison in "y" components
  $start\_py, end\_py = y_f, (y_i + d_4)$
else
  $start\_py, end\_py = y_i, (y_f + d_3)$
end if
if $(end\_px - start\_px) > 10 \& (end\_py - start\_py) < 400$ **then** ▷ corrected ROI creation
  $start\_px, start\_py = start\_px - 10, start\_py - 10$
  $end\_px, end\_py = end\_px + 10, end\_py + 10$
  $ROI = [start\_py : end\_py, start\_px : end\_px]$ ▷ corrected ROI
end if

An example of application of the ROI correction method can be seen in Figure 3, where the results are shown at 4 different positions of the driver's head. The red outline represents the irregular or deformed ROI, while the green outline is the corrected ROI.

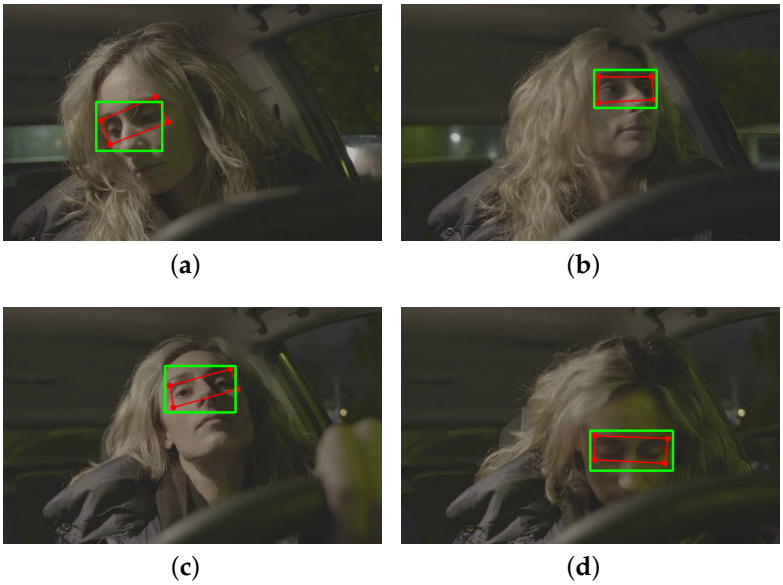


Figure 3. ROI correction in 4 different positions of the driver's head. (**a**) Head right. (**b**) Head left. (**c**) Head up. (**d**) Head down.

3.2.4. ROI Extraction

After making the ROI correction, the eye area that serves as the CNN input was extracted. This step is useful for real-time analysis since all the previous steps will be applied to the live video input. Depending on the drowsiness state of the driver, the ROI may characterize sleep or wakefulness, represented by eyes open and eyes closed, respectively (Figure 4).

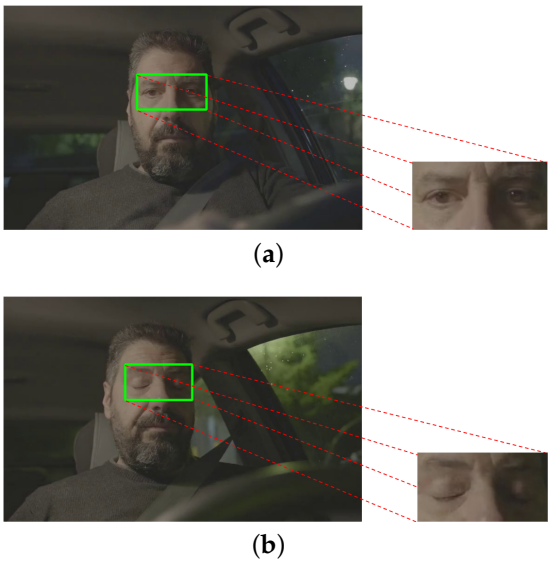


Figure 4. ROI extraction. (**a**) Eyes open. (**b**) Eyes closed.

3.3. Dataset Creation

The database was created with the ROI images obtained in the previous steps, focusing only on the eye region. From NITYMED, 6 videos were chosen for the creation of the

training, validation and test data, obtaining a total of 6800 images. The videos were chosen according to the size of the eyes of the people (4 males and 2 females), where these people have different characteristics from other people. The number of images extracted was due to the fact that some frames of the eyes were repeated, so only relevant frames were chosen.

Since this is a binary classification, 2 classes *Not drowsy* and *Drowsy* are labeled. Of the total images, 4760 (75%) images were split for the train data, 1020 (15%) images for the validation data and 1020 (15%) images for the test data. This data distribution was selected to avoid overfitting due to the limited amount of data. The final distribution of the created data set is shown in Table 1.

Table 1. Dataset distribution.

Data Set	Classes	
	Drowsy	**Not Drowsy**
Training set	2380	2380
Validation set	510	510
Test set	510	510

3.4. CNN Training Experiments

3.4.1. Dataset Processing

Before training the CNNs, image processing was performed. The extracted ROI had different pixel sizes with 3 layers of depth (mxnx3); therefore, resizing to a size accepted by the CNNs was necessary. All images were adjusted to a size of 112 × 112 × 3 pixels, then normalized by changing each pixel value from 0–255 to 0–1. Then, to avoid overfitting, data augmentation was applied with the following parameters: rotation range 20%, horizontal flip *True* and fill mode *Nearest*, resulting in the creation of 5 images from each image of the training set.

3.4.2. Model Architecture

Three CNN architectures were trained based on: InceptionV3, VGG16 and ResNet50V2. By means of transfer learning, the feature extraction weights of each CNN were obtained. Next, the binary classification architecture (not drowsy and drowsy) was designed by flattening the transfer learning output, followed by a 30% dropout with a dense hidden layer of 1000 neurons with ReLU activation and a 30% dropout with a dense layer of 2 outputs with SoftMax activation. The classification process was the same for all 3 CNNs. The proposed architecture is shown in Figure 5.

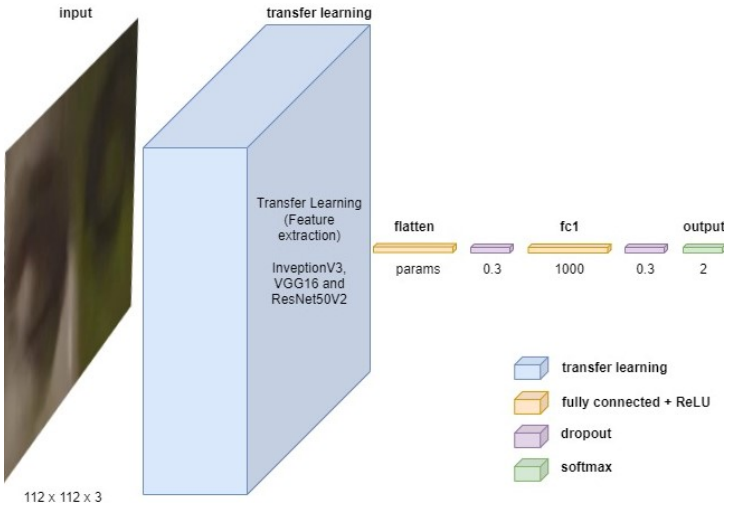


Figure 5. Architecture proposed with transfer learning.

The parameters used in the training are shown in Table 2.

Table 2. Training parameters.

Hyper-Parameters	Value
Optimizer	ADAM
β_1	0.001
β_2	0.9
Learning rate	0.999
Epochs	30
Batch size	32
Number of experiments	10 for each CNN

An example (arbitrarily chosen) of the resulting training plots for each architecture is shown in Figure 6.

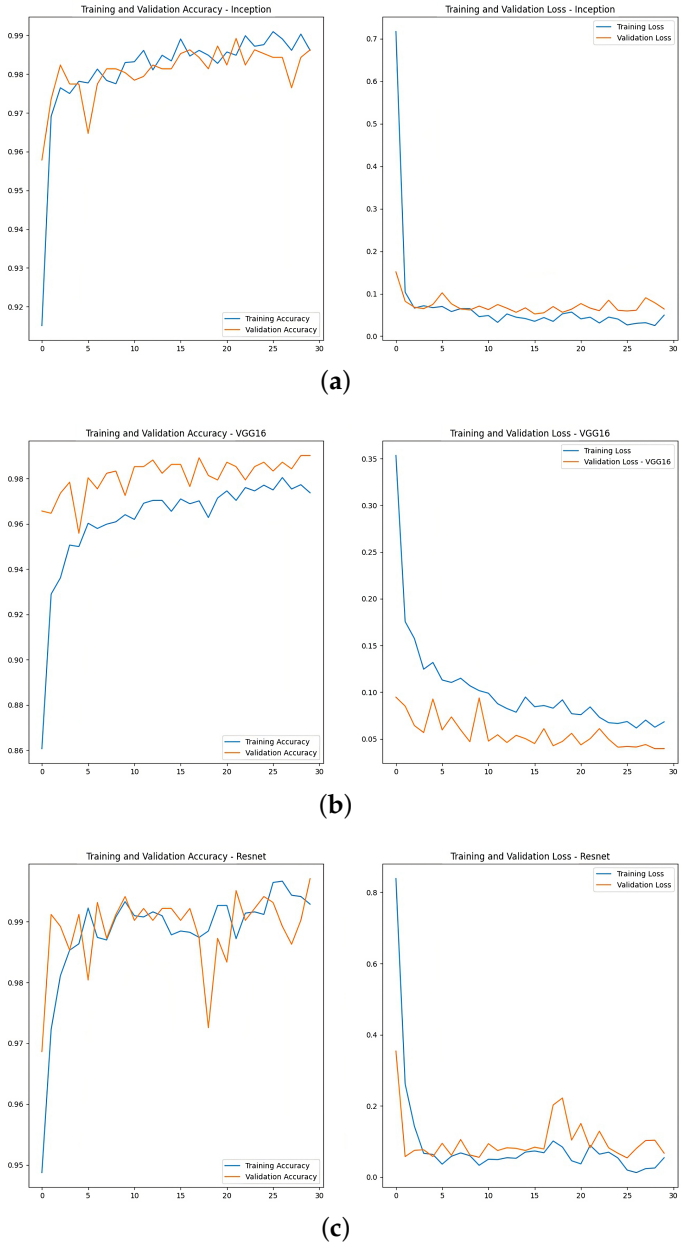


Figure 6. Graphs resulting from the CNNs. (**a**) InceptionV3 accuracy and loss. (**b**) VGG16 accuracy and loss. (**c**) ResNet50V2 accuracy and loss.

3.5. CNN Test and Evaluation Metrics

3.5.1. CNN Test

After training the CNNs, it was necessary to test them with the Test data. Where the processing of the images of the set (without data augmentation) was also performed. A batch size of 1 was used to analyze each image, testing 10 times with each trained network. An example of the confusion matrices for each CNN Test is shown in Figure 7.

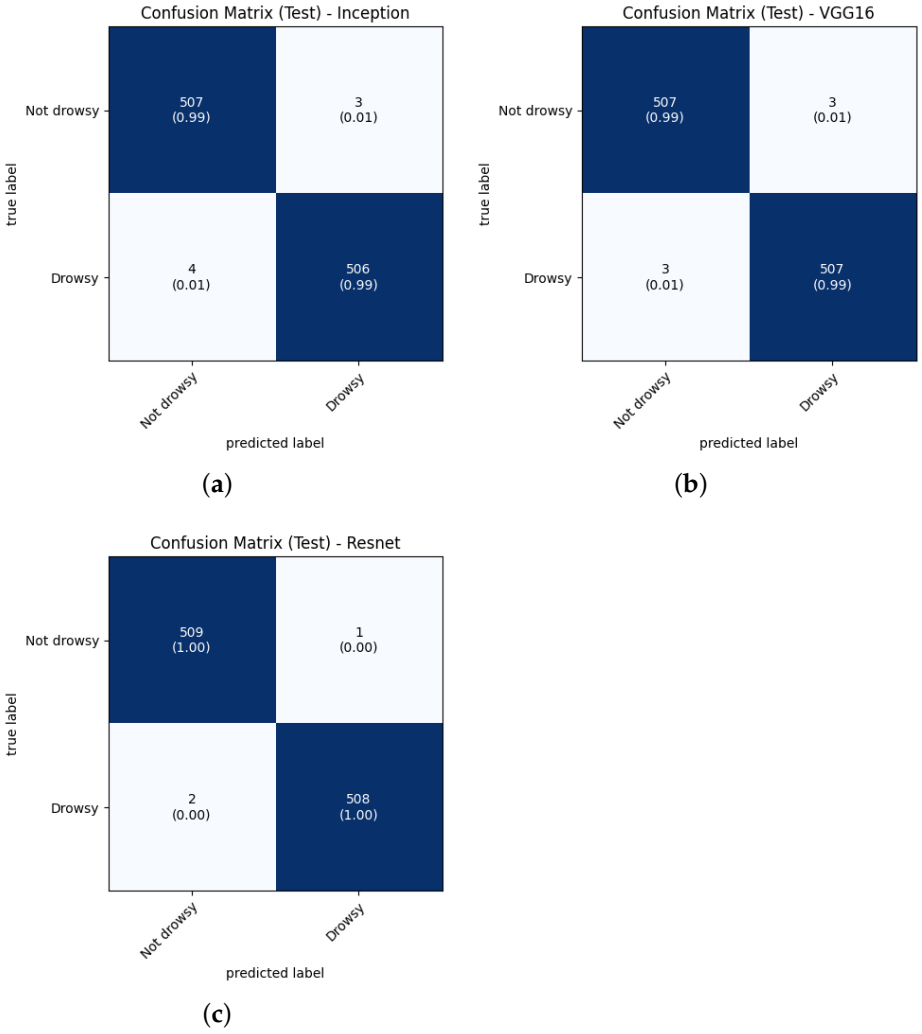


Figure 7. Confusion matrix in CNN Test. (**a**) Confusion matrix in InceptionV3 testing. (**b**) Confusion matrix in VGG16 testing. (**c**) Confusion matrix in ResNet50V2 testing.

The confusion matrix is a tool that allows us to see the performance of the models. In this particular case, it is required to decrease the false negatives for the Drowsy class where the true value is "Drowsy" and the model predicted is "Not drowsy". This error can cause problems that would cause accidents since the system would not alert the presence of drowsiness to the driver. From Figure 7a, it is observed that 4 images were recognized as "Not drowsy" out of the 510 images that are of the class "Drowsy", Figure 7b shows that 3 images were recognized as "Not drowsy". Figure 7c shows that 2 images were recognized as "Not drowsy", corresponding to CNNs based on InceptionV3, VGG16 and ResNet50V2.

3.5.2. Evaluation Metrics

By training the Train and Validation dataset, and testing the trained CNN models on the Test dataset, the confusion matrices were obtained. Based on [32], from these confusion

matrices the evaluation metrics used are: Precision (Equation (2)), Recall (Equation (3)), F1-score (Equation (4)) and Accuracy (Equation (5)), that define the system behavior.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - score = \frac{2 * precision * recall}{precision + recall} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where, TP as true positive, FP as false positive, TN as true negative and FN as false negative.

3.6. Driver Drowsiness Detection

With the models trained and tested, it is finally appropriate to try out the best approach (accuracy) in a real environment. To determine driver drowsiness, first the model estimates if the probability of the ROI extraction belonging to the Drowsy class is higher than 95%. If so, it is necessary to count the time that the eyes remain closed. If it is more than 300 ms, it is considered drowsiness and an alarm will be displayed to alert the driver. The flowchart of the driver drowsiness detection process in a real environment is show in Figure 8.

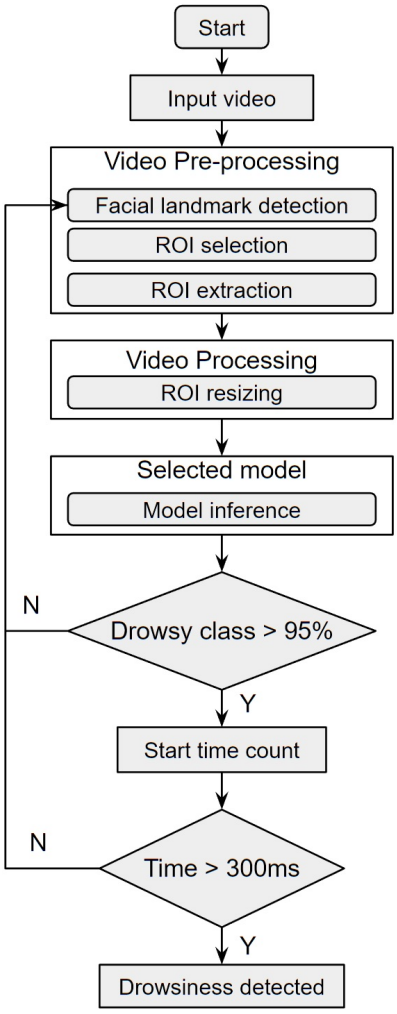


Figure 8. Drowsiness detection process flowchart.

4. Experimental Results

4.1. CNN Validation

Using Equations (2)–(5), the metrics were calculated. Table 3 presents the averages of each metric and its standard deviation corresponding to the validation in the training of the CNNs, seen in Section 3.4.

As seen in Table 3, during the training validation, the CNN based on ResNet50V2 obtained a higher accuracy with 99.89% ± 0.1% average, followed by InceptionV3 with 99.56% ± 0.1% average accuracy and finally the one based on VGG16 with 99.43% ± 0.1% average accuracy. In the case of drowsiness detection, an important metric is the Recall, because the goal is to reduce as much as possible the false negatives of the Drowsy class. Thus, the CNN based on ResNet50V2 obtained the best average Recall with 99.82% ± 0.2%.

Table 3. Metrics in Training (Validation).

CNN Based on	Class Name	Precision	Recall	F1-Score	Accuracy
InceptionV3	Not drowsy	0.9945 ± 0.002	0.9967 ± 0.001	0.9956 ± 0.001	0.9956 ± 0.001
	Drowsy	0.9967 ± 0.001	0.9945 ± 0.002	0.9956 ± 0.001	
VGG16	Not drowsy	0.9928 ± 0.002	0.9951 ± 0.003	0.9934 ± 0.001	0.9934 ± 0.001
	Drowsy	0.9951 ± 0.003	0.9928 ± 0.002	0.9934 ± 0.001	
ResNet50V2	Not drowsy	0.9982 ± 0.002	0.9996 ± 0.001	0.9989 ± 0.001	0.9989 ± 0.001
	Drowsy	0.9996 ± 0.001	0.9982 ± 0.002	0.9989 ± 0.001	

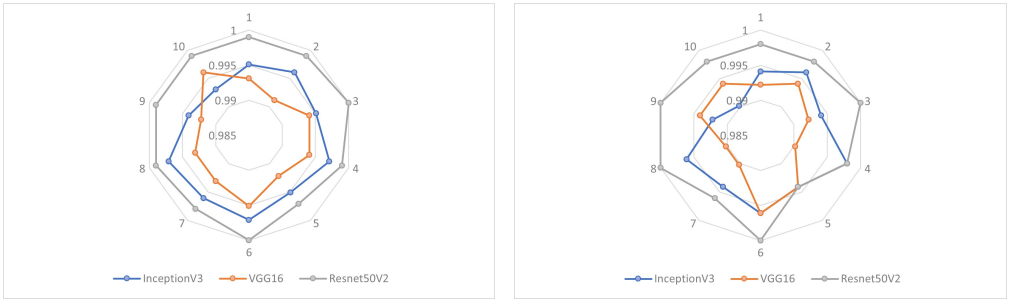
The comparative boxplot for the performance evaluation of CNNs in the training validation phase is presented in Figure 9, where the performance variations of Table 3 in the accuracy and recall metrics of the Drowsy class of each of the three networks in 10 training samples can be observed. In Figure 9a, the CNN based on ResNet50V2 presented a better performance in data validation with a median of 99.9% of the 10 training runs. Similarly, in Figure 9b, the ResNet50V2-based CNN had the best Recall performance of the Drowsy class with a median of 99.8%, thus minimizing the false negatives of the Drowsy class.

(**a**) (**b**)

Figure 9. Boxplot of two evaluation metrics for validation. Boxplots show median (solid line), minimum, maximum values with outliers shown as points. (**a**) Accuracy of the three networks. (**b**) Recall of Drowsy class of the three networks.

Figure 10 shows the radial behavior of the three CNNs, for the two metrics of the 10 trainings (experiments) performed. In Figure 10a, the CNN based on ResNet50V2 had a nearly constant behavior in the 10 experiments. On the other hand, the network based on VGG16 had a lower behavior compared to the other two networks, presenting a maximum accuracy of 99.61% in the last (10) experiment, and a minimum accuracy of 99.12% in the second experiment. Meanwhile, in Figure 10b, the CNN based on RedNest50V2 also presents a better performance compared to the other two networks. In the fifth experiment of this network, a 99.41% recall in the Drowsy class was obtained, being the minimum

value obtained, and likewise, the CNN based on VGG16 presented a lower performance in the recall of the Drowsy class.



(**a**) (**b**)

Figure 10. Radial behavior of two training metrics. (**a**) Accuracy of the three networks in radial. (**b**) Recall of Drowsy class of the three networks in radial.

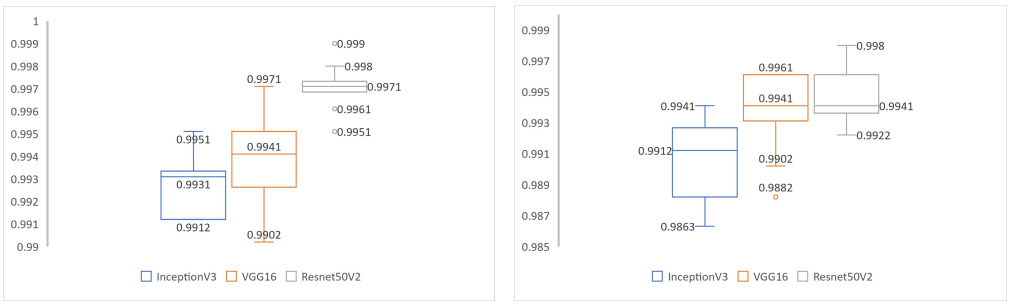
4.2. CNN Testing Evaluation

When testing the CNNs, it was observed from the evaluation metrics presented in Table 4 that the CNN based on ResNet50V2 obtained the highest accuracy with 99.71% ± 0.1% average, followed by VGG16 with 99.39% ± 0.2% average. At the same time, the ResNet50V2-based CNN had the highest Recall with 99.47% ± 0.2% average for the Drowsy class.

Table 4. Metrics in Testing (Test).

CNN Based on	Class Name	Precision	Recall	F1-Score	Accuracy
InceptionV3	Not drowsy Drowsy	0.9908 ± 0.003 0.9957 ± 0.002	0.9957 ± 0.002 0.9908 ± 0.003	0.9928 ± 0.001 0.9927 ± 0.001	0.9927 ± 0.001
VGG16	Not drowsy Drowsy	0.9937 ± 0.003 0.9941 ± 0.005	0.9941 ± 0.005 0.9937 ± 0.003	0.9939 ± 0.002 0.9939 ± 0.002	0.9939 ± 0.002
ResNet50V2	Not drowsy Drowsy	0.9948 ± 0.002 0.9994 ± 0.001	0.9994 ± 0.001 0.9947 ± 0.002	0.9971 ± 0.001 0.9971 ± 0.001	0.9971 ± 0.001

The overall accuracy and recall of the Drowsy class of 10 experiments for each of the three networks is presented in Figure 11. Where, in Figure 11a, it was observed that the ResNet50V2-based network had the best performance with a median of 99.71% accuracy. Meanwhile, in Figure 11b, it can be seen that the VGG16-based network and ResNet50V2-based network had the same median value with a 99.41% recall in class Drowsy. It can also be observed that the network based on VGG16 presents an optimal performance compared to the training results presented in Figure 9.



(**a**) (**b**)

Figure 11. Boxplot of two evaluation metrics in test. (**a**) Accuracy of the three networks. (**b**) Recall of Drowsy class of the three networks.

The radial behavior of the 10 test experiments are shown in Figure 12, showing the two evaluation metrics of overall accuracy and recall of the Drowsy class. Where, in Figure 12a, the ResNet50V2-based network performed better than the other two networks. Whereas, in Figure 12b, it was observed that the ResNet50V2 and VGG16-based networks performed similarly in the results of the 10 experiments. Thus, the performance of the VGG16-based network was significantly improved compared to the training results presented in Figure 10.

(**a**) (**b**)

Figure 12. Radial behavior of two test metrics. (**a**) Accuracy of the three networks in radial; (**b**) Recall of Drowsy class of the three networks in radial.

4.3. CNN Visual Result

To compare the behavior of the CNN architectures in the drowsiness detection and classification process, the best test results for each of the CNNs were considered, taking into account the highest accuracy and recall of the Drowsy class. Based on Figure 12, for the InceptionV3-based CNN, the fourth experiment with 99.51% accuracy and 99.41% recall was considered; for the VGG16-based CNN, the sixth experiment with 99.71% accuracy and 99.61% recall was considered and for the ResNet50V2-based CNN, the fifth experiment with 99.9% accuracy and 99.8% recall was considered.

For a better understanding of the behavior, the gradient-weighted class activation mapping (Grad-CAM) [33,34] method was used. Using Grad-CAM, it is possible to visualize the regions that are important for detection; this method seeks to identify parts of the image that guide the CNN to make the final decision for class determination. The method involves generating a heat map representing the ROI regions with the highest relevance for classification of the received input image.

Figure 13 shows five examples of different scenarios for visualization of the heat maps with Grad-CAM, in scenarios 1 and 2, the driver is in a normal state, i.e., no drowsiness; in scenario 3, the driver is in a wakeful state, which is the transition to drowsiness; in scenarios 4 and 5, the driver is drowsy. In each heat map, the red color represents the regions of highest importance for the prediction of each of the three trained CNNs, and the blue color represents the regions of lowest importance. The five examples were tested for each CNN, generating ROIs I-1 to I-5 corresponding to the outputs of the InceptionV3-based CNN, V-1 to V-5 corresponding to the VGG16-based CNN and R-1 to R-5 corresponding to the ResNet50V2-based CNN. In ROIs I-1 to I-5, we observed that the heat maps focus on the right eye (I-1), lower left eye and nose (I-2 and I-3), lower right eye (I-4) and the whole right eye (I-5). While the heat maps of ROIs V-1 to V-5 have a focus on V-1 and V-2 on the right eye, V-3 and V-4 on the left nose and cheek and V-5 on the right nose and cheek. In ROIs R-1 to R-5, it was observed that in R-1 and R-2, the heat map was on the right eye and part of the nose; in R-3, R-4 and R-5, it was focused between both eyes. Below each ROI, the Grad-CAM display is shown with its respective percentage prediction for drowsiness and non-drowsiness. Considering the example scenarios and the respective classification, it can be stated that the ResNet50V2-based CNN presents a higher focus on the eyes for better drowsiness detection.

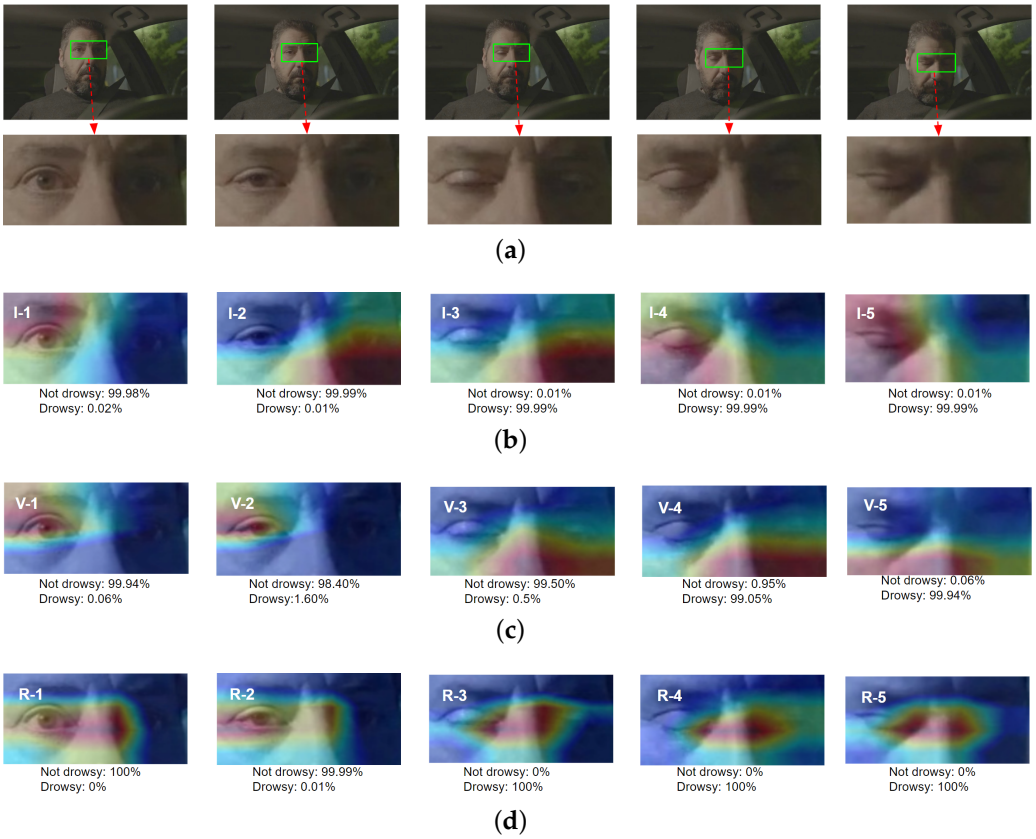


Figure 13. Visualized with Grad-CAM. (**a**) ROI extraction in 5 scenes. (**b**) Grad-CAM InceptionV3. (**c**) Grad-CAM VGG16. (**d**) Grad-CAM ResNet50V2.

4.4. CNN Processing Results

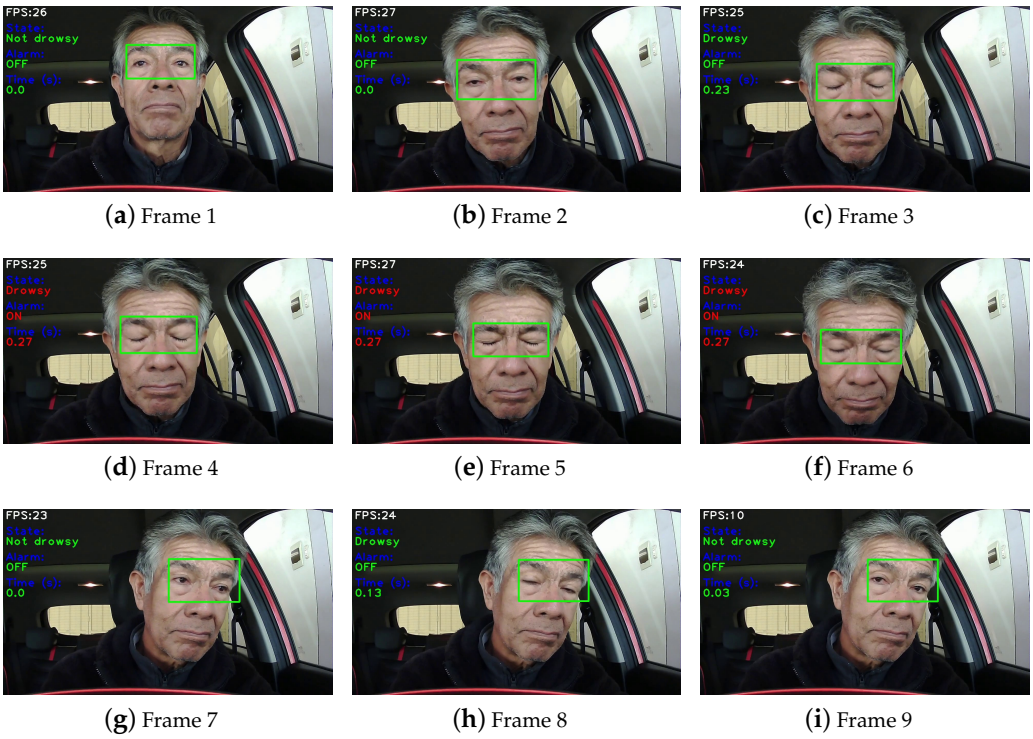
Other results of the training of the CNNs used are the file size, total number of parameters of each network and the training time, where the first two were constant, while the last one could vary in each training performed (10 experiments) for each of the three networks. For testing the behavior of the CNNs, the response time of each architecture was obtained in the 10 experiments performed. These results can be seen in Table 5.

Table 5. CNN processing results.

	Results in Training			**Results in Test**
	Training Time	**File Size (KB)**	**Total Params**	**Response Time**
InceptionV3	6.2 min ± 3 s	182,072	29,997,786	137.8 ms
VGG16	6.1 min ± 12 s	111,603	19,325,690	71.3 ms
ResNet50V2	6.2 min ± 5 s	476,612	56,335,802	106.5 ms

4.5. Driver Drowsiness Detection Results

Considering the results, the CNN based on ResNet50V2 was the most optimal in this research; therefore, it is appropriate to perform driver drowsiness detection. In Figure 14, nine consecutive scene analyses are shown, being a sequence of the process with drowsiness. In the examples it is observed that the driver starts in a normal state (Figure 14a), then goes into a state of wakefulness (Figure 14b), closing his eyes with a time of approximately 230 ms (Figure 14c). Then, the driver goes to the drowsy state, which is longer than 300 ms, activating the visual alarm, as can be observed (Figure 14d–f). This is followed by a normal blink (Figure 14g–i). Specifically, in Figure 14h, it can be seen that the system detects the closed eyes with the Drowsy class, but the time is approximately 130 ms, which does not detect it as drowsiness.



(**a**) Frame 1 (**b**) Frame 2 (**c**) Frame 3

(**d**) Frame 4 (**e**) Frame 5 (**f**) Frame 6

(**g**) Frame 7 (**h**) Frame 8 (**i**) Frame 9

Figure 14. Driver drowsiness detection in a real environment: (**a**) Driver is in a normal state. (**b**) Driver is in a wakeful state. (**c**) Driver is with closed eyes, with a time of approximately 230 ms. (**d**–**f**) Driver goes to the drowsy state seen, with time longer than 300 ms, activating the visual alarm. (**g**–**i**) Driver with normal eye blink.

4.6. Comparison and Discussion

A comparison of the accuracy of the facial methods used in the region of interest (ROI) for drowsiness detection is presented in Table 6. This research uses the NITYMED dataset, which differs from others, so the comparison is performed only considering its detection performance.

The accuracy of the methods ranged from 73% to 98.15%, while the proposed method reached an accuracy higher than 99% in the experiments. The methods used by [17,19,20] focused on the whole face using classical face recognition methods. This approach may present limitations when making use of the whole face, where drowsiness prediction may focus on facial features other than the eyes, such as the nose, forehead, cheek, etc. Meanwhile, the methods used by [18,21,22,26] focused on the eyes using Haar cascade and Dlib methods; these two classifiers are widely used but have limitations. Being so, Haar cascade tends to be prone to detecting false positives and lose information on head movements. In addition, the images used are mostly independent of each eye.

The methods used by [23,24] present another approach combining CNN and LSTM; this combination involves the use of convolutional layers of the CNN for feature extraction from the input data to then pass to the LSTM and make sequence prediction. The authors use this method by observing that drowsiness symptoms occur in small time sequences in the state of the eyes; it is a very robust technique. On the other hand, the method employed by [25], makes use of Dlib for the extraction of characteristic eye points and determining their open or closed state using the eye aspect ratio (EAR), which is a technique commonly used in various systems to detect drowsiness. The EAR determines the eye status by blink threshold values, which makes it dependent on those values when characterizing the size of the drivers' eyes; moreover, the thresholds are predefined for most drivers, while those thresholds vary for each driver.

Table 6. Comparison of drowsiness detection methods.

Autor	Facial Method	ROI	Accuracy
Park et al. [17]	VGG-FaceNet	Face	73.06%
Chirra et al. [18]	Haar Cascade	Eyes	96.42%
Zhao et al. [19]	MTCNN	Face	93.623%
Phan et al. [20]	Dlib	Face	97%
Rajkar et al. [21]	Haar Cascade	Eyes	96.82%
Hashemi et al. [22]	Haar Cascade/Dlib	Eyes	98.15%
Alameen and Alhothali [23]	3D-CNN+LSTM	Face	96%
Gomaa et al. [24]	CNN+LSTM	Face	97.31%
Singh et al. [25]	Dlib	Eyes	80%
Tibrewal et al. [26]	Dlib	Eyes	94%
Based on InceptionV3			99.31%
Based on VGG16	MediaPipe	Eyes	99.41%
Based on ResNet50V2			99.71%

This research proposes an efficient method for the correction and extraction of the region of interest of the eyes to be evaluated for drowsiness detection by means of transfer learning using deep learning with three CNNs (InceptionV3, VGG16 and ResNet50V2). In addition, a visual analysis is presented for each CNN, which other authors do not take into account or do not provide. The operation of the system can be downloaded from Supplementary Materials.

5. Conclusions and Future Works

This study presents an approach for drowsiness detection, where an enhancement method is proposed in the area surrounding the eyes to perform region of interest (ROI) extraction. Likewise, three CNNs are used as a basis: InceptionV3, VGG16 and ResNet50V2. A modification in the architecture of the fully connected network used in the classification process is proposed.

For the experiments, a database was created from NITYMED videos. The results were obtained from 10 experiments performed and showed an exceptionally high accuracy in drowsiness detection using the architectures based on the three CNNs mentioned above, with values of 99.31%, 99.41% and 99.71%, respectively. The response times used for drowsiness detection by each CNN were shown to be relatively equivalent, with the VGG16-based CNN showing a small advantage.

In addition, the Grad-CAM visual technique was used to analyze the behavior of each CNN, where the ResNet50V2-based CNN predominantly focuses on the eye region, achieving better performance in drowsiness detection. These results suggest that the proposed approach may be a good alternative for the implementation of the drowsiness detection system. Among the CNNs used, the ResNet50V2-based CNN presented the best performance, and considering the results of the examples in different scenarios (Figure 13), this architecture also presents higher robustness. When comparing the execution time for detection of this CNN with the other two CNNs (Table 5), it can be considered acceptable.

When the system based on this proposal is implemented, it can be considered a valuable tool for the prevention of automobile accidents caused by driver drowsiness.

As future work, we intend to make use of near-infrared (NIR) imaging to better focus on the eye region when there are illumination limitations. As a complement to this work, yawning detection can also be performed for preventive identification of drowsiness. Finally, the authors intend to implement this in an embedded system adapted to vehicular units.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app13137849/s1, Video S1: prueba-1.mp4.; Video S2: prueba-2.avi.

Author Contributions: Conceptualization, methodology and software, R.F. and F.P.-Q.; validation and formal analysis, R.F., F.P.-Q., T.P. and A.B.A.; resources, F.P.-Q., R.J.C.-C. and J.C.H.-L.; data curation, R.F.; writing—original draft preparation, R.F. and F.P.-Q.; writing—review and editing, T.P. and A.B.A.; supervision, A.B.A. and F.P.-Q.; project administration, F.P.-Q.; funding acquisition, F.P.-Q., R.J.C.-C. and J.C.H.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to that the person included in the test of this research gave his consent. Being a first degree relative (father) of the first author. While the other person belongs to the NITYMED database.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The public database used in this paper is NITYMED (night-time yawning–microsleep–eyeblink–driver distraction), which can be found in https://datasets.esdalab.ece.uop.gr/ (accessed on 2 November 2022).

Acknowledgments: The research was supported by the Institutional laboratory for research, entrepreneurship and innovation in automatic control systems, automation and robotics (LIECAR) of the University of San Antonio Abad del Cusco UNSAAC.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. PAHO. Road Safety. 2022. Available online: https://www.paho.org/en/topics/road-safety (accessed on 9 February 2023).
2. Gestión. Some 265 People Died Each Month of 2022 in Traffic Accidents in Peru (Spanish). 2022. Available online: https://gestion.pe/peru/unas-265-personas-murieron-cada-mes-del-2022-en-accidentes-de-transito-en-peru-noticia/ (accessed on 9 February 2023).
3. ONSV. Road Accident Report and Actions to Promote Road Safety (Spanish). 2022. Available online: https://www.onsv.gob.pe/post/informe-de-siniestralidad-vial-y-las-acciones-para-promover-la-seguridad-vial/ (accessed on 9 February 2023).
4. Albadawi, Y.; Takruri, M.; Awad, M. A Review of Recent Developments in Driver Drowsiness Detection Systems. *Sensors* **2022**, *22*, 2069. [CrossRef] [PubMed]
5. Reddy, P.V.; D'Souza, J.; Rakshit, S.; Bavariya, S.; Badrinath, P. A Survey on Driver Safety Systems using Internet of Things. *Int. J. Eng. Res. Technol.* **2022**, *11*. [CrossRef]
6. Weng, C.H.; Lai, Y.H.; Lai, S.H. Driver drowsiness detection via a hierarchical temporal deep belief network. In Proceedings of the Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, 20–24 November 2016; Springer: Berlin/Heidelberg, Germany, 2017; pp. 117–133.
7. Abtahi, S.; Omidyeganeh, M.; Shirmohammadi, S.; Hariri, B. YawDD: A yawning detection dataset. In Proceedings of the 5th ACM Multimedia Systems Conference, Singapore, 19 March 2014; pp. 24–28.
8. Fusek, R. Pupil localization using geodesic distance. In Proceedings of the 13th International Symposium, ISVC 2018, Las Vegas, NV, USA, 19–21 November 2018; Volume 11241, pp. 433–444. [CrossRef]
9. Ghoddoosian, R.; Galib, M.; Athitsos, V. A realistic dataset and baseline temporal model for early drowsiness detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019.
10. Petrellis, N.; Zogas, S.; Christakos, P.; Mousouliotis, P.; Keramidas, G.; Voros, N.; Antonopoulos, C. Software Acceleration of the Deformable Shape Tracking Application: How to eliminate the Eigen Library Overhead. In Proceedings of the 2021 2nd European Symposium on Software Engineering, Larissa, Greece, 19–21 November 2021; pp. 51–57.
11. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
12. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
14. Torrey, L.; Shavlik, J. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; IGI Global: Hershey, PA, USA, 2010; pp. 242–264.
15. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.L.; Yong, M.G.; Lee, J.; et al. Mediapipe: A framework for building perception pipelines. *arXiv* **2019**, arXiv:1906.08172.
16. Kwon, K.A.; Shipley, R.J.; Edirisinghe, M.; Ezra, D.G.; Rose, G.; Best, S.M.; Cameron, R.E. High-speed camera characterization of voluntary eye blinking kinematics. *J. R. Soc. Interface* **2013**, *10*, 20130227. [CrossRef] [PubMed]

17. Park, S.; Pan, F.; Kang, S.; Yoo, C.D. Driver drowsiness detection system based on feature representation learning using various deep networks. In Proceedings of the Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, 20–24 November 2016; Springer: Berlin/Heidelberg, Germany, 2017; pp. 154–164.
18. Chirra, V.R.R.; Uyyala, S.R.; Kolli, V.K.K. Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State. *Rev. d'Intell. Artif.* **2019**, *33*, 461–466. [CrossRef]
19. Zhao, Z.; Zhou, N.; Zhang, L.; Yan, H.; Xu, Y.; Zhang, Z. Driver fatigue detection based on convolutional neural networks using EM-CNN. *Comput. Intell. Neurosci.* **2020**, *2020*, 7251280. [CrossRef] [PubMed]
20. Phan, A.C.; Nguyen, N.H.Q.; Trieu, T.N.; Phan, T.C. An Efficient Approach for Detecting Driver Drowsiness Based on Deep Learning. *Appl. Sci.* **2021**, *11*, 8441. [CrossRef]
21. Rajkar, A.; Kulkarni, N.; Raut, A. Driver drowsiness detection using deep learning. In *Applied Information Processing Systems: Proceedings of ICCET 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 73–82.
22. Hashemi, M.; Mirrashid, A.; Beheshti Shirazi, A. Driver safety development: Real-time driver drowsiness detection system based on convolutional neural network. *SN Comput. Sci.* **2020**, *1*, 289. [CrossRef]
23. Alameen, S.A.; Alhothali, A.M. A Lightweight Driver Drowsiness Detection System Using 3DCNN with LSTM. *Comput. Syst. Sci. Eng.* **2023**, *44*, 895–912. [CrossRef]
24. Gomaa, M.W.; Mahmoud, R.O.; Sarhan, A.M. A CNN-LSTM-based Deep Learning Approach for Driver Drowsiness Prediction. *J. Eng. Res.* **2022**, *6*, 59–70. [CrossRef]
25. Singh, J.; Kanojia, R.; Singh, R.; Bansal, R.; Bansal, S. Driver Drowsiness Detection System: An Approach by Machine Learning Application. *arXiv* **2023**, arXiv:2303.06310.
26. Tibrewal, M.; Srivastava, A.; Kayalvizhi, R. A deep learning approach to detect driver drowsiness. *Int. J. Eng. Res. Technol.* **2021**, *10*, 183–189.
27. Grishchenko, I.; Ablavatski, A.; Kartynnik, Y.; Raveendran, K.; Grundmann, M. Attention mesh: High-fidelity face mesh prediction in real-time. *arXiv* **2020**, arXiv:2006.10962.
28. Liu, P.; Guo, J.M.; Tseng, S.H.; Wong, K.; Lee, J.D.; Yao, C.C.; Zhu, D. Ocular Recognition for Blinking Eyes. *IEEE Trans. Image Process.* **2017**, *26*, 5070–5081. [CrossRef]
29. Kumari, P.; KR, S. An optimal feature enriched region of interest (ROI) extraction for periocular biometric system. *Multimed. Tools Appl.* **2021**, *80*, 33573–33591. [CrossRef] [PubMed]
30. Pandey, N.; Muppalaneni, N. A novel drowsiness detection model using composite features of head, eye, and facial expression. *Neural Comput. Appl.* **2022**, *34*, 13883–13893. [CrossRef]
31. Ahmed, M.; Laskar, R. Eye center localization using gradient and intensity information under uncontrolled environment. *Multimed. Tools Appl.* **2022**, *81*, 7145–7168. [CrossRef]
32. Caelen, O. A Bayesian interpretation of the confusion matrix. *Ann. Math. Artif. Intell.* **2017**, *81*, 429–450. [CrossRef]
33. Selvaraju, R.R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; Batra, D. Grad-CAM: Why did you say that? *arXiv* **2016**, arXiv:1611.07450.
34. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.