

## Article

# An Efficient Approach for Detecting Driver Drowsiness Based on Deep Learning

Anh-Cang Phan <sup>1,\*</sup>, Ngoc-Hoang-Quyen Nguyen <sup>1</sup>, Thanh-Ngoan Trieu <sup>2</sup> and Thuong-Cang Phan <sup>3</sup>

<sup>1</sup> Faculty of Information Technology, Vinh Long University of Technology Education, Vinh Long 85110, Vietnam; nnhquyen0701@gmail.com

<sup>2</sup> La Faculté Sciences et Techniques, Université de Bretagne Occidentale, 29200 Brest, France; ngoan.trieuthanh@etudiant.univ-brest.fr

<sup>3</sup> College of Information and Communication Technology, Can Tho University, Can Tho 90000, Vietnam; ptcang@cit.ctu.edu.vn

\* Correspondence: cangpa@vlu.edu.vn; Tel.: +84-918-204-917

**Abstract:** Drowsy driving is one of the common causes of road accidents resulting in injuries, even death, and significant economic losses to drivers, road users, families, and society. There have been many studies carried out in an attempt to detect drowsiness for alert systems. However, a majority of the studies focused on determining eyelid and mouth movements, which have revealed many limitations for drowsiness detection. Besides, physiological measures-based studies may not be feasible in practice because the measuring devices are often not available on vehicles and often uncomfortable for drivers. In this research, we therefore propose two efficient methods with three scenarios for doze alert systems. The former applies facial landmarks to detect blinks and yawns based on appropriate thresholds for each driver. The latter uses deep learning techniques with two adaptive deep neural networks based on MobileNet-V2 and ResNet-50V2. The second method analyzes the videos and detects driver's activities in every frame to learn all features automatically. We leverage the advantage of the transfer learning technique to train the proposed networks on our training dataset. This solves the problem of limited training datasets, provides fast training time, and keeps the advantage of the deep neural networks. Experiments were conducted to test the effectiveness of our methods compared with other methods. Empirical results demonstrate that the proposed method using deep learning techniques can achieve a high accuracy of 97%. This study provides meaningful solutions in practice to prevent unfortunate automobile accidents caused by drowsiness.



**Citation:** 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. <https://doi.org/10.3390/app11188441>

Academic Editor: Rui Araújo

Received: 4 August 2021

Accepted: 8 September 2021

Published: 11 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

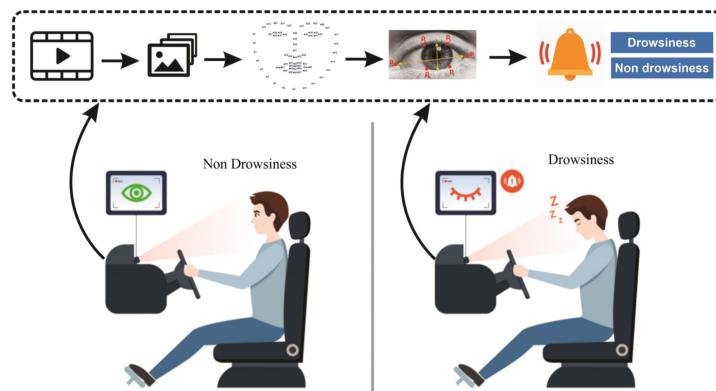


**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The American National Highway Traffic Safety Administration (<https://www.nhtsa.gov> (accessed on 4 August 2021)) has an estimated 100,000 accidents reported each year mainly due to drowsy driving. This results in more than 1550 deaths, 71,000 injuries, and 12.5 billion dollars of property damage. According to the National Safety Council (<https://www.nsc.org> (accessed on 4 August 2021)), 13% of drivers admitted to falling asleep behind the wheel at least once a month and 4% of them resulted in accidents. Morgenthaler et al. announced that drowsiness is one of the main causes of traffic accidents in their study [1]. It is estimated that about 10–15% of car accidents are related to lack of sleep. The sleep questionnaire obtained from professional drivers [2] showed that more than 10.8% of drivers are drowsy while driving at least once a month, 7% had caused a traffic accident, and 18% had near-miss accidents due to drowsiness. These alarming statistics point to the need for capable systems for monitoring drowsy drivers to prevent unfortunate traffic accidents that may occur.

In recent years, building intelligent systems for drowsy driver detection has become a necessity to prevent road accidents. Therefore, it requires a lot of research to design robust alert methods to recognize the level of sleepiness while driving. Many studies focused on constructing the smart alert techniques for intelligent vehicles that can automatically avoid traffic accidents caused by falling asleep, as illustrated in Figure 1. Rateb et al. [3] introduced real-time driver drowsiness detection for an android application using deep neural networks. A minimal network structure was proposed based on facial landmarks to identify drowsy drivers. The method presented a lightweight model and achieved an accuracy of more than 80%. This study focused only on eye facial landmarks without detecting the yawning of the drivers. Moreover, the method was based on a multilayer perceptron classifier with three hidden layers, which is a limitation that leads to low accuracy. Fatigue detection using Raspberry Pi 3 was provided by Akalya et al. [4] by processing driver's faces and eyes images. A Haar cascade classifier was applied to detect the blink duration of the driver, and the eye aspect ratio (EAR) was computed by the Euclidean distance between the eyes.



**Figure 1.** General model of the drowsy detection system.

Mohana and Sheela [5] presented a method of drowsiness detection based on eye closure and yawning detection. They recognized eyes and mouths on faces to detect eye closure and yawning. The limitation of this work is that the eye blink threshold is fixed and the boundary yawn value is 10, consecutively. Jie and Lau [6] proposed a vision-based real-time driver alert system for monitoring drivers' drowsiness and distraction conditions. However, this study only focused on eye characteristics, ignoring yawns, and the eye-opening threshold was also fixed, leading to limitations on the recognition of people with small eyes. In addition, if the drivers did not continuously stare at the camera for 5 s, the alert threshold was set to 0; thus, it was not possible to detect sleepy drivers. Ramos et al. [7] presented a method based on eye movement and yawning using facial landmarks to accurately detect driver drowsiness. However, the limitation of this study also concerns the fixed eye-opening threshold, which is not suitable for people with eyes smaller than the threshold. Sukrit et al. [8] provided a driver drowsiness detection system using a random forest classifier based on eye aspect ratio and eye closure ratio. The method was able to achieve an accuracy of 84%. Shivani et al. [9] used the Haar cascade algorithm for driver drowsiness detection by calculating the eye aspect ratio and a blink counter variable. The driver was determined to be drowsy when the counter reached a threshold value. The Haar cascade classifier for drowsiness detection is a classical approach to compute the eye opening–closure ratios. It usually requires parameter tuning when it is applied for drowsiness detection.

Many studies based on the deep learning approach using convolutional neural networks (CNNs) have been introduced to detect drowsy drivers. A drowsiness detection method using CNN-based machine learning for android applications was introduced by Rateb et al. [10]. The method detected facial landmarks by a camera and passed through a CNN model to detect drowsy driving, with an average accuracy of 83.33%.

Zuopeng et al. [11] introduced a driver fatigue detection method using a proposed EM-CNN to detect the states of the eyes and mouth from the region of interest images. The proposed algorithm, EM-CNN, showed an accuracy of 93.62%. Biswal et al. [12] proposed an IoT-based smart alert system for drowsy driver detection using Raspberry Pi3 and Pi camera modules to make a persistent recording of face landmarks for eye detection. This method was based on the idea of determining blinks through an eye aspect ratio (EAR) and Euclidean distance of the eye. Although this method reached a high accuracy of 97.1% for the experiment, it had many limitations for practical systems because it only focused on eye detection, and an EAR threshold was pre-defined and unchanged for all drivers ( $EAR < 0.25$ ) to detect drowsiness. Ajinkya et al. [13] introduced a driver drowsiness detection method using deep learning with an average accuracy of 96%. This method included two stages of the pre-processing and drowsiness detection. Haar feature-based cascade classifiers, a machine learning-based approach, were used to detect the mouth and eye regions of the drivers in pre-processing. To identify drowsiness, the frames of the mouth and eye regions were then forwarded to the proposed CNN models, which were the basic CNN models using four conv2d, four max-pooling layers, and two dense layers to identify the state of blinks and yawns in the given time threshold. There were only two features of the eyes and mouth trained on these two CNN models for drowsiness detection without considering other physiological factors. Moreover, experiments were implemented on a small dataset of 2423 subjects, including 1192 people with closed eyes, and 1231 people with open eyes. In addition, Madhav et al. [14] presented a deep learning approach to detect driver drowsiness. This method extracted the characteristics of the driver's eyes on each frame using Dlib's API and passed through a classification model to predict the state of drowsiness. Adam was used as the optimizer and the average accuracy reached 94%.

Unfortunately, most of these studies focused on analyzing the mouth and eye regions to detect blinks and yawns without considering other regions of the head and face. Therefore, they are not accurate enough and have revealed many limitations on drowsiness detection for actual systems, because dozing is a natural state in the human body amongst other behaviors. Moreover, there is a problem surrounding physiological measures in these works, in that they may not be feasible in practice. They are hard to apply to actual systems, since the measuring devices are not available on vehicles and are often uncomfortable for drivers. Drowsiness is a natural phenomenon in a human body that happens due to different factors causing distraction. It is not merely determined by recording the number blinks or yawns. Meanwhile, the deep learning approach is able to learn all features automatically for drowsiness detection. In this paper, we propose two approaches for detecting drowsiness using the techniques of facial landmark identification and deep learning. We make the following contributions: (1) the collection of a dataset of drowsiness and non-drowsiness from images and videos monitored through a camera for experiments; (2) the proposal of two methods for drowsiness detection and prediction using facial landmarks and deep learning. In the facial landmark-based method, we improve drowsiness detection by determining the appropriate thresholds of blinks and yawns for each driver. In the deep learning-based method, we propose the use of two adaptive deep neural networks with the transfer learning approach for drowsiness detection. We designed and perfected these networks developed on the advanced networks of MobileNet-V2 and ResNet-50V2, which are more efficient in terms of memory and complexity. The proposed networks are very good feature extractors, since they can capture and learn relevant features of drowsiness automatically; (3) the use of a transfer learning approach to well solve the problem of fast training, a small training dataset and accuracy improvement; (4) the comparison and discussion the performance, accuracy, and advantages of the proposed methods with other methods. The experimental results show that the proposed methods achieve an accuracy of 97%. It can be concluded that the proposed method using deep learning has similar efficiency to other effective methods using a combination of behavioral and physiological features, but it is more feasible and has outstanding advantages.

The rest of the paper is presented as follows: Section 2 presents the background on identifying facial features and deep learning models. Details of the proposed methods are presented in Section 3. We provide some experimental results and evaluation in Section 4. Finally, we give conclusions in Section 5.

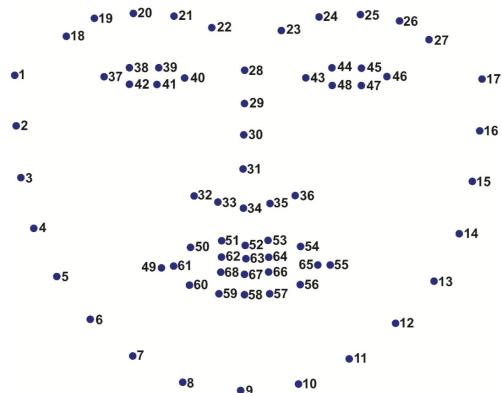
## 2. Background

### 2.1. Drowsiness

Sleep [15] is the natural cyclical rest state of the body and mind. In this state, people often close their eyes and lose consciousness partially or completely, thereby reducing their response to external stimuli. Sleep is not an option. It is necessary and inevitable to help the body rest and restore energy. The term “microsleep” or “drowsiness” is defined as brief and involuntary intrusions of sleep that can occur at any time due to fatigue or a prolonged conscious effort. Microsleep can last for a few seconds, and during this time, the brain falls into a rapid and uncontrolled sleep, which can be extremely dangerous, especially in the case of driving or in situations demanding focused attention. There are some signs that show that drivers are not awake: yawning, blinking repeatedly and difficulty opening eyes, the inability to concentrate, the inability to keep the head straight, a distracted mind, feelings of tiredness, and blurred vision.

### 2.2. Identification of Facial Landmarks

Kazemi and Sullivan [16] presented a method that precisely estimates the positions of facial landmarks using a training set of labeled facial landmarks on images. This method can be used for real-time detection to identify the facial features after detecting faces on an image. This face features detector identifies 68 main positions ( $x, y$  coordinates) on human faces, as shown in Figure 2.



**Figure 2.** Illustration of 68 landmarks on a human face.

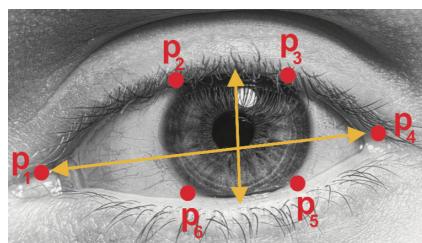
The positions of facial landmarks are defined as follows: jaw: points 1 to 17; right eyebrow: points 18 to 22; left eyebrow: points from 23 to 27; nose: points 28 to 36; right eye: points 37 to 42; left eye: points 43 to 48; mouth: points 49 to 68. In this work, we only use 32/68 feature points, namely coordinates of the left eye, right eye, and mouth to calculate eye-opening (EAR) and mouth-opening ( $LIP_{distance}$ ).

#### 2.2.1. Blink Detection

To determine the feature positions, we apply facial feature detection including the eyes, eyebrows, nose, ears, and mouth [17,18]. We extract specific facial structures by knowing the index of specific parts of a face. For blink detection, we are interested in eyes features. Each eye is represented by six coordinates, as illustrated in Figure 3, starting at the left corner of the eye and going clockwise around the rest of the area.

$$EAR = \frac{(|p_2 - p_6| + |p_3 - p_5|)}{(2|p_1 - p_4|)}. \quad (1)$$

Based on Soukupova and Cech [18], we can derive Equation (1), reflecting the eye-opening value, where  $p_1, \dots, p_6$  are features located on a 2D face. The numerator of Equation (1) calculates the distance between the eye features vertically, while the denominator calculates the distance between the eye features horizontally. Denominator balancing is suitable since there is only one set of horizontal points but there are two sets of vertical points. The eye-opening value is almost unchanged when the eye is opened, but it quickly decreases to 0 when blinks occur. Using this method, we can avoid image processing techniques and simply rely on eye-opening values to determine blinking actions.



**Figure 3.** Example of 6 facial landmarks related to eyes.

#### 2.2.2. Yawning Detection

Yawning is characterized by mouth-opening. Like blink detection, facial features are used to detect an open mouth. The mouth-opening value is a parameter used to determine whether a mouth is open. It is calculated by subtracting the mean of all points in the upper mouth with the mean of all points in the lower mouth [19], as in Equation (2).

$$LIP_{distance} = |LIP_{Top} - LIP_{Low}|. \quad (2)$$

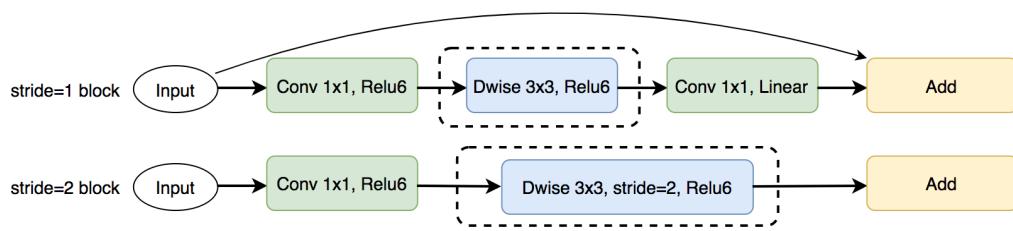
If a mouth-opening value calculated from the frames is higher than a mouth-opening threshold, it is determined to be yawning. An alarm sounds if a person yawns for more than a fixed value threshold. Small loopholes that are caused by talking and eating are ignored.

#### 2.3. Deep Neural Networks for Drowsiness Prediction

Taking advantage of deep learning techniques, we apply deep neural networks with MobileNet-V2 and ResNet-50V2 for driver drowsiness detection. A detailed description of the network architectures and an analysis of their advantages is presented as follows.

##### 2.3.1. MobileNet-V2 Network

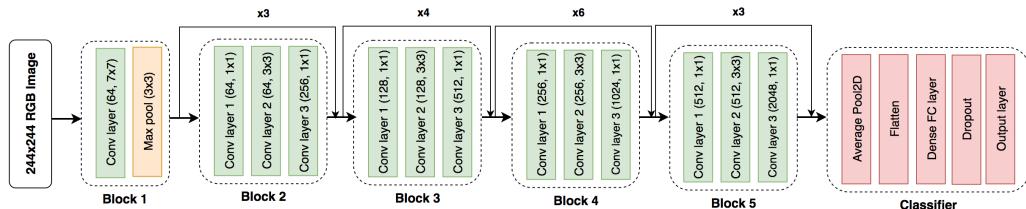
MobileNet-V2 [20] is a significant improvement over MobileNet-V1, enhancing image recognition capabilities on mobile devices including classification, object detection, and semantic segmentation. Figure 4 illustrates the MobileNet-V2 network architecture. Overall, MobileNet-V2 models are faster with the same accuracy across the entire latency spectrum, using 2 times fewer operations, 30% fewer parameters, and being 30–40% faster than MobileNet-V1. MobileNet-V2 has the advantages of being fast, lightweight, and having high accuracy, which makes it suitable for training with limited datasets. MobileNet-V2 is trained on datasets such as ImageNet, COCO, and VOC [21], and has been applied in object classification such as Fruit Image Classification [22] and Flower Classification [23], with accuracy up to 85.12% and 96%, respectively. With these advantages and practical results, we decided to choose MobileNet-V2 for drowsiness prediction. Besides, we would like to apply this network to build a mobile application for driver drowsiness detection while driving.



**Figure 4.** Illustration of MobileNet-V2 network architecture.

### 2.3.2. ResNet Network

ResNet [24] won the ImageNet Large Scale Visual Recognition Competition in 2015 for image classification, object localization, and object detection. The main challenge in training deep learning models is that the accuracy decreases with the depth level of the networks. ResNet converges very quickly and can be trained with hundreds or thousands of layers. At the same time, ResNet is easy to optimize and can achieve accuracy gains from greatly increased depth, producing better results [24]. ResNet-50V2 was trained on ImageNet and CIFAR-10 datasets [25]. The ResNet-50V2 network architecture is shown in Figure 5. The results show that ResNet-50V2 performs better and gives better results than ResNet50 and ResNet101 on the same ImageNet dataset. ResNet-50V2 applies the transfer learning technique with ImageNet pre-trained weights, which are used in object classification, such as COVID-19 and pneumonia detection from chest X-ray images and video classification with accuracy up to 94–98%. Taking advantage of ResNet-50V2, we choose this architecture as the training model for driver drowsiness detection.



**Figure 5.** Illustration of ResNet-50V2 network architecture.

### 2.4. Model Evaluation Metrics

A confusion matrix is usually represented as a table to describe the performance of classification models. The values of TP (True Positive), FP (False Positive), TN (True Negative), FN (False Negative) are defined as in Table 1.

**Table 1.** Confusion matrix.

Predicted Class	Actual Class	
	P—Positive	N—Negative
P—Positive	TP	FP
N—Negative	FN	TN

#### 2.4.1. Accuracy

When building a classification model, it is necessary to know the accuracy of that model. In general, it is a ratio of correctly predicted observation to the total observations [26]. Accuracy helps us evaluate the prediction performance of a model on a dataset. Accuracy is calculated as Equation (3).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}. \quad (3)$$

#### 2.4.2. Precision, Recall and $F1$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The higher the precision, the better the model is on positive classification. Precision is calculated as Equation (4).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (4)$$

Recall measures the rate of correct predictions of positive cases across all samples belonging to the positive group. Recall measures the ratio of correctly predicted positive observations to all positive observations. Recall is as calculated as Equation (5).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (5)$$

$F1$  is the weighted average of precision and recall. Therefore, it is more useful in assessing the model on both precision and recall.  $F1$  is calculated as Equation (6).

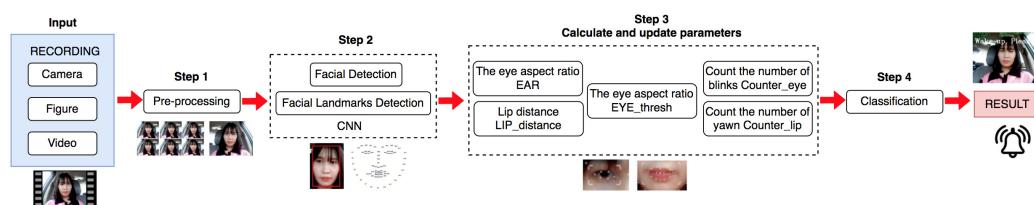
$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (6)$$

### 3. Proposed Methods

In this work, we propose two efficient methods for doze alert systems. The first method is based on a combination of two blink and yawn features (EAR, LIP) using facial landmarks. However, adaptive thresholds of blinks and yawns were calculated appropriately for each driver without needing to pre-determine for all drivers. The second method uses advanced deep learning techniques with the transfer learning approach. Facial images are detected with the SSD- ResNet-10, and then forwarded to the proposed networks to detect drowsiness. We designed and perfected two adaptive deep neural networks developed on the advanced networks of MobileNet-V2 and ResNet-50V2 by making improvements in some layers to adapt the drowsiness detection. In addition, we applied the transfer learning approach to train the proposed networks in order to achieve faster learning, no requirement for large training datasets, and an improvement in classification accuracy. The details of the implementation steps of the two proposed methods are presented below.

#### 3.1. Method 1: Drowsiness Detection and Prediction Based on Facial Landmarks

The proposed method is inspired by the methods introduced in preceding studies [5–7]. However, the threshold of eye-opening is fixed for all drivers, leading to inaccuracies for drivers with large or small eyes. To overcome this problem, we improve the determination of the adaptive eye-opening threshold ( $EAR_{Thresh}$ ) for each driver. The model of the proposed method is shown in Figure 6. The values of the  $EAR$  and  $EAR_{Thresh}$  are first computed for each driver. Then, we determine the drowsiness level of drivers by a comparison of  $EAR$  and  $EAR_{Thresh}$ . This work is carried out repeatedly during driving. This method does not rely on the yawning frequency, since it would be inaccurate in cases where drivers wear a mask or talk while driving. The details of this method are described as follows.



**Figure 6.** Model of the proposed method 1 for detecting drowsiness using facial landmarks.

### 3.1.1. Step 1: Pre-Processing

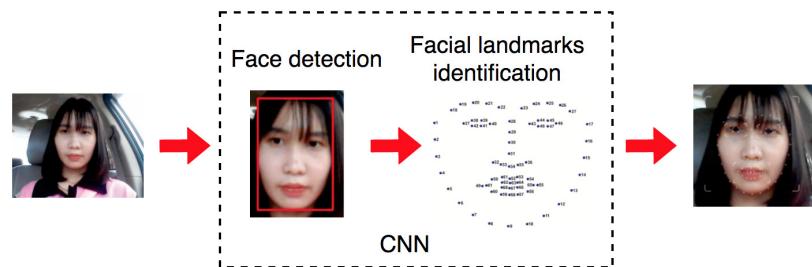
In this step, we extract video frames from input videos. The rate of selecting images from videos is 25 frames per second. These images are flipped to accurately identify the facial landmarks (Figure 7).



**Figure 7.** Illustration of extracting image frames from input videos.

### 3.1.2. Step 2: Detection of Facial Landmarks

We detect and identify facial landmarks on faces in the images as presented in Section 2.2. The facial landmarks considered in this work include the eyebrows, eyes, nose, mouth, and jaw (see Figure 8).



**Figure 8.** An example of detecting faces and facial landmarks.

### 3.1.3. Step 3: Determination of the Eye-Opening Threshold for Each Driver

In this step, we use 32 feature points based on the identification of facial landmarks in step 2 (Figure 9). We determine the coordinates of points on the mouth and eyes.



**Figure 9.** Determination of the coordinates of points on the eyes and mouth.

### 3.1.4. Step 4: Drowsiness Detection and Prediction

The input data are a video tracked from a camera on a vehicle. This video is split into clips with lengths of 300 s. Through the experiments, we tested many different values before reaching the recommended optimal value for  $Counter_{EyeLimit} = 15$  and  $Counter_{YawnLimit} = 25$ . The algorithm by pseudocode for the proposed method is presented in Algorithm 1.

**Algorithm 1** Drowsiness Detection and Prediction

---

**Input:** Input video monitored from a camera  
**Output:** Drowsiness\_Detection = True/False (state of drowsiness or non-drowsiness)

**Begin**

▷ video is split into clips with lengths of 300 s

**for each** clip in video **do**

▷ initialize variables to count blinks and yawns respectively.

$Counter_{eye} = 0;$   
      $Counter_{yawn} = 0;$   
      $Drowsiness\_Detection = \text{False};$

**for each**  $i = 1 \dots n$  **do**

▷  $n$  is the number of image frames extracted from clip

        Calculate  $EAR_i$  by Equation (1);  
         Calculate  $LIP_{distance_i}$  by Equation (2);

**end for**

▷ compute an adaptive eye-opening threshold for each driver

Calculate  $EAR_{Thresh} = \text{Average}(EAR_1, EAR_2, \dots, EAR_n);$

**for each**  $i = 1 \dots n$  **do**

▷ Count blinks and yawns

**if**  $EAR_i \geq EAR_{Thresh}$  **then**  
          $Counter_{eye} += 1;$   
     **end if**  
     **if**  $LIP_{distance_i} > LIP_{Thresh}$  **then**  
          $Counter_{yawn} += 1;$   
     **end if**

**end for**

▷  $Counter_{EyeLimit}$  and  $Counter_{YawnLimit}$  are thresholds of blinks and yawns

**if**  $Counter_{eye} > Counter_{EyeLimit}$  **then**  
      $Drowsiness\_Detection = \text{True};$      ▷ Turn on a driver drowsiness alarm: Wake up, please!

**end if**

**if**  $Counter_{yawn} > Counter_{YawnLimit}$  **then**  
      $Drowsiness\_Detection = \text{True};$      ▷ Turn on a driver drowsiness alarm: Wake up, please!

**end if**

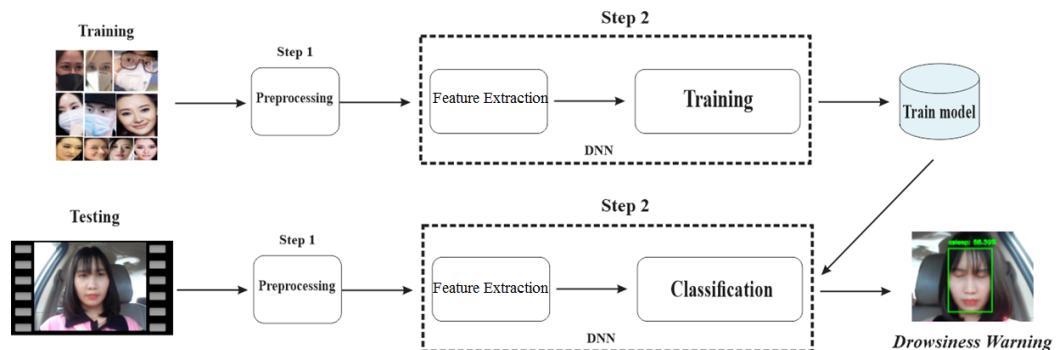
**end for**

**End.**

---

### 3.2. Method 2: Drowsiness Detection and Prediction Using Deep Learning

In this method, the drowsiness detection using deep neural networks (DNNs) includes two phases of the training and testing, as illustrated in Figure 10. In the first phase, we train the proposed network models by training a dataset after pre-processing and feature extraction. In the second phase, we evaluate the network models with a test set for drowsiness detection.



**Figure 10.** Model of the proposed method 2 for detecting drowsiness using deep learning.

#### 3.2.1. Phase 1: Training

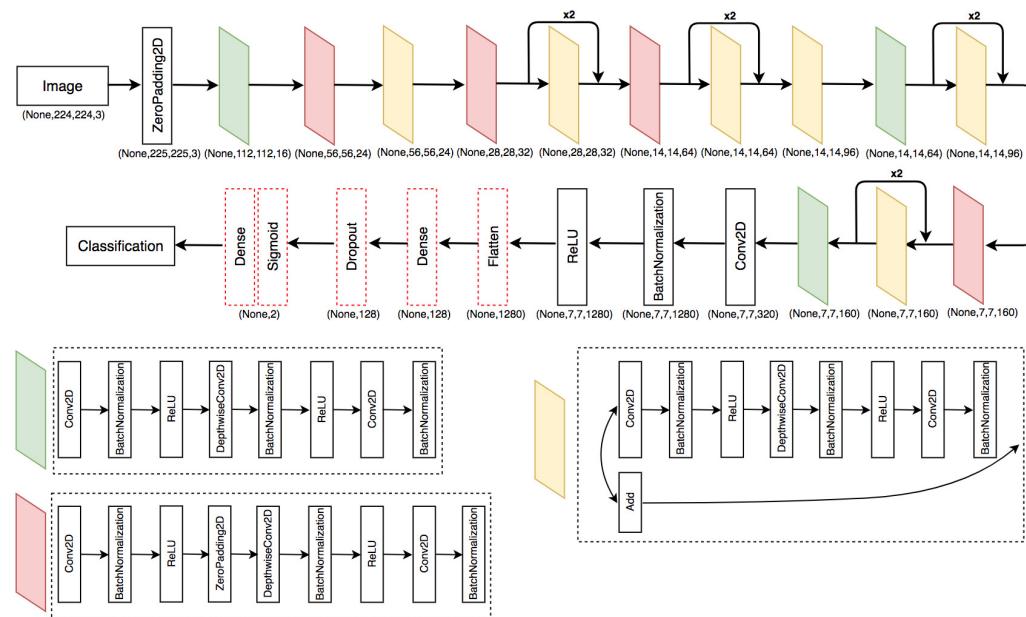
##### Step 1: Pre-Processing

In order to create a good training dataset, we pre-process a set of drowsy and non-drowsy images extracted from videos. In this step, we perform the extraction of images from the input videos. The extraction rate of images from videos is 25 frames per second.

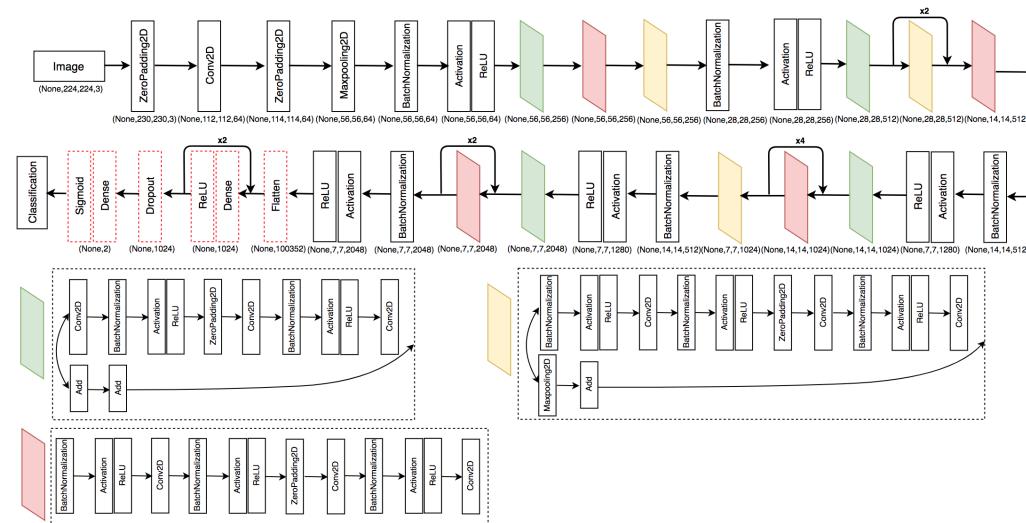
We first detect faces in these images using the SSD network (Single Shot MultiBox Detector) with ResNet-10. Then, the facial images are normalized to the size of  $224 \times 224$  to create the training dataset.

### Step 2: Feature Extraction and Training

In order to extract features and train, the training dataset is passed through deep neural networks (DNNs). From the advantages of the neural networks described in Section 2.3, we develop two adaptive networks for drowsiness detection, inspired by ResNet-50V2 and MobileNet-v2, by making improvements in some layers of these networks to adapt the drowsiness detection. We construct an adaptive network by adding layers such as Flatten, Relu (Densen), Dropout, Dense (Sigmoid) and by suppressing the AveragePooling2D layer of the MobileNet-v2 network, as shown in Figure 11. It aims at solving the problem of overfitting and helps the model to converge faster. Additionally, we propose another adaptive network by adding layers such as Flatten, two Relu (Densen), Dropout, Dense (Sigmoid) to the ResNet-50V2, as shown in Figure 12.



**Figure 11.** The network architecture of the proposed method 2 developed from MobileNet-v2.



**Figure 12.** The network architecture of the proposed method 2 developed from ResNet-50V2.

Taking advantage of the transfer learning approach, the proposed networks are pre-trained on datasets of Bing Search API, Kaggle, and RMFD. We use the pre-trained weights and re-train them on our training dataset to fine-tune the parameters of these networks. This leads to faster learning, a shorter training time, and no requirement for large training datasets. During the training process, we decide to stop the training phase when the Loss\_value is not reduced.

### 3.2.2. Phase 2: Testing

After training, the input data for testing are images extracted from videos tracking from a camera to detect drowsiness by applying the proposed network models from training phase. The results of this phase are detailed in the below section.

## 4. Experiments

### 4.1. Dataset and Installation Environment

#### 4.1.1. Dataset Description

It is not safe to make field experiments for driver's drowsiness on real roads. Therefore, in order prevent any risk for drivers, we generate an experimental dataset consisting of drowsy and non-drowsy faces extracted from datasets of Bing Search API, Kaggle, RMFD, and iStock by selecting images and videos recorded from cameras that are related to the driver's drowsy states. The main advantages of this work are its efficiency, low cost, safety, and the ease of data collection. In these images and videos, driving environments are relatively similar to actual road experiments. The dataset contains 16,577 images of faces with 2659 images of drowsy states, 3789 images of non-drowsy states, and 10,129 images of both states. In particular, for method 2, the dataset of 6448 images is divided at the ratio of 80% (5158) for training and 20% (1290) for testing. The dataset directly affects the accuracy of drowsiness detection and prediction. The number and size of the datasets are described in Table 2.

**Table 2.** Experimental datasets.

Status	Type	Number	Num Faces	Size
Drowsy	Image	2659	2659	229 MB
Non-drowsy	Image	3789	3789	64 MB
Both	Video	21	10,129	1316 MB

#### 4.1.2. Installation Environment

In order to compare and evaluate the two proposed methods, we program them in PHP language on the same environment, Visual Studio Code with Windows 10. The configuration of the computer is 8 GB RAM and Nvidia Geforce GPU. The library for training the proposed network models is Tensorflow version 2.2.

### 4.2. Scenarios and Parameters

#### 4.2.1. Scenarios

Experiments of the proposed methods are conducted based on three scenarios, as presented in Table 3.

**Table 3.** Scenarios of the proposed methods for experiments.

Scenario	Method	Technique of Feature Extraction and Prediction
1	1	Facial landmarks
2	2	Adaptive deep neural network developed from MobileNet-V2
3	2	Adaptive deep neural network developed from ResNet-50V2

#### 4.2.2. Experimental Parameters

During the experiments, we fine-tune the parameters before reaching the optimal parameter values to obtain an improvement in accuracy. Details of the parameters of the three scenarios are presented in Tables 4 and 5.

**Table 4.** Experimental parameters for scenario 1.

Scenario	LIP <sub>Thresh</sub>	Counter <sub>yawn</sub>	Counter <sub>YawnLimit</sub>	Counter <sub>eye</sub>	Counter <sub>EyeLimit</sub>	Time (s)
1	60	0	25	0	15	300

**Table 5.** Experimental parameters for scenarios 2 and 3.

Scenario	Batch Size	Learning Rate	Epoch	Score Converter	Image Size	Additional Layers	Removed Layers
2	30	$1 \times 10^{-4}$	30	sigmoid	$224 \times 224$	Dense (Relu) Dropout (0.5) Dense (Sigmoid)	Average Pooling2D
3	250	$1 \times 10^{-5}$	30	sigmoid	$224 \times 224$	2 Dense (Relu) Dropout (0.5) Dense (Sigmoid)	None

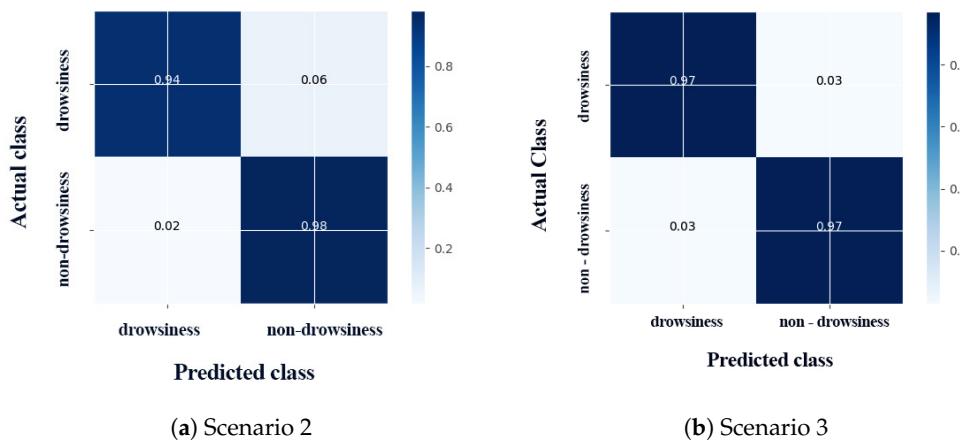
#### 4.3. Experimental Results

##### 4.3.1. Training Model Evaluation

The training phase is only carried out for scenarios 2 and 3 using deep learning techniques. To evaluate the performance of the network models before detecting drowsiness, we use a confusion matrix and compute metrics of the accuracy, Loss\_value, and training time.

##### Metrics of Precision, Recall, and Accuracy

We determine metrics of the precision, recall, and accuracy defined by the confusion matrix, as presented in Section 2.4. Figure 13 shows a confusion matrix for scenarios 2 and 3. In scenario 2, the TP (true positive) and TN (true negative) values of drowsiness and non-drowsiness prediction are 94% and 98%, respectively. In scenario 3, the TP and TN results are 97% and 97% for drowsiness and non-drowsiness, respectively. The metrics of the precision, recall, and accuracy are shown in Table 6. The accuracy of the proposed network models after the training process is 96% for scenario 2 and 97% for scenario 3.



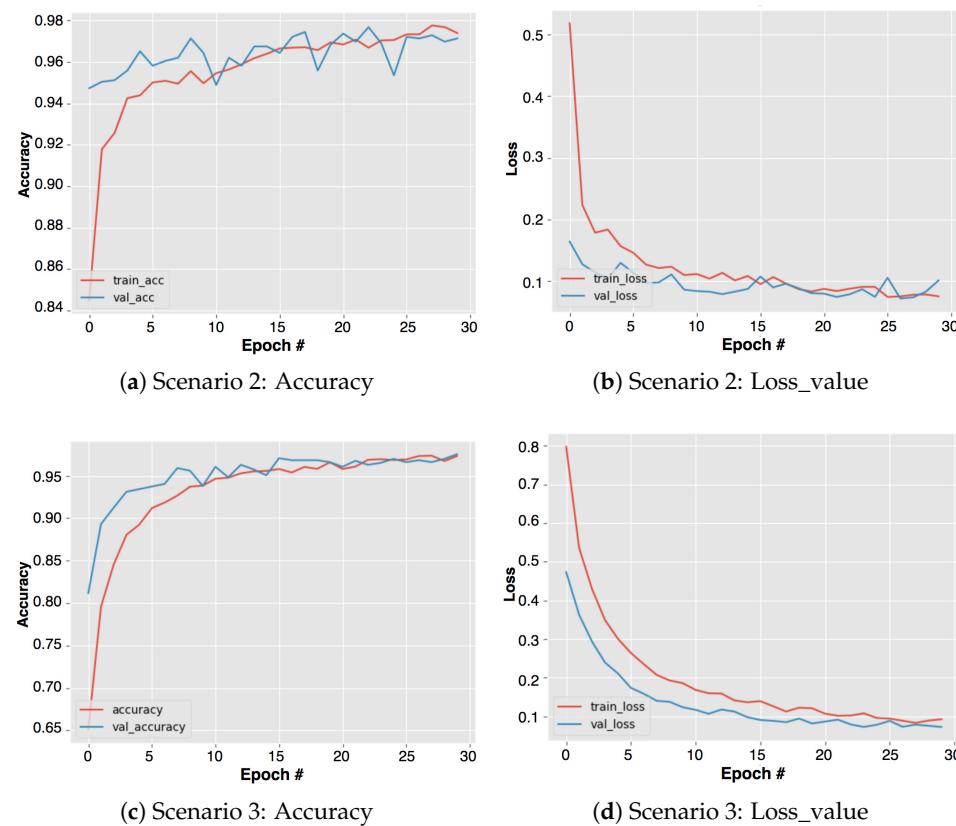
**Figure 13.** Confusion matrix for scenarios 2 and 3.

**Table 6.** Results of training in scenarios 2 and 3.

Scenario	Label	Precision	Recall	F1-Score	Support	Accuracy
2	Drowsy	0.93	0.97	0.95	532	0.96
	Non-drowsy	0.98	0.96	0.97	758	
3	Drowsy	0.96	0.97	0.97	532	0.97
	Non-drowsy	0.98	0.97	0.98	758	

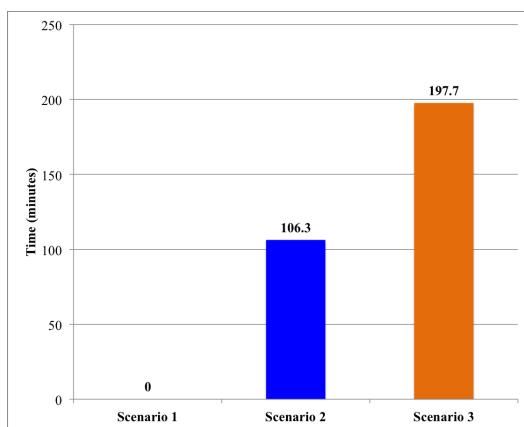
#### Metrics of Loss\_Value

Figure 14 shows the results of the Loss\_value and accuracy of the two scenarios after several training steps (epoch) with the learning\_rate of  $1 \times 10^{-4}$  and  $1 \times 10^{-5}$ , respectively. The y-axis represents the accuracy and Loss\_value of the two scenarios; the x-axis represents the epochs. The accuracy gradually increases while the epoch increases and the Loss\_value decreases. When the epoch reaches 30 and the Loss\_value is not improved further, we stop training and move to the testing phase. As observed in Figure 14, Loss\_value and Accuracy metrics in scenario 3 tend to be more stable than scenario 2. Loss\_value in scenario 2 is higher than scenario 3 and tends to increase at 30 epochs. This means that the level of error (information loss) for training and validation is high in scenario 2. As a result, the accuracy of recognition in scenario 2 is not higher than scenario 3.

**Figure 14.** Charts of accuracy and Loss\_value for training and validation in scenarios 2 and 3.

#### Training Time

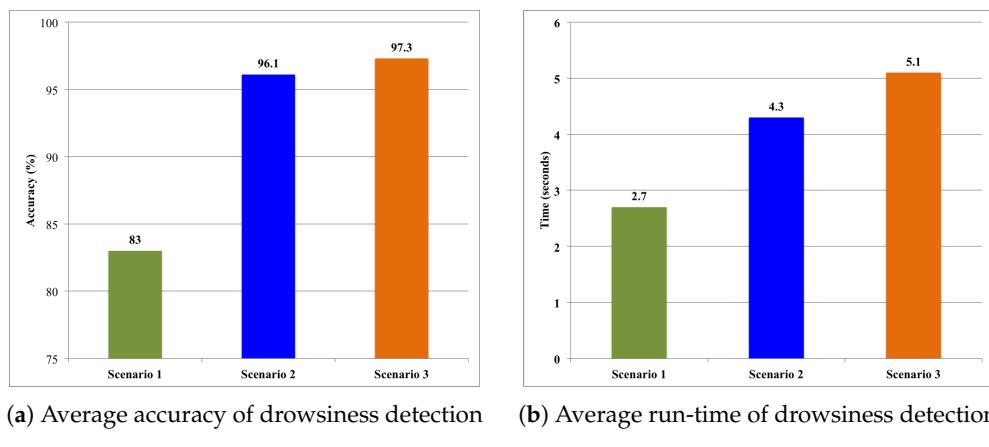
Figure 15 shows the training time of the three scenarios. Since scenario 1 has no training process, the training time is 0. For the other two scenarios, the training time is 106 min 3 s for scenario 2, and 197 min 7 s for scenario 3. The training time of scenario 2 is lower than that of scenario 3.



**Figure 15.** Training time of the three scenarios.

#### 4.3.2. Testing Results

Experiments for testing are conducted on a total of 30 different drivers from videos consisting of drowsy and non-drowsy states. Figure 16 shows the results of the average accuracy and run-time in the testing phase for drowsiness detection. The average accuracy of scenario 1 is 83%, scenario 2 is 96.1%, and scenario 3 is 97.3%. As can be seen, scenarios 2 and 3 outperform scenario 1 in terms of accuracy. On the other hand, the average prediction time of the three scenarios is 2.7, 4.3, and 5.1 s for three scenarios 1, 2, and 3, respectively. Scenario 1 gives results faster than the remaining two scenarios. However, giving quick results will lead to incorrect drowsiness detection since it can be confused between blinking or eyes closing but not sleeping.



**Figure 16.** Average accuracy and run-time of drowsiness detection in testing phase.

#### 4.3.3. Drowsiness Detection and Prediction

The experimental results of drowsiness detection of the three proposed scenarios are presented in Tables 7 and 8. Table 7 provides experiments for drowsiness detection from videos for the three proposed scenarios. In cases a and e, it does not work (no detection) for scenario 1 due to the inability to determine the eye-opening threshold in input frames; scenarios 2 and 3 give correct results of detection but the determination time of scenario 2 is slower; the driver completely sleeps in scenario 2, while the driver has their head tilted in scenario 3. On the other hand, in scenario 2, face noise occurs when the driver's face is placed too close to the camera. In case b, all three scenarios can detect the driver, but scenario 2 gives incorrect results. In case c, all three scenarios give the correct results. In case d, scenario 1 does not give a warning when the driver is drowsy, scenario 2 gives correct results, while scenario 3 only shows signs of eye strain. It can be seen that scenario 3 has more signs of drowsiness, which manifest as feelings of tiredness and eye strain without having a clear face.

**Table 7.** Illustration of drowsiness detection and prediction on videos for three scenarios.

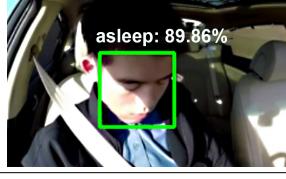
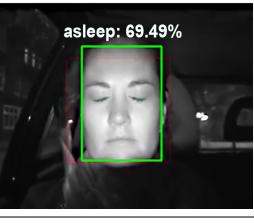
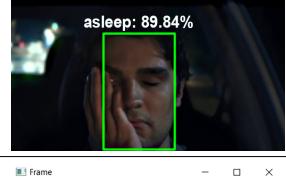
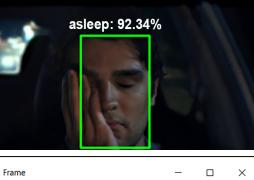
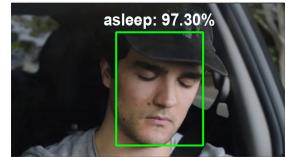
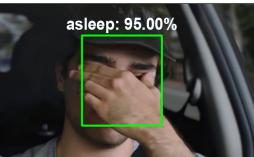
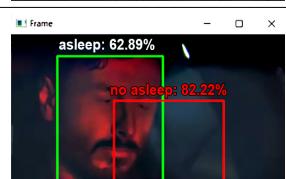
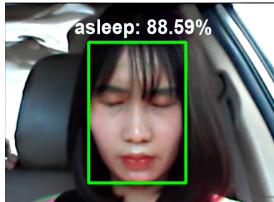
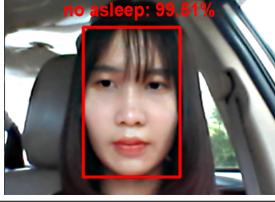
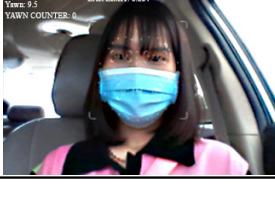
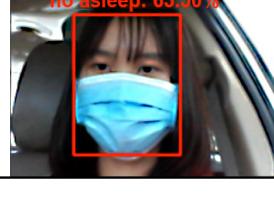
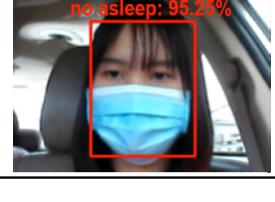
Scenario	1	2	3
a.	No detection		
b.			
c.			
d.			
e.	No detection		

Table 8 shows the experimental results for the proposed scenarios. These scenarios give the correct results in all cases. However, in case a, scenarios 2 and 3 give results when the driver completely closes their eyes and shows signs of tilting their head. In case b, scenarios 2 and 3 still give correct identification results in the case where the driver wears a mask; scenario 2 is bases the identification on the head being tilted and eyes being closed, while scenario 3 requires only a slight tilt of the head and lightly closed eyes; scenario 1 only relies on the eye-opening value, which can lead to confusion when the driver only closes their eyes but does not completely fall asleep; thus, in scenario 1, we need to adjust the blink counter depending on each driver to alert when the driver completely falls asleep. In cases c and d, the driver is in non-drowsy status and the results of all three scenarios are correct.

**Table 8.** Some experimental results of drowsiness detection for three scenarios.

Scenario	1	2	3
a.			
b.			
c.			
d.			

#### 4.3.4. Comparison and Discussion

Table 9 represents a comparison of the accuracy of the drowsiness detection methods. The accuracy of these methods achieves from 83% to 97.1%, while the proposed method reaches the accuracy of 97%. Most of the preceding methods in [5–7,10–14] focus on analyzing the eye and mouth areas to detect blinks and yawns. They use Haar cascade classifiers for facial landmark detection and deep learning or machine learning for drowsiness prediction. These are classical approaches that use hand-engineered features to estimate the state of the eyes and mouth, which have revealed many limitations on drowsiness detection. Besides, Haar cascade classifiers tend to be prone to false positive detections and require parameter tuning when they are applied for detection. Although these methods reach the accuracy of 96–97.1% for the experiment, they have many limitations on practical systems because they only rely on two key features of blinks and yawns for drowsiness detection without considering other physiological factors. Moreover, the blink and yawn time thresholds to detect drowsiness are pre-defined for all drivers, while these thresholds are different for each driver. However, drowsiness is a natural phenomenon in the human body that happens due to different factors causing distraction. Therefore, drowsiness prediction is not accurate enough if it is only based on blinks and yawns. Hence, we need take some signs of head and eye movements (and other physiological features) into account to be able to make more accurate predictions. We propose an efficient method that uses deep learning with two advanced deep neural networks developed from MobileNet-V2 and ResNet-50V2.

**Table 9.** Accuracy comparison of drowsiness detection methods.

Authors	Method	Accuracy
Mohana and Sheela [5]	Drowsiness detection based on eye closure and yawning using facial landmarks and Haar cascade classifiers.	85%
Jie and Lau [6]	Facial landmark detector and Raspberry Pi for a real-time driver alert system.	83.39%
Ramos et al. [7]	Driver drowsiness detection based on eye movement and yawning using facial landmark analysis.	85%
Mohammed Shinoy et al. [10]	Driver drowsiness detection model using convolutional neural network techniques for android application.	83.33%
Zuopeng Zhao et al. [11]	Driver fatigue detection based on convolutional neural networks using EM-CNN.	93.623%
Biswal et al. [12]	Facial landmark predictor and Raspberry Pi3 Model B Module.	97.1%
Ajinkya Rajkar et al. [13]	Driver drowsiness detection using deep learning.	96%
Madhav Tibrewal et al. [14]	A deep learning approach to detect driver drowsiness.	94%
The proposed method with scenario 1	Facial landmark detection.	83%
The proposed method with scenario 2	Deep learning network developed from MobileNet-V2.	96%
The proposed method with scenario 3	Deep learning network developed from ResNet-50V2.	97%

In our work, we address a drowsy driver alert system that has been developed by the techniques of the facial landmarks with adaptive thresholds and deep learning with the transfer learning approach. Through experimental results and Table 9, the training results in three proposed scenarios, 1, 2 and 3, have the accuracy of 83%, 96%, and 97%, respectively. Figure 14 shows that the accuracy in scenario 3 is more stable and the loss is also lower than scenarios 1 and 2 at 30 epochs. It can be seen that scenario 3 gives better results in terms of accuracy and loss but its training time is longer than scenario 2. We can conclude that scenarios 2 and 3 give better results than scenario 1. The outstanding advantage of the proposed method with scenario 1 is that it proceeds with calculations and gives quick results. The limitation of the method is that the determination of the eye-opening threshold is not accurate when the driver does not look at the camera upon the system starting. In addition, the prediction only based on eye movements is not accurate, since it leads to confusion if the driver blinks continuously or has too-small eye-openings. Moreover, the determination of input thresholds that affects the prediction accuracy even causes the failure of no detection for doze detection. In a case where the driver only closes his eyes but is not sleepy, it is difficult for scenario 1 to give accurate results. The drowsiness detection based on blinks and yawns is not accurate enough. Hence, we need take some signs of movements of the head and eyes (head titled, inability to keep the head straight) into account to be able to make more accurate predictions, and this has been implemented in scenarios 2 and 3.

The experimental results have highlighted the main advantage of the proposed method as follows: (a) two proposed methods with three scenarios use facial landmark and deep

learning approaches. The former detects the driver's state from blinks and yawns based on the appropriate thresholds defined as dynamically suitable for each driver instead of a predefined threshold for all drivers. The latter uses advanced deep learning techniques for drowsiness detection without estimating the defined thresholds. Two adaptive deep neural networks are proposed and developed on the advanced networks of MobileNet-V2 and ResNet-50V2, which are more efficient in terms of memory and complexity compared to the basic CNN models in the prior works. The use of these proposed deep neural networks is motivated by the fact that they are very good feature extractors, since they can capture and learn relevant features of drowsiness from an image or video at different levels similar to a human brain; (b) instead of focusing on the two regions of the eyes and mouth, the proposed deep neural networks analyze the videos and detect not only two features of blinks and yawns, but also the driver's activities in every frame to automatically learn all features of drowsiness such as head orientation, yawning, eyelid opening, eye blinking, pupil diameter, and gaze direction, etc. for drowsiness prediction. Experiments were conducted on a large dataset to test the effectiveness of the proposed method in accuracy and reliability. Therefore, our method achieves the high accuracy of 97%; (c) leveraging the advantage of transfer learning, the proposed networks are trained on our training dataset. This solves the problem of small training datasets, training time, and keeps the advantage of the deep neural networks; (d) both of our methods can detect drowsiness accurately in the case of the driver wearing a mask (Table 9), while some methods do not consider or provide accuracy in this case. Besides, the proposed methods can recognize drowsiness well for drivers with very small eyes.

## 5. Conclusions

Most of the traditional methods for drowsiness detection are based on behavioral factors, while some require expensive sensors and devices to measure sleepiness, and may even interfere with the driving process, distracting drivers. Therefore, in this paper, we propose two methods with three scenarios for driver's drowsiness detection systems. The proposed method with scenario 1 uses facial landmarks to detect drowsiness. This method analyzes the videos and detects drivers' faces in every frame using image processing techniques. Facial landmarks are determined in order to compute the eye aspect ratio (EAR) and the mouth-opening value ( $LIP_{distance}$ ) to detect drowsiness based on adaptive thresholds. We propose an improvement of the eye-opening threshold for each driver without using a pre-defined threshold for everyone, as in preceding studies. In each video frame, the frequent detection of eye blinking and yawning will help to properly compute the drowsiness level. The driver is alerted when the blink and yawn thresholds reach the adaptive maximum thresholds. However, the drowsiness detection based on blinking and yawning is not accurate enough, since drowsiness has different surrounding factors. Therefore, we propose method 2, with two scenarios for a drowsy alert system using deep learning techniques with the transfer learning approach. We design two adaptive deep neural networks developed from MobileNet-V2 and ResNet-50V2 for scenarios 2 and 3, respectively. Method 2 analyzes the videos and detects the driver's activities in every frame to automatically learn all features for drowsiness detection. It takes advantage of deep neural networks to extract all features and movements of the head and face. This method does not require the definition of input thresholds as in method 1, especially the eye-opening and yawning thresholds. Additionally, method 2 uses a combination of many typical signs of drowsiness to give accurate results, such as eye-opening, head movements, eyebrows, mouth, etc. Moreover, we leverage the advantage of transfer learning to pre-train the proposed networks on datasets of Bing Search API, Kaggle, and RMFD. We then use the pre-trained weights and re-train them on our training dataset to fine-tune the parameters of these networks. This helps to solve the problem of small training datasets and gives a fast training time.

Experiments were conducted to test the efficacy of the proposed approaches. The results show that the proposed methods can achieve a high accuracy of 97% using deep

learning techniques. Method 2 with scenario 3 provides more accurate results than scenario 2 and method 1 because it detects most of the drowsiness in various experimental contexts. Some experimental comparisons between the proposed methods and the preceding methods are made to discuss the advantage and limitation of the proposed methods. We also highlight the limitations of the preceding studies that affect the effectiveness in actual applications. Method 2 improves the accuracy of drowsiness detection compared with eyelid and mouth movement-based methods.

From the above results, the proposed method using deep learning techniques can be useful for monitoring the fatigue of drivers to give early warning of falling asleep, avoiding unfortunate traffic accidents. The experimental results show that the method is feasible and adaptable to the development of applications for doze alert systems, especially mobile applications. This study helps to prevent automobile accidents caused by falling asleep behind the wheel. However, a key requirement for drowsiness detection is that the solutions work in real time or near real time. Thus, we will make an improvement in our further research using big data analysis to build a real-time system for drowsiness detection.

**Author Contributions:** Conceptualization, A.-C.P. and T.-C.P.; methodology, A.-C.P., N.-H.-Q.N. and T.-C.P.; software, A.-C.P. and N.-H.-Q.N.; validation, T.-N.T. and T.-C.P.; formal analysis, A.-C.P. and N.-H.-Q.N.; investigation, A.-C.P. and T.-C.P.; resources, A.-C.P. and T.-C.P.; data curation, T.-N.T. and N.-H.-Q.N.; writing—original draft preparation, A.-C.P. and N.-H.-Q.N.; writing—review and editing, T.-N.T. and T.-C.P.; visualization, T.-N.T. and N.-H.-Q.N.; supervision, T.-C.P.; project administration, A.-C.P. and T.-C.P.; funding acquisition, no funding. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are available on request by contacting the corresponding author.

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

## References

1. Morgenthaler, T.I.; Lee-Chiong, T.; Alessi, C.; Friedman, L.; Aurora, R.N.; Boehlecke, B.; Brown, T.; Chesson, A.L., Jr.; Kapur, V.; Maganti, R.; et al. Practice parameters for the clinical evaluation and treatment of circadian rhythm sleep disorders. *Sleep* **2007**, *30*, 1445–1459. [[CrossRef](#)] [[PubMed](#)]
2. Marjorie, V.; Heather, M.E.; Neil, J.D. Sleepiness and sleep-related accidents in commercial bus drivers. *Sleep Breath.* **2009**, *14*, 39–42.
3. Jabbar, R.; Al-Khalifa, K.; Kharbeche, M.; Alhajyaseen, W.K.M.; Jafari, M.; Jiang, S. Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques. *Procedia Comput. Sci.* **2018**, *130*, 400–407. [[CrossRef](#)]
4. Chellappa, A.; Reddy, M.S.; Ezhilarasie, R.; Suguna, S.K.; Umamakeswari, A. Fatigue Detection Using Raspberry Pi 3. *Int. J. Eng. Technol.* **2018**, *7*, 29–32. [[CrossRef](#)]
5. Mohana, B.; Sheela Rani, C.M. Drowsiness Detection Based on Eye Closure and Yawning Detection. *Int. J. Recent Technol. Eng.* **2019**, *8*, 1–13.
6. Wong, J.Y.; Lau, P.Y. Real-Time Driver Alert System Using Raspberry Pi. *ECTI Trans. Electr. Eng. Electron. Commun.* **2019**, *17*, 193–203. [[CrossRef](#)]
7. Ramos, A.L.A.; Erandio, J.C.; Mangilaya, D.H.T.; Carmen, N.D.; Enteria, E.M.; Enriquez, L.J. Driver Drowsiness Detection Based on Eye Movement and Yawning Using Facial Landmark Analysis. *Int. J. Simul. Syst. Technol.* **2019**, *20*, 1–8. [[CrossRef](#)]
8. Mehta, S.; Dadhich, S.; Gumber, S.; Bhatt, A.J. Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio. In Proceedings of the International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Jaipur, India, 26–28 February 2019; pp. 1333–1339.
9. Sheth, S.; Singhal, A.; Ramalingam, V.V. Driver Drowsiness Detection System using Machine Learning Algorithms. *Int. J. Recent Technol. Eng.* **2020**, *8*, 990–993.
10. Jabbar, R.; Shinoy, M.; Kharbeche, M.; Al-Khalifa, K.; Krichen, M.; Barkaoui, K. Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, Doha, Qatar, 2–5 February 2020.
11. 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](#)]
12. Biswal, A.K.; Singh, D.; Pattanayak, B.K.; Samanta, D.; Yang, M.H. IoT-Based Smart Alert System for Drowsy Driver Detection. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1–13. [[CrossRef](#)]

13. Ajinkya Rajkar, N.K.; Raut, A. Driver Drowsiness Detection Using Deep Learning. In *Applied Information Processing Systems: Proceedings of ICCET 2021*; Springer: Singapore, 2021; pp. 73–82.
14. Tibrewal, M.; Srivastava, A.; Kayalvizhi, R. A Deep Learning Approach To Detect Driver Drowsiness. *Int. J. Eng. Res. Technol.* **2021**, *10*, 183–189.
15. Kingman, P.S.; Jesse, B.; Forrest, C.; Kate, G.; James, K.; Roger, K.; Anne, T.M.; Sharon, L.M.; Allan, I.P.; Susan, R.; et al. Drowsy driving and automobile crashes. In *Ncsdr/nhtsa Expert Panel on Driver Fatigue and Sleepiness*; National Highway Traffic Safety Administration: Rockville, MD, USA, 1999.
16. Kazemi, V.; Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1867–1874.
17. Sagonas, C.; Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 3–8 December 2013; pp. 397–403.
18. Cech, J.; Soukupova, T. Real-time eye blink detection using facial landmarks. In Proceedings of the 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia, 3–5 February 2016; pp. 1–8.
19. Kumar, C.P.; Thamanam, S.; Karthik, M.; Likitha, S. Driver Drowsiness Monitoring System Using Visual Behavior and Machine Learning. *Ann. Rom. Soc. Cell Biol.* **2021**, *25*, 19969–19977.
20. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
21. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
22. Xiang, Q.; Wang, X.; Li, R.; Zhang, G.; Lai, J.; Hu, Q. Fruit image classification based on Mobilenetv2 with transfer learning technique. In Proceedings of the 3rd International Conference on Computer Science and Application Engineering, Sanya, China, 22–24 October 2019; pp. 1–7.
23. Dai, W.; Dai, Y.; Hirota, K.; Jia, Z. A Flower Classification Approach with MobileNetV2 and Transfer Learning. In Proceedings of the 9th International Symposium on Computational Intelligence and Industrial Applications (ISCIIA2020), Beijing, China, 31 October–3 November 2020.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
25. Rahimzadeh, M.; Attar, A. A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. *Inform. Med. Unlocked* **2020**, *19*, 100360. [[CrossRef](#)] [[PubMed](#)]
26. Joshi, R. Accuracy, precision, recall & f1 score: Interpretation of performance measures. *Retrieved April 2016*, 1, 2016.