electronics

MDPI

Article

4D: A Real-Time Driver Drowsiness Detector Using Deep Learning

Israt Jahan [1], K. M. Aslam Uddin [1,*], Saydul Akbar Murad [2], M. Saef Ullah Miah [2], Tanvir Zaman Khan [1], Mehedi Masud [3], Sultan Aljahdali [3] and Anupam Kumar Bairagi [4,*]

[1] Department of Information & Communication Engineering, Noakhali Science and Technology University, Noakhali 3814, Bangladesh
[2] Faculty of Computing, College of Computing & Applied Sciences, Universiti Malaysia Pahang, Pekan Pahang 26600, Malaysia
[3] Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
[4] Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh
* Correspondence: aslam@nstu.edu.bd (K.M.A.U.); anupam@ku.ac.bd (A.K.B.)

Abstract: There are a variety of potential uses for the classification of eye conditions, including tiredness detection, psychological condition evaluation, etc. Because of its significance, many studies utilizing typical neural network algorithms have already been published in the literature, with good results. Convolutional neural networks (CNNs) are employed in real-time applications to achieve two goals: high accuracy and speed. However, identifying drowsiness at an early stage significantly improves the chances of being saved from accidents. Drowsiness detection can be automated by using the potential of artificial intelligence (AI), which allows us to assess more cases in less time and with a lower cost. With the help of modern deep learning (DL) and digital image processing (DIP) techniques, in this paper, we suggest a CNN model for eye state categorization, and we tested it on three CNN models (VGG16, VGG19, and 4D). A novel CNN model named the 4D model was designed to detect drowsiness based on eye state. The MRL Eye dataset was used to train the model. When trained with training samples from the same dataset, the 4D model performed very well (around 97.53% accuracy for predicting the eye state in the test dataset). The 4D model outperformed the performance of two other pretrained models (VGG16, VGG19). This paper explains how to create a complete drowsiness detection system that predicts the state of a driver's eyes to further determine the driver's drowsy state and alerts the driver before any severe threats to road safety.

Keywords: CNN; drowsiness detection; VGG16; VGG19; 4D

check for updates

Citation: Jahan, I.; Uddin, K.M.A.; Murad, S.A.; Miah, M.S.U.; Khan, T.Z.; Masud, M.; Aljahdali, S.; Bairagi, A.K. 4D: A Real-Time Driver Drowsiness Detector Using Deep Learning. *Electronics* **2023**, *12*, 235. https://doi.org/10.3390/electronics12010235

Academic Editor: Dimitris Apostolou

Received: 31 October 2022
Revised: 12 December 2022
Accepted: 23 December 2022
Published: 3 January 2023

1. Introduction

Driver sleepiness detection is an important part of car safety technology for preventing car accidents. Many people use cars to get to and from work every day, to improve their living standards, for comfort, and when they need to get somewhere quickly. Highways and metropolitan areas see heavy traffic as a result of this trend. However, drowsy driving is one of the major causes of road accidents. Accidents can be prevented in two ways: by catching drivers who are getting sleepy early and by setting off alarms. Every year, traffic accidents claim the lives of over 1.3 million individuals. Lack of sleep for drivers is a major factor contributing to accidents. To decrease traffic accidents, technology for driver sleepiness detection systems is needed. The detection of drowsy drivers, i.e., using cameras, sensors, and other tools to warn about and stop fatal crashes, is of tremendous interest. Driver assistance systems are used by automakers, including Tesla [1] , Mercedes-Benz [1], and others. These innovations have aided drivers in preventing collisions. Recently, Samsung and Eyesight teamed up to track a driver's concentration by analyzing facial patterns and features. Their innovations included assisted steering, automatic braking, lane departure warnings, and variable cruise control. The creation of this technology is a significant problem for the scientific and industrial communities.

The development of real-time applications for human safety has been made possible by the development of revolutionary, smart, and human-interacting devices and technologies [2]. One of the crucial factors taken into account by researchers [2] is the ability to identify tiredness with behavioral cues, such as those in the eyes, lips, facial features, etc. However, other methods can be used to identify driver inattention, including those that are vehicle-based and physiology-based (electroencephalography, electrocardiography, etc.) [3]. By increasing models' accuracy and precision, much work has been consistently put into improving drowsiness detection [4–7]. For behavioral measurements, a camera is used to observe the driver's actions, such as head swaying, yawning, and eye blinking, and then the the driver is alerted if any signs of tiredness are found [8,9]. To identify tiredness in a driver, other sorts of measurements, such as subjective measures, are also employed. These actions are based on feedback from the driver, who is asked a series of questions to gauge their level of tiredness. This rating is the basis on which the degree of a driver's tiredness is determined [10,11].

It is widely acknowledged that driver drowsiness contributes significantly to the rising number of accidents on today's highways. Numerous researchers that have found links between driver tiredness and accidents on the road have validated this proof. The number of accidents caused by tiredness is difficult to determine, but it is almost certainly underestimated. To date, researchers have attempted to simulate behavior by establishing associations between tiredness and specific signs pertaining to the car and the driver. Previous methods of drowsiness detection included machine learning algorithms, such as SVM, KNN, and Haar Cascade classifiers [12,13], among others, to make assumptions about the relevant behavior. Despite the fact that many restrictions on these systems were previously noted, in the instance of image classification problems, deep learning algorithms outperform machine learning techniques significantly, and DL algorithms are also more able to handle complex problems than ML algorithms are. The goal of this project is to implement DL algorithms to overcome the shortcomings of the aforementioned techniques and to provide a user-friendly solution for identifying drowsiness at an early stage that can be used on a desktop or other mobile device.

This study suggests a deep-neural-network-based approach called the deep driver drowsiness detector (4D) for detecting driver sleepiness. The methods used previously were often based on the blinking rate and on open and closed eyes. The proposed technique uses features that are learned by using convolutional neural networks to capture numerous facial traits and other nonlinear characteristics. A sigmoid classifier is used to determine whether the driver is sleepy. This technology is used to provide a warning with a sound alarm to prevent road accidents in the case of tiredness or inattention. Future intelligent vehicles built to detect driver drowsiness and analyze driver weariness may use autonomous technologies to help prevent accidents brought on by driver fatigue. The proposed deep networks acquire the necessary features for the job and then forecast whether or not the driver is tired. Three deep neural networks—VGG16 [14], VGG19, and a customized model (4D)—make up the network. We retrieved the image dataset from Kaggle, which included pictures of people's faces in various situations, including those with their eyes closed and eyes open, with some wearing glasses or with hair in front of their faces, etc. The contributions of this research are:

- A convolutional-neural-network-based novel classification model was developed for drowsiness detection on the basis of eye state.
- The results show that the model is capable of classifying eyes as either open or closed.
- A class activation map (CAM) for the proposed model is shown to visualize the learning area of the images that is used to make predictions.

The remainder of this paper is organized as follows. A literature review on solutions for driver sleepiness detection is provided in Section 2. The proposed algorithm and methodology based on CNNs are discussed in Section 3. The experimental results in

Section 4 provide information about the supplied model's precision and effectiveness and about comparisons. The real-time detection implementation using a webcam technique is also discussed in this section. Finally, conclusions and future research directions are offered in Section 5.

2. Related Work

The literature focuses on the issue raised here, and research into related advancements was our primary focus for this literature review. Therefore, our main focus was on three drowsiness detection methods: physiology-, behavior-, and transportation-based indicators. In this research, we suggest a system for detecting drowsiness, training the system/model, and eventually delivering optimal outcomes. Given that it was more exaggerated towards drowsy qualities, it produced accurate and satisfactory outcomes. By concentrating on characteristics such as excessive eye blinking and prolonged eye closure, among others, it was possible to make more accurate forecasts and findings.

In [15], a survey on drowsiness techniques was undertaken. It included a comparison of each of the three measures. It combined the features of EEG and ECG and then investigated the performance by using a support vector machine classifier. The merits and cons of each of these strategies were thoroughly examined. The accuracy of different measurements should be merged into a hybrid system in order to produce an effective sleepiness detection system. We did not think of developing a hybrid model because it was not used in real time.

The authors of [16] created an experiment to determine tiredness in an attempt to address the problem. They used a Raspberry Pi camera and Raspberry Pi 3 module to estimate drivers' levels of tiredness. The regularity of head tilting and eye movement was recorded. The accuracy was estimated to be up to 99.59% in a test on ten subjects. The classifier that it utilized, the Haar cascade, is ineffective for large datasets because it has a high calculation speed and it is rotation-invariant. However, this algorithm is costly and ineffective in scaling and lighting detection.

Maneesha V. Ramesh, Aswathy K. Nair, and Abhishek Kunnath [17] recommended using a multiplexed sensor system in real time with the goal of creating a wireless network of sensors with intelligence to track and identify the real-time sleepiness of the driver. It was made up of several intrusive sensors that repeatedly tracked the person's physiological characteristics and broadcast a first-level warning to both the operator and the occupant. Because this tactic is intrusive and our primary objective is to work on behavioral measures, we opted against utilizing it.

Advanced artificial-intelligence-based methods were utilized in [18] by Challa Yashwanth and Jyoti Singh Kirar. They employed yawning, eye closure, and the distance between the mouth and eyes. Although the presented classifiers were capable of producing reasonable results, there is still room for improvement in their efficiency. The system is difficult to construct and keep up, and training takes much time. By conducting research on further datasets, a drowsiness detection classifier that is more reliable can still be enhanced.

Mardi et al. [19] suggested an electroencephalography (EEG)-based model for detecting drowsiness. To discriminate between drowsiness and alertness, the logarithms of the signal energy and chaotic properties were extracted. The classification was performed by using an artificial neural network, which had an accuracy of 83.3%.

Noori et al. [20] designed a system based on a combination of reliable driving signals, electrooculography, and EEG to detect tiredness. They utilized a feature selection method to find the optimal subset of characteristics. A self-designed network was employed for categorization, and the accuracy was 76.51%.

Picot et al. [21] employed both ocular and cerebral activity. An EEG with a single channel was used to monitor the nervous system. Graphical activity and blinking were used for monitoring and categorization. The blinking characteristics were extracted by using EOG. A fuzzy-logic-based EOG detector was constructed by fusing these two characteristics. The accuracy in this study was 80.6% when tested on a dataset containing twenty different drivers.

The above three systems were quite expensive because they necessitated the attachment of numerous sensors to the driver's body. Additionally, knowing that falling asleep is a possibility might make drivers anxious because their EEG readings may show a combination of tension and sleepiness.

Krajewski et al. [22] created a model based on steering patterns to detect drowsiness. To capture the steering patterns in this model, complex signal processing approaches were used to build three feature sets. Five machine learning techniques, including an SVM and K-nearest neighbor, were used to assess the performance, with an 86% detection accuracy for sleepiness. However, techniques based on driving patterns are greatly influenced by driving behaviors, road conditions, and vehicle attributes. The parameters can be updated by using more complex models, such as neural networks or ensembles, in order to achieve better results. K-NN can be trained very quickly, but when the size or dimension of the dataset is enormous, it runs slowly. This is due to the laziness problem, where all computation is postponed until categorization.

Mandal et al. [23] developed a bus driver tracking system with a vision-based fatigue warning system. An HOG and an SVM were utilized in this study for driver identification and head–shoulder detection, respectively. For face detection, they utilized the OpenCV face detector, while for eye detection, they used the OpenCV eye detector. SVMs have been used to classify tiredness in a number of studies [12]. SVMs have certain limitations, even though they are some of the most effective classification approaches and have been successfully used to solve numerous real-world situations. Selecting the ideal kernel function for a given problem may be the toughest challenge for the support vector approach. The speed of an SVM's training and testing phases is its second flaw.

The authors of [24] presented an extensive large-scale multi-camera dataset that was designed to study real-world drowsiness detection during driving scenarios. The dataset was collected via a multi-camera platform with novel collection strategies that were employed to appeal to the challenges of real-world applications. A machine learning SVM algorithm was implemented here to detect various levels of drowsiness. The presented method was more complex and time-consuming. Special devices were also needed to detect drowsiness. However, our method is more user-friendly, can be implemented by using a desktop or any other mobile device with a camera, and can detect drowsiness in a very short duration (1.39 ms).

3. Materials and Methods

Deep learning focuses on imitating the processes and rules of the human brain [25]. The term "deep learning" is primarily justified because it involves a dense and multi-layered network of artificial neural networks (ANNs). Automatic/implicit feature extraction and selection are performed in deep learning. Deep learning models operate most effectively and deliver superior outcomes when given large amounts of unstructured data as input. Information clustering and target class classification are efficient when using neural networks. They are use the information that one manages, processes, and stores, and they can be thought of as a categorization and clustering layer. When a labeled dataset on which to train a model is given, deep neural networks are proven to be effective in classifying the data by grouping them according to the similarities among the acquired inputs. Neural networks may also extract features from individual images or videos; these are then used to feed algorithms for categorization. Of the different kinds of neural networks, our system's implementation uses a convolutional neural network (CNN).

3.1. Dataset Selection

The model was developed by using the MRL Eye dataset, which consisted of 47,173 images of one eye (open and closed). The ambient illumination and/or changes in the distance between the camera and the driver significantly impact these circumstances [26]. Figure 1 shows datasets of various types of reflections (none, mild, and strong) and lighting conditions (excellent or bad). Thirty-seven different people, both with and without spectacles for the left and right eyes, provided the samples for this study. Additionally, the MRL Eye dataset was based on manually cropped images of the eye region, which were entirely acceptable for use as input for our suggested CNN model. The ratio of the training dataset to the test dataset was 80 to 20. Some images from the collection are presented in Figure 1.

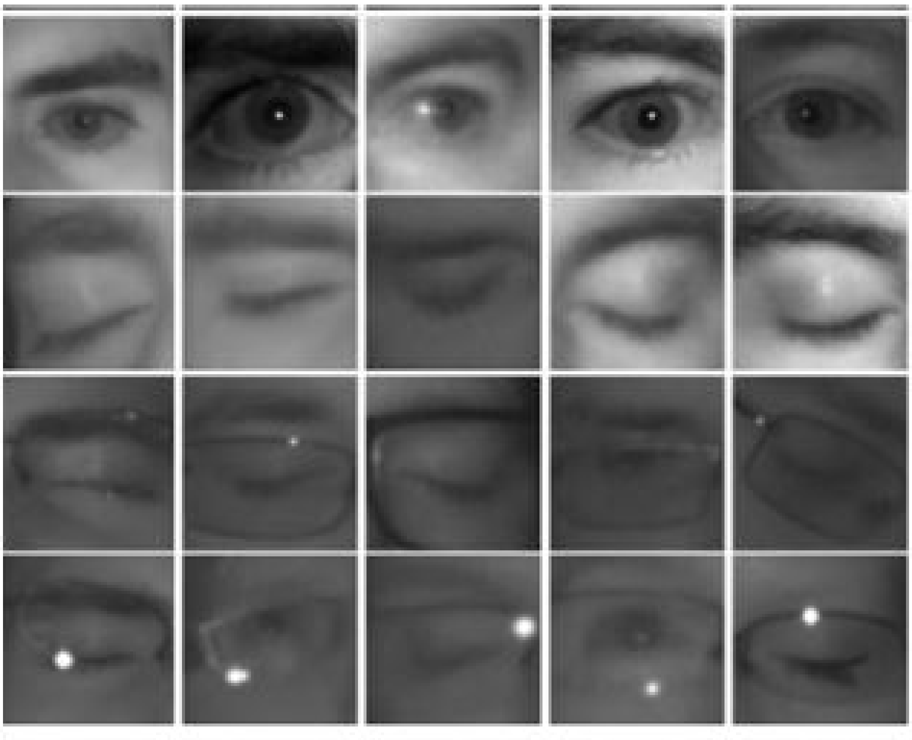


Figure 1. Media Research Lab (MRL) Eye dataset.

3.2. Data Preprocessing

The dimensions of 100 × 100 pixels were used to shrink 256 level-gray-color photos. The *Minmax* normalizer was used to normalize the image pixels; it converted every grayscale pixel in the range [0, 1] to lessen the results of the lighting variations.

3.3. Data Augmentation

For proper training, methods for classification based on deep learning require a large dataset. However, manually collecting such a large number of samples is quite challenging. Alternatively, by taking into account the diversity of the samples of some small or somewhat large datasets, the dataset size can be expanded. Random rotations, shifting, and zooming were used to enhance the photos. Figure 2 shows a diagram of the suggested methodology.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

where x' is the normalized value of the first intensity and x is the original intensity.

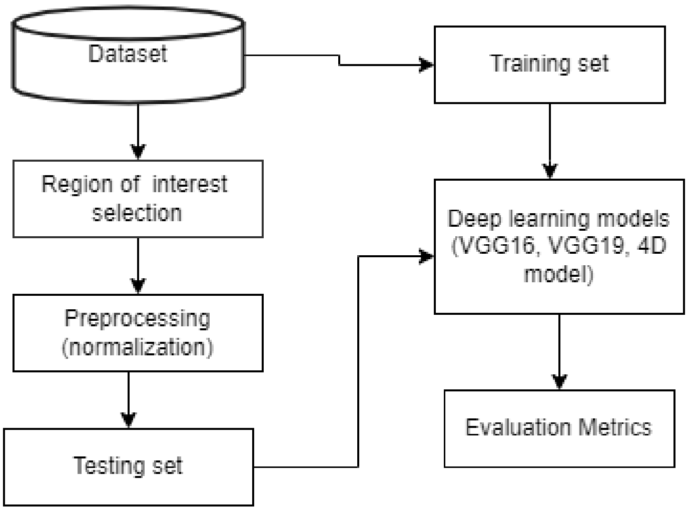

Dataset
Training set
Region of interest selection
Deep learning models (VGG16, VGG19, 4D model)
Preprocessing (normalization)
Evaluation Metrics
Testing set

Figure 2. Diagram of the proposed model.

3.4. Region of Interest Selection

In this step, our goal is to find and prepare the ocular region before feeding it into the network. Figures 3 and 4 illustrate the preferred structure for this section. When locating the eyes, we first identify the headbox. The Haar cascade classification algorithm was employed for head detection and for the facial landmark technique. The algorithm located optic landmarks, calculated the absolute distance between different points, as shown in Figure 3, and selected the greater distance. This tactic enhanced the technique's accuracy in detecting if an eye was closed.

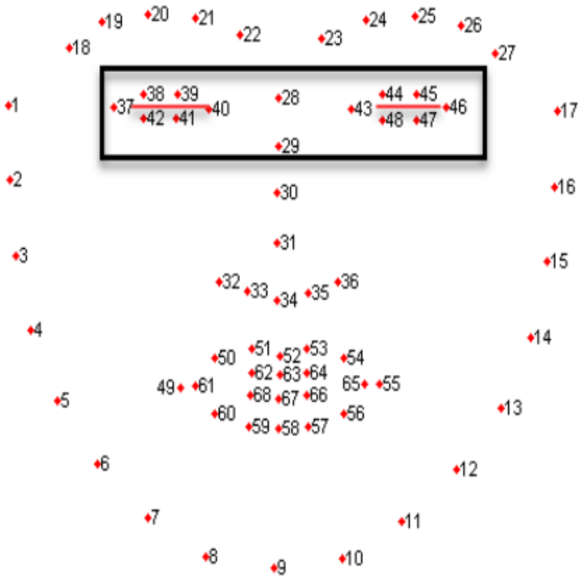


Figure 3. Facial extraction.

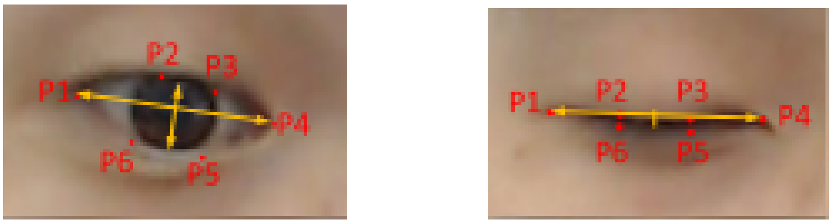


Figure 4. Eye extraction.

3.5. Eye Aspect Ratio Calculation

As previously mentioned, the state of a driver's eyes can indicate whether or not they are drowsy beecause there are considerable changes in the amount of time for which people who are awake or drowsy spend with their eyes closed. We noticed that the following facts might limit the performance: (1) The values of the pixels are sensitive. A changing environment easily harms image segmentation. (2) In practice, pixel values between pupils and glasses are very close, resulting in incorrect ellipse fitting. In this study, we used the Dlib toolkit [27] to provide a new, more stable parameter for evaluating the status of the driver's eyes. Using the Dlib toolbox, we were able to collect facial landmarks. As indicated in Figure 4, six dots were dispersed around each eye to locate its position. Between the open and closed states, the distribution of ocular landmarks differed significantly. The following formula can be used to calculate the EAR based on the position of eye landmarks:

$$\text{EAR} = \frac{||P_2 - P_6|| + ||P_3 - P_5||}{2||P_1 - P_4||} \tag{2}$$

From Equation (2), the coordinates of the eye landmarks are P_i, $i = 1, 2, \ldots, 6$. Whenever the driver is awake, the EAR is more than 0.2, as demonstrated in Figure 4. The EAR, on the other hand, is less than 0.2. This new parameter, the EAR, is substantially more robust because of the placement of solid facial landmarks.

3.6. Convolutional Neural Network

The concept of the CNN originated in [28], and it was was motivated by the brain's visual cortex and interpretation of visual data. Typically, a CNN's nonlinear and subsampling layers introduce other convolution layers. The network creates feature maps for the input image after extracting a massive amount of data from each pixel. Convolution layers are designed to extract features. The output layer of the final FC layer is the one that makes the ultimate decision. The convolution layer here was trained by using a back-propagation training method to take standout feature information from the input data. Figure 5 shows the CNN model architecture used here.

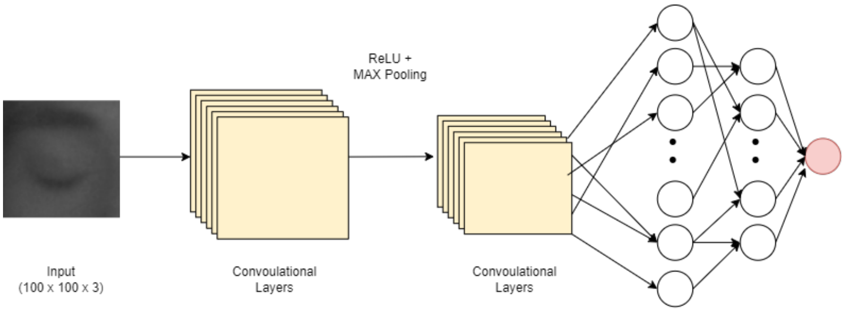


Figure 5. Convolutional neural network.

This paper suggests a custom CNN model called 4D as the most promising technique for detecting drowsiness based on eye state. Figure 6 illustrates the basic components of the proposed 4D model. The following is a quick rundown of each of the model's layers:

1. Convolution layer: The convolution operation was carried out with a stride of 1 by sliding a convolutional filter with a size of 33 and 5 across the input data matrix. We experimented with various kernels, ranging from 64 to 1024, and varying step sizes before settling on a combination of kernels that maximized the validation accuracy.
2. Activation function: A weighted total was calculated by the activation function, and then bias was added to it to decide whether or not to activate a neuron. We used the nonlinear activation function of ReLU [29], which converted negative elements into *0*, and it may be written as *Relu(x)* = *max(0,x)*, where *x* indicates a neuron's input.
3. Batch normalization: The batch normalization layer allowed each layer to increase its independent learning. It made the outputs of the previous layers more normal. Every

activation function followed the batch normalization layer. It aided in the speeding up of the learning process [30] and reduces the sensitivity to fluctuations in the input data. As a result, the neural network was stabilized. Batch normalization was also used to keep the data's distribution consistent.

4. Dropout: A model is considered to be over-fitted when it learns and performs poorly on the test dataset, but gives good accuracy on the training dataset. By randomly setting the activation to zero, the dropout layer is responsible for preventing over-fitting and improving performance [31].
5. Maxpooling layer: The process of choosing the largest element from a feature map is known as maxpooling. A feature map is produced from the output of the maxpooling layer.
6. Fully connected layer: When an input has been multiplied by a weight matrix, a bias vector is then produced in a fully connected layer. One or more fully connected layers are added after the convolution layers.
7. Output layer: Calculating the probability of each class occurring with a given input image is the responsibility of the output layer. With a sigmoid activation function, it produces a two-dimensional vector (3). The sigmoid activation function can be presented as follows:

$$\mathrm{S(x)} = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} = 1 - S(-x) \tag{3}$$

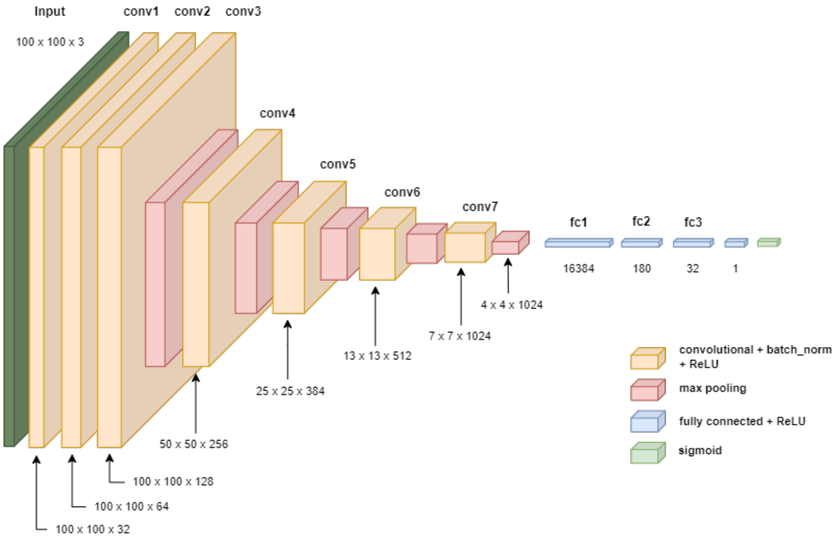


Figure 6. Proposed 4D model.

3.7. VGG16 and VGG19

In deep learning, instead of creating a unique CNN model for the image classification task, we often utilize a transfer learning strategy in which a CNN model that was previously trained on a sizable benchmark dataset, such as "ImageNet", is reused. Rather than beginning the learning process from zero, transfer learning builds on past knowledge. We used a pre-trained CNN and transfer learning theory to extract characteristics. VGG16 and VGG19 were chosen as the suggested pre-trained networks for this purpose. VGG16 is a deep CNN network developed by Simonyan and Zisserman [14]. The ImageNet dataset, which includes many images and has 1000 classes, was used to train the 16-layer VGG16; VGG19, which has 19 layers, is a more complex version of VGG16. It was also trained on the ImageNet database. However, the network could also imply different sizes. It was built for images with dimensions of 224 × 224 pixels. These networks extracted high-level characteristics by using the three most recent fully connected layers and low-level training features with the weights in the ImageNet dataset. Figures 7 and 8 show the architecture of the VGG16 and VGG19 models used.

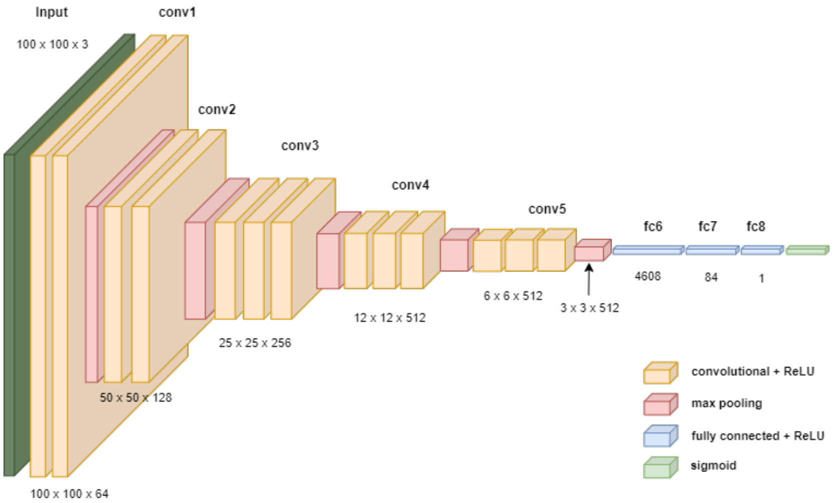


Figure 7. Architecture of the VGG16 model.

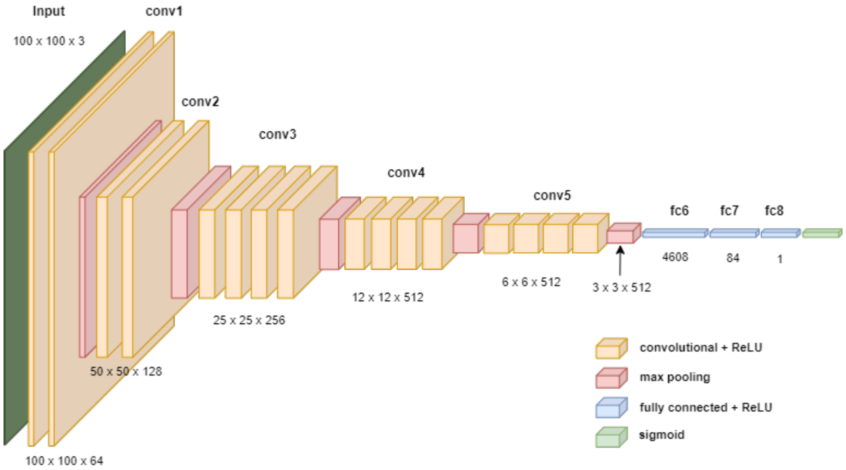


Figure 8. Architecture of the VGG19 model.

4. Results and Discussion

Here, we conducted two distinct sorts of experiments. In the first category, an experiment was conducted by using a recorded dataset. The second form of the experiment was conducted via video. We created a dataset containing 47,000 images to conduct the first type of experiment.

4.1. Accuracy Evaluation

The MRL Eye dataset was used to evaluate the model; it contained static images of eyes in various lighting conditions. In the training phase, the eyes were divided into two groups (open and closed). The MRL dataset was used to train three networks (VGG16, VGG19, and the 4D model). As an optimizer and loss function, *rmsprop* and binary cross-entropy were chosen. The learning rate was set to *0.001*, which was used along with a scheduler. The learning rate dropped if the validation accuracy did not improve after three epochs. A total of 75% of the data were used for training, while the rest were used for validation. We will go over the performance of the model on both subsets and assess it by using commonly employed evaluation metrics, such as the accuracy, precision, recall, and confusion matrix of the classification. The accuracy of the networks on the MRL Eye dataset, as shown in Table 1, indicated that VGG16, VGG19, and the 4D model gave 95.93%, 95.03%, and 97.53% accuracy on ROI images, respectively. A comparison of the training and testing times is shown in Table 2. We were able to determine the total amount of calculation that the model would need to carry out in order to estimate the inference time for that model. Flops—floating-point operations per second—and MACCs were used. Here, the 4D model showed a balanced result in comparison with the other two models, as shown in Table 3.

The accuracy, precision, and recall curves of all three models are presented in Figures 9–11, respectively. Figure 9 presents the classification accuracy at each epoch of the proposed model and the pre-trained models. The experiment was carried out for 50 epochs. Figure 9a shows the classification accuracy in the case of the VGG16 model. On the testing subset, the classification accuracy at the last epoch was 93.87%; Figure 9b shows the accuracy of the VGG19 model, which was 95.47% at the last epoch. The 4D model provided an excellent curve in both the training and testing phases, and it achieved an accuracy of 97.53% at the last epoch (Figure 9c). Figure 10 shows the precision curves for each of the models. A classification's positive predictive value (PPV), which is the ratio of the samples that are positively recognized to all of the samples that are accurately identified as belonging to a given class, is determined by precision. The experiment was carried out for 50 epochs. Figure 9a shows the classification accuracy in the case of the VGG16 model. On the testing subset, the classification accuracy at the last epoch was 93.87%; Figure 9b shows the accuracy of the VGG19 model, which was 95.47% at the last epoch. The 4D model provided an excellent curve in both the training and testing phases, and it achieved an accuracy of 97.53% at the last epoch (Figure 9c). Figure 10 shows the precision curves for each of the models. A classification's positive predictive value (PPV), which is the ratio of the samples that are positively recognized to all of the samples that are accurately identified as belonging to a given class, is determined by precision. Here, the 4D model also provided better precision values (97.35%) than those of the other two pre-trained models, and it provided better recall values, as depicted in Figure 11c. The confusion matrices for VGG16, VGG19, and the proposed 4D model are shown in Figure 12. The number of falsely predicted values was only 62, and the number of accurately predicted values was 6279 in the 4D model. A performance comparison between the different approaches on the MRL Eye dataset is shown in Table 4. It is worth mentioning that our approaches achieved the highest accuracy, outperforming the other approaches by a good margin.

Table 1. Accuracy evaluation.

Network Model	Accuracy	Precision	Recall
VGG16	95.93%	93.15%	93.87%
VGG19	95.03%	94.82%	95.47%
4D model	**97.53%**	**97.35%**	**97.06%**

Table 2. Comparison of the training and testing times.

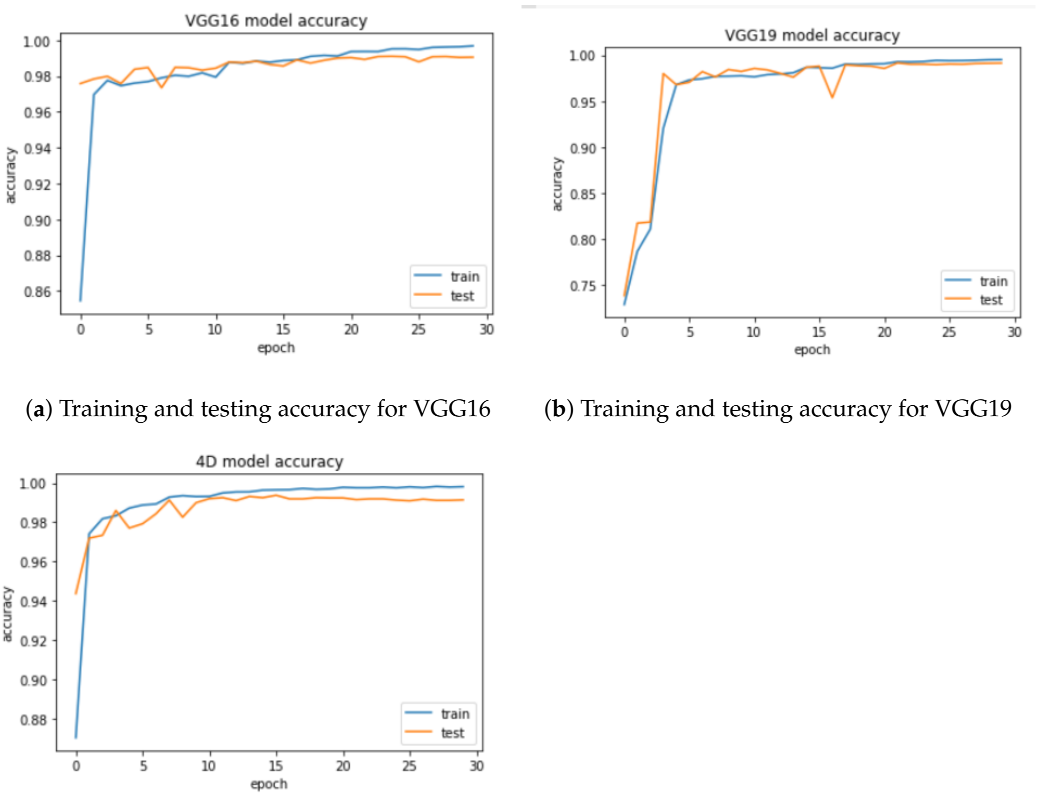
Network Model	Training Time	Prediction Time
VGG16	1036.07 s	16.17 s
VGG19	1144.900 s	19.69 s
4D model	**1205.379 s**	**19.35 s**

Table 3. Flops and MACCs.

Network Model	Total Flops (10^9)	Total MACCs (10^9)
VGG16	5.93124	2.9650
VGG19	7.51787	3.7583
4D model	**6.19578**	**3.0967**

Table 4. Comparison of the proposed networks' accuracy with that of various approaches when using the MRL Eye dataset.

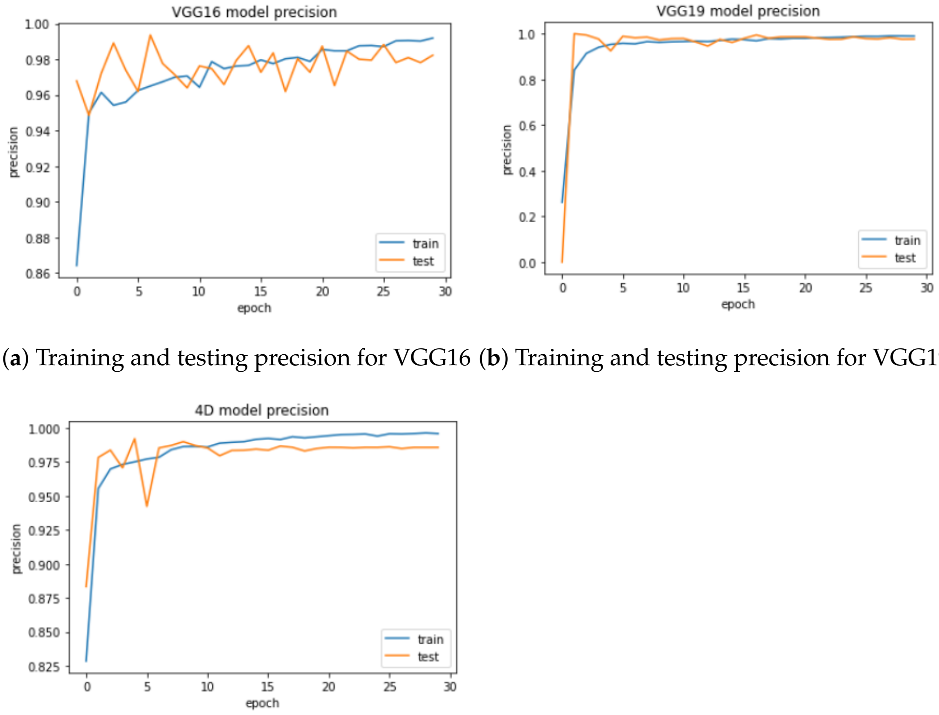
Research	Method	Accuracy
W. Kongcharoen [32]	Haar cascade+CNN	94%
Y. Suresh [33]	CNN	86.05%
M.E. Walizad [34]	CNN	95%
M. Tibrew [35]	TEDD+CNN	95%
Ours	VGG16	95.93%
Ours	VGG19	95.03%
Ours	**4D model**	**97.53%**



(**a**) Training and testing accuracy for VGG16 (**b**) Training and testing accuracy for VGG19

(**c**) Training and testing accuracy for 4D

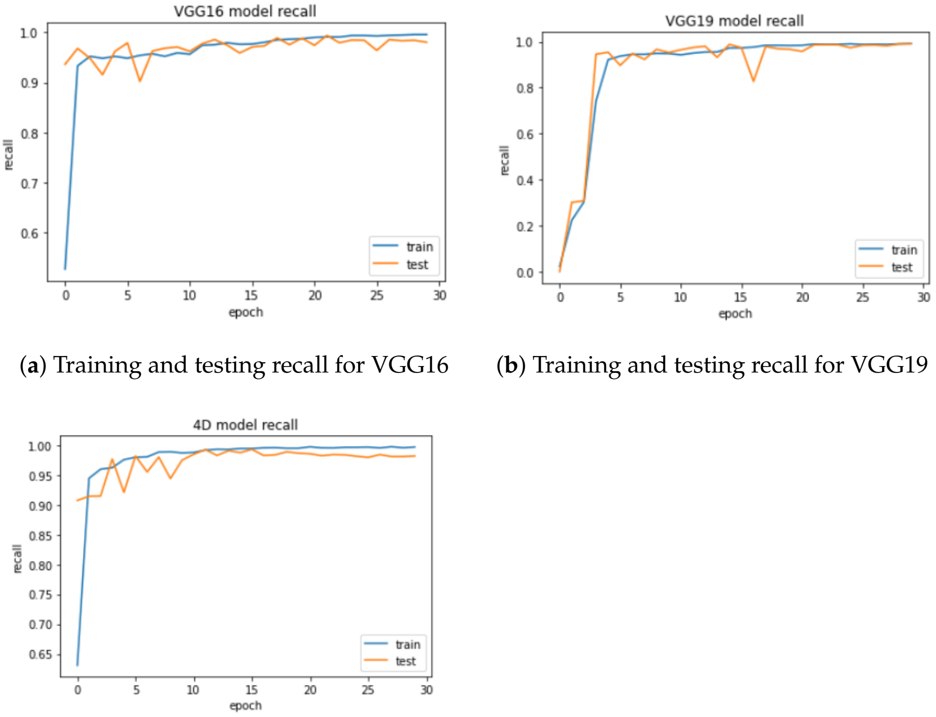
Figure 9. Training and testing accuracy curves for different models.



(**a**) Training and testing precision for VGG16 (**b**) Training and testing precision for VGG19

(**c**) Training and testing precision for 4D

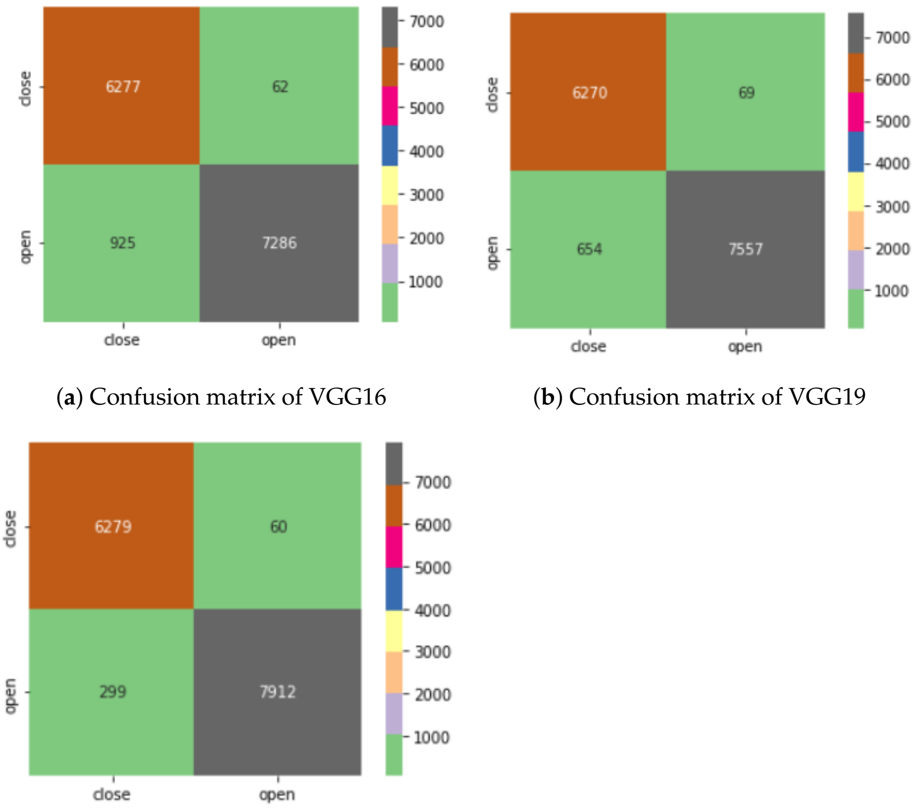
Figure 10. Training and testing precision curves for different models.



(**a**) Training and testing recall for VGG16 (**b**) Training and testing recall for VGG19

(**c**) Training and testing recall for 4D

Figure 11. Training and testing recall curves for different models.



(**a**) Confusion matrix of VGG16

(**b**) Confusion matrix of VGG19

(**c**) Confusion matrix of 4D Model

Figure 12. Confusion matrices of different models.

4.2. Model Learning Visualization

Model learning visualization is implemented by using a class activation map (CAM). This permits the scientist to examine the image to be classified and identify the elements/pixels in the image that had the most impact on the model's (4D model) output. It accomplishes the job by designing a heatmap that highlights pixels in the input image that influence the categorization of the image. A blue color indicates discriminative image regions associated with the model (4D)-predicted class. The second column shows the picture from our MRL Eye dataset, and the third column represents the prediction area of the image.

Nowadays, neural networks may make such complicated decisions that evaluating a model based on accuracy is no longer sufficient. It is also crucial to make decisions based on the suitable region. The model implemented in this paper was accurate and provided results from the correct region. Figure 13 shows the activation map for open and closed eyes.

4.3. Real-Time Implementation of Drowsiness Detection

Throughout the testing stage, by using the OpenCV library, a video stream from a camera was used as input. Facial landmark detection was carried out on each sampled video frame by utilizing Dlib's API; the trained model was used as input by separating the left and right eyes from the isolated face. The number of frames in which closed eyes were observed over a specific period of time was referred to as eye closure. When the eye went from being open to being closed, blinking was seen. The driver would be regarded as sleepy if their eyes blinked less than the threshold value (2 s). This method was user-friendly and did not need any special hardware other than a webcam. This made the system suitable to be implemented on a desktop computer, mobile device, and so on. Figure 14 depicts an experimental flow diagram. Figure 15 shows the results of the second type of experiment. The real-time implementation consisted of five steps:

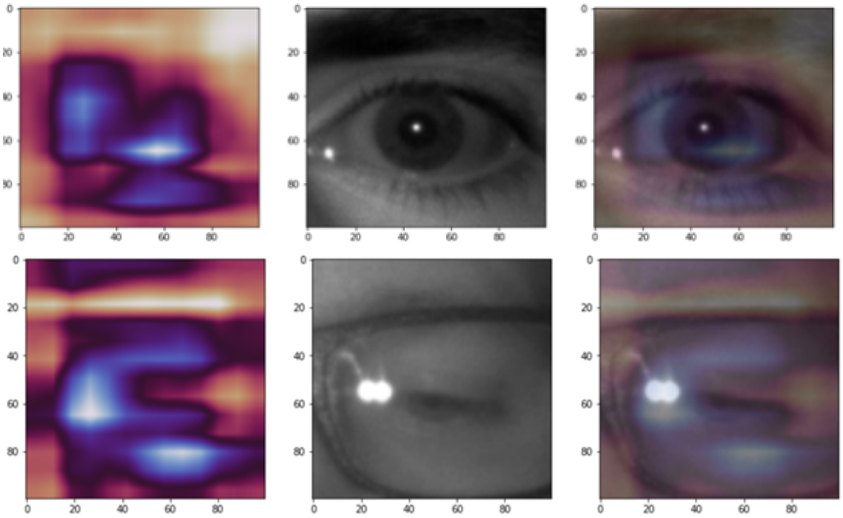


Figure 13. Activation map for open and closed eyes.

Step 1—Extracting videos: We take an image from a camera as input and read it using OpenCV.

Step 2—Extracting images from video frames: We extract every video frame of the proposed model as an image at a rate of 30 frames per second while picking films from the collection.

Step 3—Landmark coordination from image extraction: This uses the Dlib18 library in the third phase to derive landmark coherence from images.

Step 4—Training the algorithm: Here, a training process is conducted. Numerous predictions from which a model will be formed are made; if the predictions are incorrect, the model will be corrected. The training is carried out until it achieves the desired degree of accuracy.

Step 5—Model extraction: Finally, based on the rate of eye blinking, the algorithms determine whether or not the driver is drowsy.

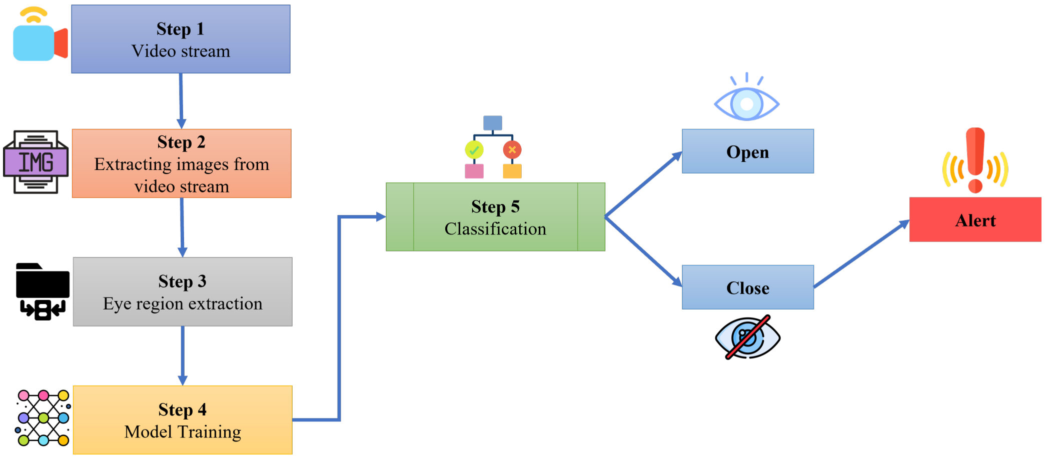


Figure 14. Flow diagram of the real-time implementation.

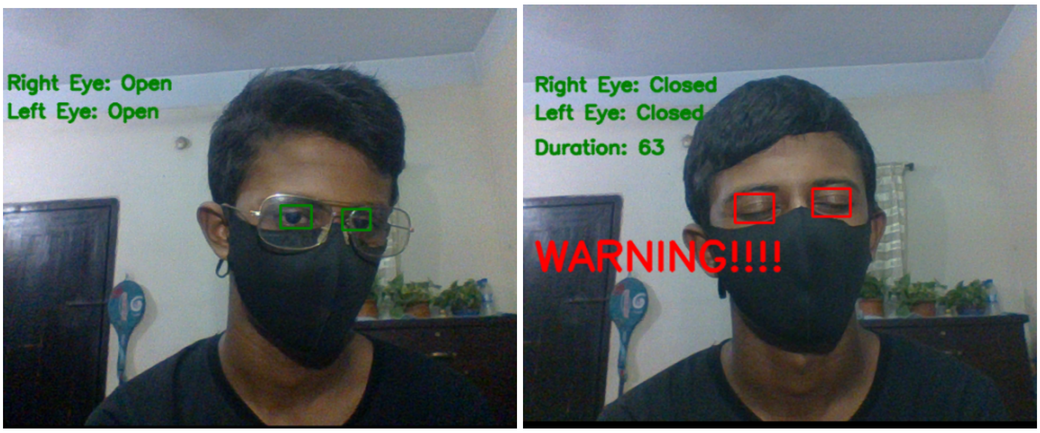


Figure 15. Real-time implementation of drowsiness detection (with and without glasses).

5. Conclusions

The study presented a deep-learning-based drowsiness detection system for classifying eye states in order to detect drowsiness. The major goal was to create a system that was lightweight enough to be implemented in embedded systems and still achieve good performance. In the initial stage, the system recorded a stream of frames, then picked the eye area by preprocessing frames, and selected the camera's eye for the following stage. Finally, the image was reduced to 100 × 100 pixels in size. The authors fed the result of this stage into the network to classify eye states, and based on the eye state, drowsiness is detected. The accomplishment, in this case, was the creation of a deep learning model that was compact but had a high level of accuracy in all three networks, namely, VGG16 (95.93%), VGG19 (95.03%), and the 4D model (97.50%). A comparison with similar drowsiness detection methods revealed that the proposed method showed superior performance to most of theirs.

6. Future Work

This research's future goals include improving this single-process sleepiness detection method by using several threads in which the processes are shared and several processes are executed at the same time. Simultaneously running or executing processes can improve performance by reducing the time that it takes to complete them. This also improves the responsiveness of the user interface. By using a nano camera, this device can also monitor the rays reflected from the eye; the disappearance of reflected rays can be equated to the closing of the eyes. These additions may be able to improve the drowsiness detection system. To improve the robustness of drowsiness detection, the head position can also be included as a component. The following consideration can also be included for future implementations: There can also be devices that monitor a patient's heart rate to determine if they are qualified to operate a vehicle. There could be a major discrepancy between the driving-behavior-based measurements used in real-world driving and those used in simulations.

Author Contributions: Conceptualization, I.J. and K.M.A.U.; methodology, S.A.M., M.S.U.M. and T.Z.K.; software, S.A.M. and A.K.B.; validation, M.M., and K.M.A.U.; formal analysis, S.A. and I.J.; investigation, I.J. and S.A.M.; resources, M.S.U.M. and T.Z.K.; data curation, S.A.M.; writing—original draft preparation, I.J., K.M.A.U. and S.A.M.; writing—review and editing, M.M., S.A. and A.K.B.; visualization, I.J.; supervision, K.M.A.U. and S.A.M.; project administration, K.M.A.U.; funding acquisition, M.M. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: Taif University Researchers Supporting Project Number (TURSP-2020/73), Taif University, Taif, Saudi Arabia.

Data Availability Statement: The data is collected from a publicly available repository named MRL Eye Dataset.

Acknowledgments: The authors would like to thank the Taif University Researchers Supporting Project Number (TURSP-2020/73), Taif University, Taif, Saudi Arabia.

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

References

1. 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 ICIoT 2020, Doha, Qatar, 2–5 February 2020; pp. 237–242. [CrossRef]
2. Ahsan, M.M.; Li, Y.; Zhang, J.; Ahad, M.T.; Yazdan, M.M.S. Face recognition in an unconstrained and real-time environment using novel BMC-LBPH methods incorporates with DJI vision sensor. *J. Sens. Actuator Netw.* **2020**, *9*, 54. [CrossRef]
3. Ramzan, M.; Khan, H.U.; Awan, S.M.; Ismail, A.; Ilyas, M.; Mahmood, A. A survey on state-of-the-art drowsiness detection techniques. *IEEE Access* **2019**, *7*, 61904–61919. [CrossRef]
4. Abtahi, S.; Omidyeganeh, M.; Shirmohammadi, S.; Hariri, B. YawDD: A yawning detection dataset. In Proceedings of the 5th ACM Multimedia Systems Conference, Singapore, 19–21 March 2014; pp. 24–28.
5. Weng, C.H.; Lai, Y.H.; Lai, S.H. Driver drowsiness detection via a hierarchical temporal deep belief network. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 117–133.
6. Revelo, A.; Álvarez, R.; Grijalva, F. Human drowsiness detection in real time, using computer vision. In Proceedings of the 2019 IEEE Fourth Ecuador Technical Chapters Meeting (ETCM), Guayaquil, Ecuador, 13–15 November 2019; pp. 1–6.
7. Adhikary, A.; Murad, S.A.; Munir, M.S.; Hong, C.S. Edge Assisted Crime Prediction and Evaluation Framework for Machine Learning Algorithms. In Proceedings of the 2022 International Conference on Information Networking (ICOIN), Jeju-si, Republic of Korea, 12–15 January 2022; pp. 417–422.
8. Fan, X.; Yin, B.; Sun, Y. Yawning detection based on gabor wavelets and LDA. *J. Beijing Univ. Technol.* **2009**, *35*, 409–413.
9. Zhang, Z.; Zhang, J. A new real-time eye tracking based on nonlinear unscented Kalman filter for monitoring driver fatigue. *J. Control Theory Appl.* **2010**, *8*, 181–188. [CrossRef]
10. Philip, P.; Sagaspe, P.; Moore, N.; Taillard, J.; Charles, A.; Guilleminault, C.; Bioulac, B. Fatigue, sleep restriction and driving performance. *Accid. Anal. Prev.* **2005**, *37*, 473–478. [CrossRef] [PubMed]
11. Tremaine, R.; Dorrian, J.; Lack, L.; Lovato, N.; Ferguson, S.; Zhou, X.; Roach, G. The relationship between subjective and objective sleepiness and performance during a simulated night-shift with a nap countermeasure. *Appl. Ergon.* **2010**, *42*, 52–61. [CrossRef] [PubMed]
12. Savaş, B.K.; Becerikli, Y. Real time driver fatigue detection based on SVM algorithm. In Proceedings of the 2018 6th International Conference on Control Engineering & Information Technology (CEIT), Istanbul, Turkey, 25–27 October 2018; pp. 1–4.
13. Jalilifard, A.; Pizzolato, E.B. An efficient K-NN approach for automatic drowsiness detection using single-channel EEG recording. In Proceedings of the 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, USA, 16–20 August 2016; pp. 820–824.
14. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
15. Awais, M.; Badruddin, N.; Drieberg, M. A hybrid approach to detect driver drowsiness utilizing physiological signals to improve system performance and wearability. *Sensors* **2017**, *17*, 1991. [CrossRef] [PubMed]
16. 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]
17. Bhandarkar, S.; Naxane, T.; Shrungare, S.; Rajhance, S. Neural Network Based Detection of Driver's Drowsiness. *TechRxiv* **2021**. [CrossRef]
18. Khushaba, R.N.; Kodagoda, S.; Lal, S.; Dissanayake, G. Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm. *IEEE Trans. Biomed. Eng.* **2010**, *58*, 121–131. [CrossRef] [PubMed]
19. Mardi, Z.; Ashtiani, S.N.M.; Mikaili, M. EEG-based drowsiness detection for safe driving using chaotic features and statistical tests. *J. Med. Signals Sens.* **2011**, *1*, 130. [PubMed]
20. Noori, S.M.R.; Mikaeili, M. Driving drowsiness detection using fusion of electroencephalography, electrooculography, and driving quality signals. *J. Med. Signals Sens.* **2016**, *6*, 39. [PubMed]
21. Picot, A.; Charbonnier, S.; Caplier, A. On-line detection of drowsiness using brain and visual information. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2011**, *42*, 764–775. [CrossRef]
22. Krajewski, J.; Sommer, D.; Trutschel, U.; Edwards, D.; Golz, M. Steering wheel behavior based estimation of fatigue. In Proceedings of the Driving Assesment Conference, Big Sky, MT, USA, 22–25 June 2009; Volume 5.
23. Mandal, B.; Li, L.; Wang, G.S.; Lin, J. Towards detection of bus driver fatigue based on robust visual analysis of eye state. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 545–557. [CrossRef]
24. Yang, C.; Yang, Z.; Li, W.; See, J. FatigueView: A Multi-Camera Video Dataset for Vision-Based Drowsiness Detection. *IEEE Trans. Intell. Transp. Syst.* **2022**. [CrossRef]

25. Islam, M.S.; Hasan, M.M.; Abdullah, S.; Akbar, J.U.M.; Arafat, N.; Murad, S.A. A deep Spatio-temporal network for vision-based sexual harassment detection. In Proceedings of the 2021 Emerging Technology in Computing, Communication and Electronics (ETCCE), Dhaka, Bangladesh, 21–23 December 2021; pp. 1–6.
26. Fusek, R. Pupil localization using geodesic distance. In Proceedings of the International Symposium on Visual Computing, Las Vegas, NV, USA, 19–21 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 433–444.
27. King, D.E. Dlib-ml: A machine learning toolkit. *J. Mach. Learn. Res.* **2009**, *10*, 1755–1758.
28. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
29. Zheng, H.; Yang, Z.; Liu, W.; Liang, J.; Li, Y. Improving deep neural networks using softplus units. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–4.
30. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning (PMLR), Lille, France, 6–11 July 2015; pp. 448–456.
31. Park, S.; Kwak, N. Analysis on the dropout effect in convolutional neural networks. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 189–204.
32. Kongcharoen, W.; Nuchitprasitchai, S.; Nilsiam, Y.; Pearce, J.M. Real-Time Eye State Detection System for Driver Drowsiness Using Convolutional Neural Network. In Proceedings of the 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, Thailand, 24–27 June 2020; pp. 551–554.
33. Suresh, Y.; Khandelwal, R.; Nikitha, M.; Fayaz, M.; Soudhri, V. Driver Drowsiness Detection using Deep Learning. In Proceedings of the 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 20–22 October 2021; pp. 1526–1531.
34. Walizad, M.E.; Hurroo, M.; Sethia, D. Driver Drowsiness Detection System using Convolutional Neural Network. In Proceedings of the 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 28–30 April 2022; pp. 1073–1080.
35. Tibrewal, M.; Srivastava, A.; Kayalvizhi, R. A deep learning approach to detect driver drowsiness. *Int. J. Eng. Res. Technol.* **2021**, *10*, 183–189.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.