Internet of Things 22 (2023) 100705

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

Driver drowsiness detection and smart alerting using deep learning and IoT

Anh-Cang Phan [a,*], Thanh-Ngoan Trieu [b,c], Thuong-Cang Phan [c]

[a] Vinh Long University of Technology Education, Vinh Long, 85110, Viet Nam
[b] Université de Bretagne Occidentale, Brest, 29200, France
[c] Can Tho University, Can Tho, 94115, Viet Nam

ARTICLE INFO

Keywords:
Drowsiness detection
LSTM
DenseNet
VGG-16
Inception-V3

ABSTRACT

Drowsiness is a common problem that many drivers encounter due to long working hours, lack of sleep, and tiredness. Tired drivers are as dangerous as drunk drivers because they have slower reaction times and suffer from reduced attention, awareness, and ability to control their vehicles. Drowsy driving causes many traffic accidents, especially fatal crashes. Therefore, the best way to prevent accidents involving drowsiness is to alert the drivers ahead of time. The accuracy of the drowsiness prediction reduces if the studies only focus on facial landmarks, ignoring other fatigue features such as tilting head, blinking, and yawning. To solve these problems, we propose an approach to detect driver drowsiness efficiently and accurately using IoT and deep neural networks improved from LSTM, VGG16, InceptionV3, and DenseNet. The use of transfer learning technique combined with multiple drowsiness signs is to improve the accuracy of the drowsiness detection in various driving conditions. The time-varying factor is also taken into consideration in the models developed from LSTM and DenseNet. When the driver's fatigue is detected, the IoT module emits a warning message along with a sound through a Jetson Nano monitoring system. The experimental results demonstrate that our approach using deep neural networks can achieve high accuracy of up to 98%. Notably, this approach has also been verified in cases with/without wearing a mask and glasses. This has a practical meaning in the Covid-19 pandemic situation when everyone needs to comply with the wearing of masks in public places.

1. Introduction

Feeling abnormally sleepy or tired during the day is commonly known as drowsiness. Drowsiness may lead to additional symptoms, such as forgetfulness or falling asleep at inappropriate times. This is a natural phenomenon in the human body that causes distraction and affects the lives of road users. According to statistics from the US National Highway Traffic Safety Administration, 50,000 injuries and nearly 800 deaths have been reported with 91,000 traffic accidents related to drowsiness [1]. According to the National Sleep Foundation, in 2005, 60% of drivers committed drowsy driving in the previous year [2] and an estimated of 6,400 people died annually in crashes involving drowsy driving [3]. The Foundation for Traffic Safety reported that 21% of all fatal crashes involved a drowsy driver from 2009 to 2013 [4]. About 1/25 drivers admitted that they were drowsy driving in the last 30 days according to the Centers for Disease Control and Prevention [5]. In the first quarter of 2021, there were approximately 8,730 car accident fatalities in the United States [6]. The above alarming statistics have shown the necessity to implement a system for driver drowsiness monitoring and alerting, thereby preventing unfortunate traffic accidents from happening. Recently, many models

* Corresponding author.
E-mail addresses: cangpa@vlute.edu.vn (A.C. Phan), ngoan.trieuthanh@etudiant.univ-brest.fr (T.N. Trieu), ptcang@cit.ctu.edu.vn (T.C. Phan).

https://doi.org/10.1016/j.iot.2023.100705
Received 2 October 2022; Received in revised form 22 January 2023; Accepted 22 January 2023
Available online 14 February 2023

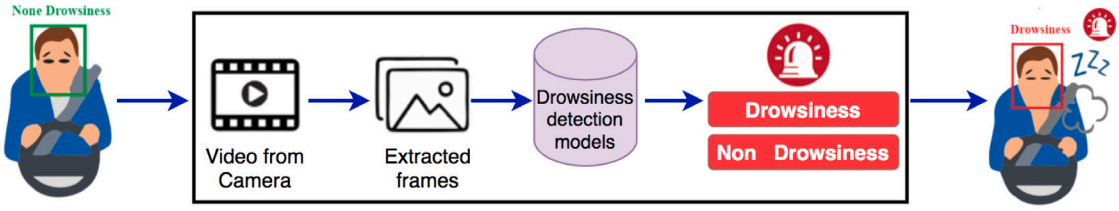


Fig. 1. General model of the drowsiness detection system.

have been developed for automatic drowsiness detection systems. The inputs to the systems are images obtained from a camera that will be used in detection models to conclude whether the driver is asleep. The general model of the system is shown in Fig. 1.

In our previous work [7], we implemented two methods of drowsiness detection with three scenarios. The first method applied facial landmarks to detect blinks and yawns based on the respective thresholds for each driver. The second method used deep learning techniques with two neural networks developed based on MobileNet-V2 and ResNet-50V2. The results showed that the proposed method using deep learning techniques achieved high accuracy of 97%. In this work, we continue to improve the doze detection methods using deep learning networks developed on LSTM, VGG-16, Inception-V3, and DenseNet. These networks are good feature extraction tools thus they can learn the relevant features of dozing states (tilting head, face area, eyes, and mouth). The experimental results show that the detection accuracy can be up to 98%. In this paper, we make an extensive study with the following contributions: (1) Propose drowsiness detection models based on several deep learning networks and combine IoT techniques using devices such as Jetson Nano and Camera to create a real-life detection system. We have successfully implemented the driver drowsiness detection experiment with IoT and machine learning techniques at center of Toyota Technical Education Program, Vinh Long University of Technology and Education, to serve as a basis for further research and confirm the feasibility of the method in practical use. (2) Expand the previously built experimental dataset by collecting drowsy driver videos taken from YouTube, Kaggle, and iStock. The dataset contained 21,542 facial images with 3425 images of the drowsy state, 4388 images of the non-drowsy state, and 13,729 images of both states. The data context includes wearing a mask, wearing glasses, yawning, tilting heads, not looking directly at the camera, and head down. (3) Based on the results of our previous work, we propose the doze detection models developed based on LSTM, VGG16, Inception-V3, and DenseNet, which are good CNNs widely used in object classification and detection studies. There is an incorporation of time-varying signs of drowsiness in the model improved from LSTM during the training process. This study presents, compares, and contrasts various algorithms of deep learning to find the most promising approach that can be used for the detection of driver fatigue and drowsiness. (4) Take advantage of the transfer learning approach and the advantages of proposed neural networks, we improve the models' accuracy by combining many different factors such as the state of the eyes, the mouth, and the tilt of the head. The emotion classification model is integrated into the system to remove frames containing emotions that are not related to drowsiness to shorten the feature extraction and detection processing time. (5) The proposed system detects correctly in cases of wearing glasses and masks. In addition, we determine the level of drowsiness to provide early warning thus drivers can avoid unintended accidents.

The research has established the ability to detect drowsiness with a combination of deep learning and IoT. We studied drowsy driving in a high-fidelity driving simulator and evaluated the ability of an automotive production-ready driver monitoring system to detect drowsy driving. The primary objective of this study is to build and evaluate predictive models for drowsiness events. The models were effective at discriminating low levels of drowsiness from moderate to severe drowsiness. The remaining of the paper is presented as follows. Section 2 presents the literature review on drowsiness detection with deep learning networks. Details of the methodology are described in Section 3. We provide some experimental results for the training and test process of the proposed methods in Section 4. A comparison and discussion between the proposed methods and the recent methods are made in Section 5. After the evaluation, we choose a suitable model to implement an IoT-based drowsiness detection system presented in Section 6. Finally, we draw a conclusion in Section 7.

2. Related works

Driver drowsiness represents an important cause of accidents or near-missed accidents proven by a number of studies establishing links between driver drowsiness and traffic accidents [8–10]. Drowsiness detection is different from other road safety problems that can be detected by measuring the content in the driver's body. Drowsiness detection is being researched using a range of approaches including physical and physiological methods [11,12]. The physical techniques detect drowsy states by features such as eyes state (closed or opened), eye blinking rate, yawning, and head movement. The input for this detection comes from a video camera capturing driver images. One can argue that video camera methods have difficulties when the environmental light conditions are highly variable and when drivers wear glasses/sunglasses. The physiological methods monitor the electroencephalogram (EEG), the electrooculogram (EOG), and the electromyogram (EMG) signals to evaluate the degree of driver drowsiness. However, drivers need to be attached to electronic devices causing uncomfortable and inappropriate for monitoring drivers regularly.

In the physical-based methods, there are two popular approaches used to detect drowsiness, i.e., using facial landmarks and machine learning techniques. The methods based on facial landmarks usually calculate the Eye Aspect Ratio (EAR) of the eye area

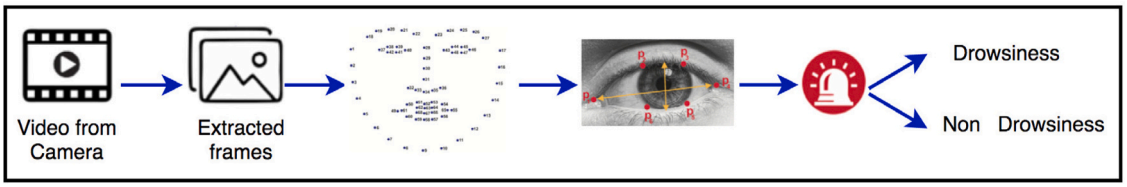


Fig. 2. Model of the drowsiness detection system based on facial landmarks.

and predict drowsy driving (Fig. 2). Wong and Lau [13] proposed a system integrated into the real-time device for driver drowsiness detection. This study calculated EAR with a fixed eye-opening threshold for detecting drowsy drivers, which limited the recognition of people whose eyes are smaller than the threshold. Ramos et al. [14] came up with an idea using blink and yaw recognition based on facial landmarks to detect drowsy driving. This study is similar to [13] in using a fixed eye-opening threshold. Shivani et al. [15] proposed a method using Haar Cascade classifier together with OpenCV library for real-time video monitoring of drivers. The EAR was calculated based on detecting the driver's eyes, thereby alarming occur when the driver is dozing. Biswal et al. [16] built a system using the Raspberry Pi3 camera in combination with facial landmarks for determining the blink rate. Similarly, this study also focused only on the eyes and ignored other drowsiness factors. In general, the above studies provided complete dozy driving detection systems. However, most studies just focused on the eyes area, ignored other drowsiness factors, and used a fixed EAR threshold leading to inaccurate predictions. This makes eye tracking based drowsiness detection difficult to implement as a real system.

Machine learning has been applied in many fields of study with outstanding advantages such as high accuracy, application on different types of datasets, supporting small datasets, and scalability in terms of data and computation [17]. Unlike the methods of using facial landmarks, the methods applying deep learning techniques will be conducted based on two phases, the training phase and the test phase (Fig. 3). He et al. [18] proposed a dozy driving detection method using a convolutional neural network (CNN) to perform feature extraction and classification for eyes and mouth state recognition. Zhao et al. [19] proposed a convolutional neural network namely EM-CNN for fatigue driving detection focusing on the eyes and mouth states. Venkata and Suchismitha [20] introduced the use of wavelet packet transform to detect drowsy driving with an accuracy of 94.45%. Wavelet packet transform is extracted from single channel electroencephalogram signals. Chand and Karthikeyan [21] proposed a multilevel distribution model for detecting doze based on CNN using emotion analysis. The facial features of drivers are processed by CNN to detect the drivers' behavior and emotions. The experimental results proved the effectiveness of the proposed model with an accuracy of up to 93%. Ajinkya et al. [22] used DNN based on Haar feature-based cascade classifier for mouth and eyes region detection. In addition, time series data have been introduced to detect drowsy drivers in a number of studies. Azhar Quddus et al. [23] proposed a method using long short term memory (LSTM) and CNNs for driver drowsiness detection. There were two types of LSTMs employed consisting of the R-LSTM and convolutional LSTM. They performed a power spectral analysis of multichannel electroencephalogram signals and an array of LSTM cells to model eye movements. Their approach resulted in an accuracy in the range of 82%–97%. This study only dealt with the states of the left and right eye areas without considering other sleepiness features. It also did not address specific experimental cases of wearing a mask, glasses, or a combination of both. Vishnu Yarlagadda and colleagues [24] introduced driver drowsiness detection using RNNs with LSTM to handle the sequential multimedia data. Facial parameters related to eye and mouth organs had also been extracted to estimate the drowsiness level of a driver. This approach obtained an accuracy of 97.25%. However, the problem of vanishing gradient and over-fitting had not been eliminated yet. Faraji et al. [25] applied the LSTM network to learn driver temporal behaviors including yawning and blinking time periods as well as sequence classification. YOLOv3 was employed to extract facial features automatically. The accuracy of 91.7% was a result of their approach. In this study, the authors presented the experimental process for two cases of wearing and not wearing glasses but did not consider some cases of wearing a mask, or a combination of both mask and glasses.

In the physiological methods, ECG records the electrical activity of the driver heart whereas the EEG records the electrical activity of the driver's brain. When drivers start to go into a drowsy state, the heartbeat is slowing down and there are changes in the alpha and theta waves of the brain signals showing a slowdown of brain activity. Chaabene et al. [26] proposed an EEG classification system for drowsiness detection based on a simple CNN model. The results showed that the accuracy of the system reached 90.14% in distinguishing between two states of drowsy and awake. The proposed system mainly focused on data acquisition and model analysis without any actual empirical analysis of different cases of wearing glasses and masks. The authors also did not consider validating this system on large datasets collected under real driving conditions. To improve the accuracy of dozy detection, Geoffroy et al. [27] introduced drowsiness detection using joint EEG and ECG signals with convolutional neural networks and recurrent neural networks. This method was obtained with accuracy scores up to 97% on a validation set. A short-time Fourier transform (STFT) was applied to each frame signal for temporal analysis. The STFT is widely used in signal feature extraction for time–frequency decomposition. However, it is not an effective solution to provide a frequency resolution that varies with the temporal resolution since it has a fixed time–frequency window. This makes it inaccurate to analyze signals that change rapidly with time. Meanwhile, drowsiness is considered to be a state that occurs after a progressive attention decrease, as a consequence, EEG and ECG signals are able to vary suddenly with time and contain non-periodic and fast transient features.

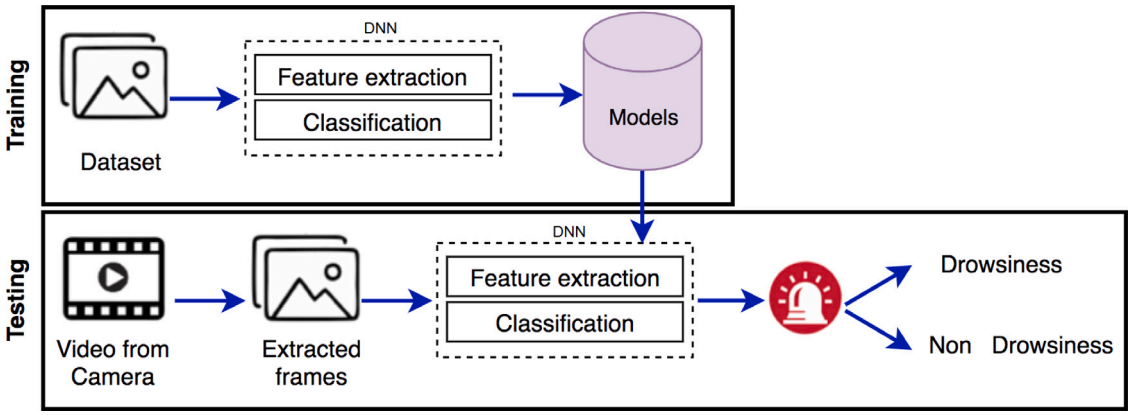
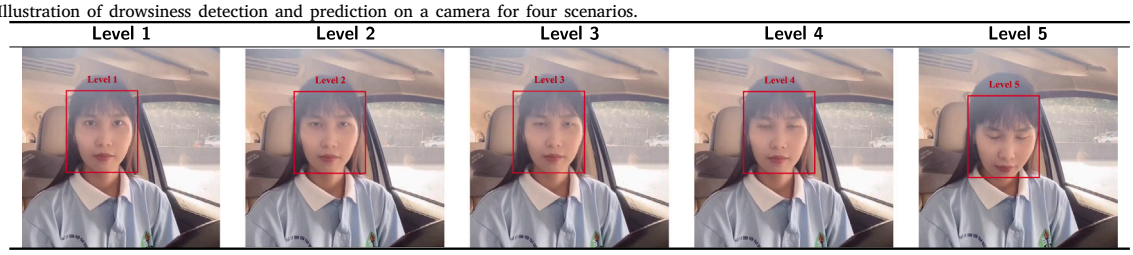


Fig. 3. Model of the drowsiness detection system using deep neural networks.

Table 1
Drowsiness levels [28].

Drowsiness level	Features
1. Not drowsy	Line of sight moves fast and frequently. Facial movements are active, accompanied by body movements.
2. Slightly drowsy	Line of sight moves slowly. Lips are open.
3. Moderately drowsy	Blinks are slow and frequent. There are mouth movements.
4. Significantly drowsy	There are blinks that seem conscious. Frequent yawning.
5. Extremely drowsy	Eyelids close. Head tilts forward or falls backward.

Table 2
Illustration of drowsiness detection and prediction on a camera for four scenarios.



It can be seen that the above studies provide high experimental results and highlight the advantages of deep learning networks. However, most of the studies are based on the eye and mouth areas ignoring some other drowsiness features such as head behavior and eyebrow movement, which affect the prediction accuracy. Besides, although the EEG and ECG-based methods achieve high accuracy, EEG and ECG devices are expensive and not readily available in vehicles. Moreover, these devices are not easy and inconvenient for the driver to use because the driver has to wear them while driving and the size of the devices is not compact. As a consequence, it is quite difficult to apply these methods to the implementation of drowsy driver monitoring systems in practice.

Dozing is a natural physiological state of human being represented by various diverse behaviors of the head, face, mouth, eyes, etc. To determine the level of driver drowsiness, we are based on a method proposed by Kitajima et al. [28] with alternative evaluation scales. The authors defined drowsiness as an interval scale consisting of five levels shown in Table 1. In this work, when the driver has drowsy characteristics greater than or equal to level 3, the detection system will make an alert to wake up the driver. The levels of the drowsy state are illustrated in Table 2.

In this study, we propose a method for drowsiness detection by analyzing data extracted from surveillance cameras thus it can be easily deployed in actual applications. Deep neural networks of the LSTM, VGG-16, Inception-V3, and DenseNet are considered to detect the drowsiness level of the driver. They are among the good CNNs widely used in object classification and detection studies. Detailed descriptions of these network architectures and their advantages are presented below.

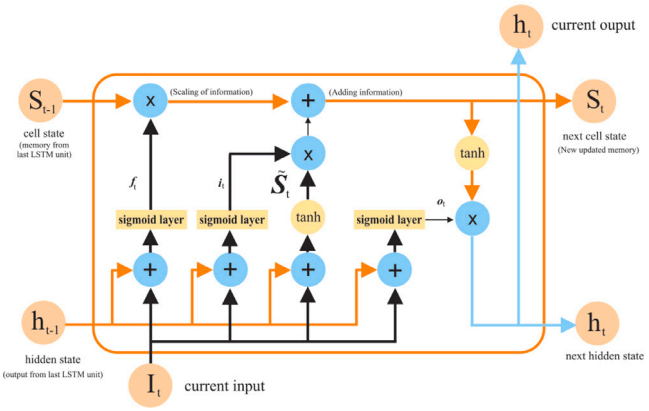


Fig. 4. Illustration of a LSTM layer.

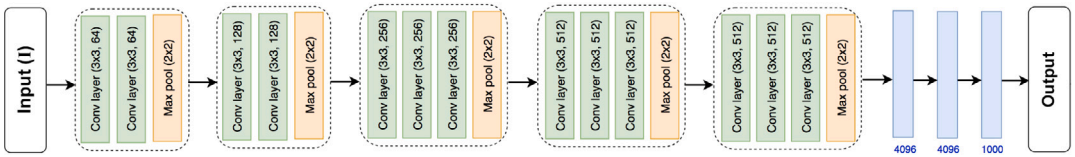


Fig. 5. Illustration of VGG-16 network architecture.

2.1. Neural network of long short term memory

Long Short Term Memory (LSTM) is a special form of Recurrent Neural Network (RNN) introduced by Hochreiter & Schmidhuber [29] that is designed to give the capability of the learning long-term dependencies. Standard Recurrent Neural Networks (RNNs) suffer from vanishing and exploding gradient problems, which hampers learning of long data sequences. Theoretically, RNNs can carry information from previous layers to later layers, but the reality is that the information can only be carried through a certain number of states and then suffer from the vanishing gradient problem. For instance, given a paragraph "I am Vietnamese. Currently, I am living abroad. I can speak fluently ...", it is clear that using only the words in the last sentence or the previous sentence is not able to predict the correct word to be filled in that is Vietnamese. LSTMs deal with these problems by introducing new gates, such as input and forget gates, which allow for better control over the gradient flow and enable better preservation of "long-range dependencies". The description of a LSTM layer is shown in Fig. 4. A typical LSTM network consists of memory blocks responsible for remembering things. Cell state (S_{t-1}, S_t) is the long-term memory allowing information from previous intervals to be stored. Data can be added to or removed from the cell state through sigmoid layers. The process of identifying and excluding data is decided by the sigmoid function, which takes the output of the last LSTM unit (h_{t-1}) at time $t-1$ and the current input (X_t) at time t. The forget gate (f_t) is a vector with values from 0 to 1 deciding how much information to get from the previous cell state. The input gate (i_t) decides how much information to get from the input of the state and the hidden layers of the previous layer. The output gate (o_t) decides how much information should be taken from the cell state to become the output of the hidden state.

2.2. Neural network of VGG-16

VGG-16 was proposed by Simonyan and Zisserman for the annual ImageNet Large Scale Visual Recognition Challenge in 2014 [30]. VGG-16 has a deep architecture with up to 138 million parameters to improve the accuracy of the model. The description of VGG-16 network architecture is shown in Fig. 5. It consists of 16 convolutional layers, which are very appealing because of its very uniform architecture. The input is passed through a stack of convolutional layers, where the filters were used with a very small receptive field of 3×3. The architecture can be described as five sets of convolutional layers, each followed by a max pool layer of stride 2×2. The first set includes two layers having 64 channels; the second set has two layers having 128 channels; the third set consists of three layers with 256 filters; the fourth and fifth sets have three layers of 512 filters. There are three fully connected layers, in which the last layer outputs 1000 channels to classify 1000 classes.

2.3. Neural network of Inception-V3

Inception-V3 [31] was introduced in 2016 with three main blocks in the Inception-V3 architecture as shown in Fig. 6. Inception-A block replaces the 5×5 convolution (in the original Inception module) by two 3×3 convolutions as 5×5 convolution is more than

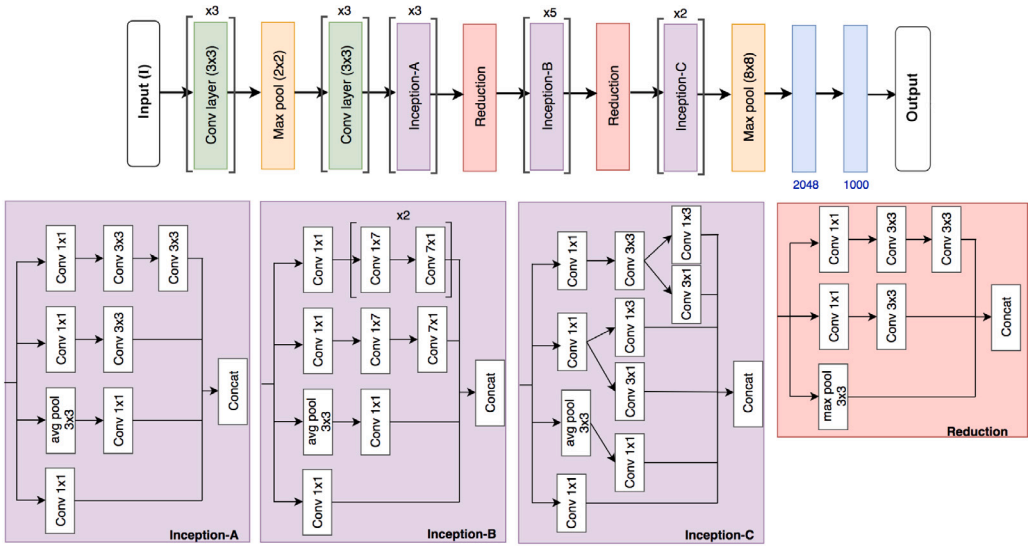


Fig. 6. Illustration of Inception-V3 network architecture.

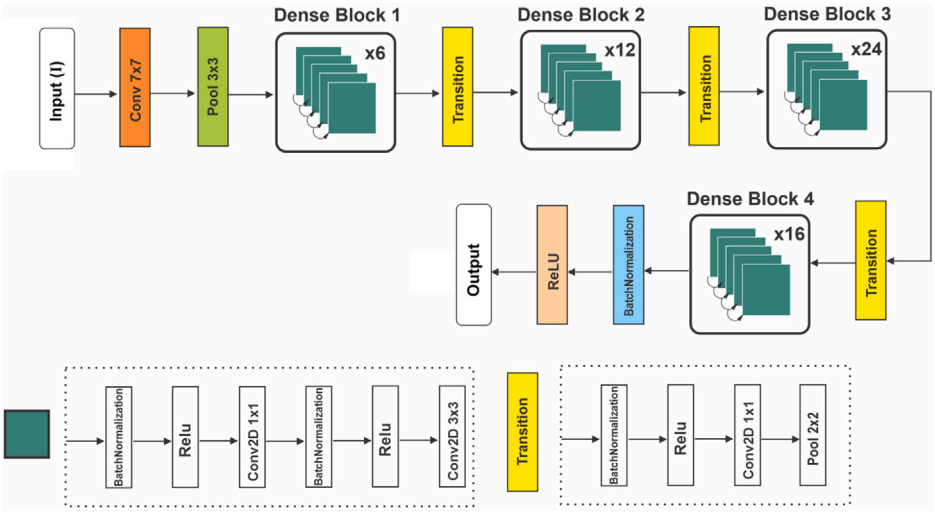


Fig. 7. Illustration of DenseNet network architecture.

2 times computationally expensive than a 3 × 3 convolution with the same number of filters. Inception-B block factorizes the n × n convolution into a 1 × n convolution followed by a n × 1 convolution and the computational cost saving increases dramatically. Inception-C block is designed to promote high dimensional representations. Besides, variations of reduction technique are used to reduce the grid sizes between the Inception blocks. In general, Inception-V3 solves the problem of representational bottlenecks, i.e., the size of layers is not reduced suddenly. At the same time, Inception-V3 has a more efficient way of computing by using factorization methods. The description of Inception-V3 network architecture is shown in Fig. 6.

2.4. Neural network of DenseNet

DenseNet [32] connects each layer to every other layer in a feed-forward fashion. It improves the information flow between layers by introducing direct connections from any layer to all subsequent layers. The description of DenseNet network architecture is shown in Fig. 7. DenseNets are divided into DenseBlocks, where the dimensions of the feature maps remain constant within a block but the number of filters between them is changed. For each layer in the block, there are three consecutive operations: batch normalization, a rectified linear unit, and a convolution. In each block, there are two convolutions: a 1 × 1 sized kernel as the bottleneck layer and 3 × 3 kernel to perform the convolution operation. The layers between the blocks are called Transition layers, which perform downsampling via convolution and pooling operations. Each transition layer has a 1 × 1 convolutional layer and a

Table 3
Confusion matrix.

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

2×2 average pooling layer with a stride of 2. DenseNet encourages heavy feature reuse so that all layers can access feature maps from their preceding layers. Therefore, it adapts for the analysis of time-varying signs of drowsiness in sequential multimedia data.

2.5. Model evaluation metrics

2.5.1. Accuracy, precision, recall, and F1

The confusion matrix is a very popular measure used for binary classification as well as multi-class classification problems. An example of a confusion matrix for classification is shown in Table 3. The confusion matrix represents counts from predicted and actual values. The output TN stands for True Negative, which shows the number of negative examples classified accurately. Similarly, TP stands for True Positive, which indicates the number of positive examples classified accurately. The term FP shows a False Positive value, i.e., the number of actual negative examples classified as positive; and FN means a False Negative value which is the number of actual positive examples classified as negative. The performance of the neural network models for classification is evaluated through the commonly used metrics of Accuracy, Precision, Recall, and F1, which are calculated in the following equations:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}. \tag{1}$$

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

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

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{4}$$

2.5.2. Loss value

The cross-entropy loss function is applied to calculate the input data loss and adjust the parameters of the networks VGG-16, Inception-V3, and DenseNet. Cross-entropy represents the distance between the actual value and the predicted value. In the back-propagation process, the larger the error between the actual value and the predicted value is, the larger the harmonic amplitude of the parameter and the faster the convergence of the model are. The cross-entropy value during training can be used to determine if overfitting occurs in the model. The cross-entropy function is represented as Eq. (3) [31], where p(k) is the predicted value, q(k) is the actual value, and k is the number of classes.

$$L = -\sum_{k=1}^{K} log(p(k))q(k). \tag{5}$$

With the LSTM network, it is related to the time and sequence of driver activities before and after dozing in the image frames. The loss function L of all time steps is determined based on the loss at each time step as in Eq. (6), where: y is the actual value, $\hat{y}$ is the predicted value, and T_y is the time step.

$$L(y, \hat{y}) = \sum_{t=1}^{T_y} L(\hat{y}^t, y^t). \tag{6}$$

3. Methodology

In this study, drowsiness detection using the proposed network models consists of two phases, the training phase and the testing phase. In order to detect drowsiness, the video clips extracted from the vehicle's surveillance camera are preprocessed by step 1 as illustrated in Fig. 8. In this step, faces and head regions are detected by SSD-ResNet-10 from image frames in these clips recorded while driving. As a result, we obtain time series data including the real-time image sequence of drowsiness duration. Then, the time series data is passed to the proposed network models in step 2 for doze detection. Based on recent works [29–36], the VGG-16, LSTM, Inception-V3, and DenseNet are among the efficient networks commonly used in object detection and classification. Therefore, we choose these networks to develop our network models by making improvements in their layers to accommodate drowsiness detection. In the training phase, we train the proposed networks on our training dataset. In the test phase, we evaluate the proposed network models on the test dataset for drowsy driver detection. Different from the training phase, the input data of the testing phase is passed

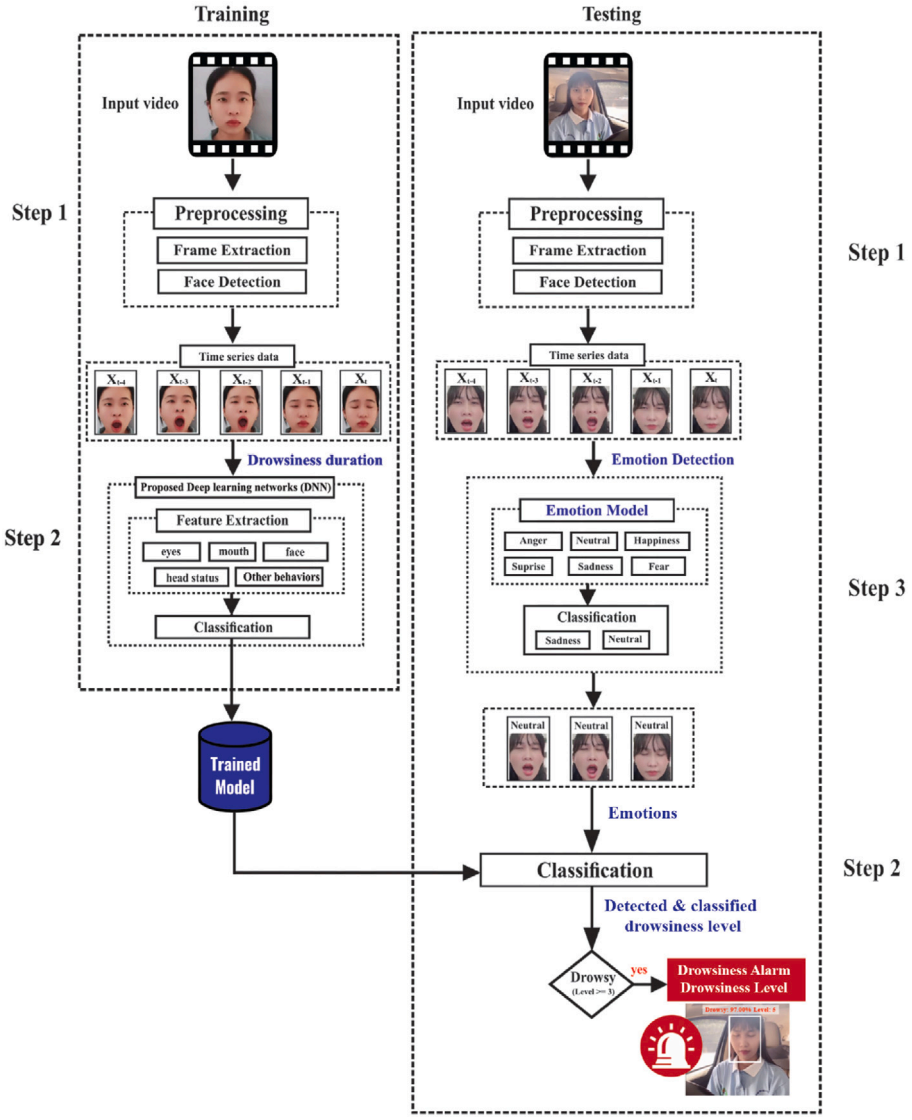


Fig. 8. Proposed method for drowsiness detection with real-time image sequence analysis.

to a pre-trained emotion detection model. This model is based on CNN (Fig. 14) proposed by Mostafa et al. [37] on the Fer2013 dataset.[1] We found that when drivers show emotions such as Angry, Disgusted, Surprised, and Fear, they rarely fall asleep. On the other hand, doze states often appear when the driver has emotions such as Neutral and Sad. We tested the emotion detection model on our experimental video dataset, and the results showed that the states associated with drowsiness fall into the Neutral and Sad emotions (Table 4). Thus, we incorporated the emotion detection model to reduce computational costs and increase the accuracy of drowsiness detection. After the emotions are classified, only images with Neutral and Sad emotions will be fed into our trained network models to detect drowsy states on the input data. If the driver has the drowsiness level greater than or equal to 3, the detection system will make an alert to wake up the driver.

The following is a detailed description of the steps of the proposed method.

3.1. Training phase

Preprocessing: In step 1, images are extracted from videos related to drowsy driving from large datasets. The extraction rate of images from videos is 25 frames per second. We then perform face detection from the image dataset using SSD network (Single

[1] https://www.kaggle.com/datasets/msambare/fer2013.

Table 4
Illustration of emotion detection experimental dataset.

Fig. 9. Illustration of preprocessing of face images extracted from videos.

Shot MultiBox Detector) with a backbone like ResNet-10 [33]. It can detect faces at different angles in a fast computation time. We then normalize the face images to the size 224 × 224 to generate a face dataset. As a result, in this step, we obtain the dataset with many different features and signs of the face and head area. Finally, we proceed to divide this dataset into two sub-datasets of drowsy and non-drowsy states. The preprocessing is illustrated in Fig. 9.

Feature extraction and training: The training dataset will be passed through the proposed deep neural networks in step 2 for feature extraction and training. We design and perfect the deep neural network models for drowsiness detection developed on LSTM, VGG-16, Inception-V3, and DenseNet by improving some of their layers. Details of the improvements are presented as follows.

3.1.1. Model 1

The previous studies of drowsiness prediction mostly focus on features of facial landmarks without considering the time variation of the driver drowsiness and leading to a prediction accuracy decrease. Hence, we propose a deep learning network developed on the LSTM network, namely model 1, to record the time information of fatigue features and analyze the complex correlation between fatigue features and the corresponding temporal information as shown in Fig. 10. Fig. 10(a) shows a general description of the proposed LSTM-based network architecture. The inputs are the frames extracted from a camera's video at time $X_{(t-i)}$. Then, inputs are passed through the cells to process the data series. The output is the drowsiness prediction results. Fig. 10(b) details the proposed network architecture from the general model in Fig. 10(a). Inputs that are signs of the drowsy states at time $X_{(t-i)}$ fed through the proposed LSTM-based network to extract features. Finally, the softmax layer helps to predict the drowsy state or not in the video data sequence.

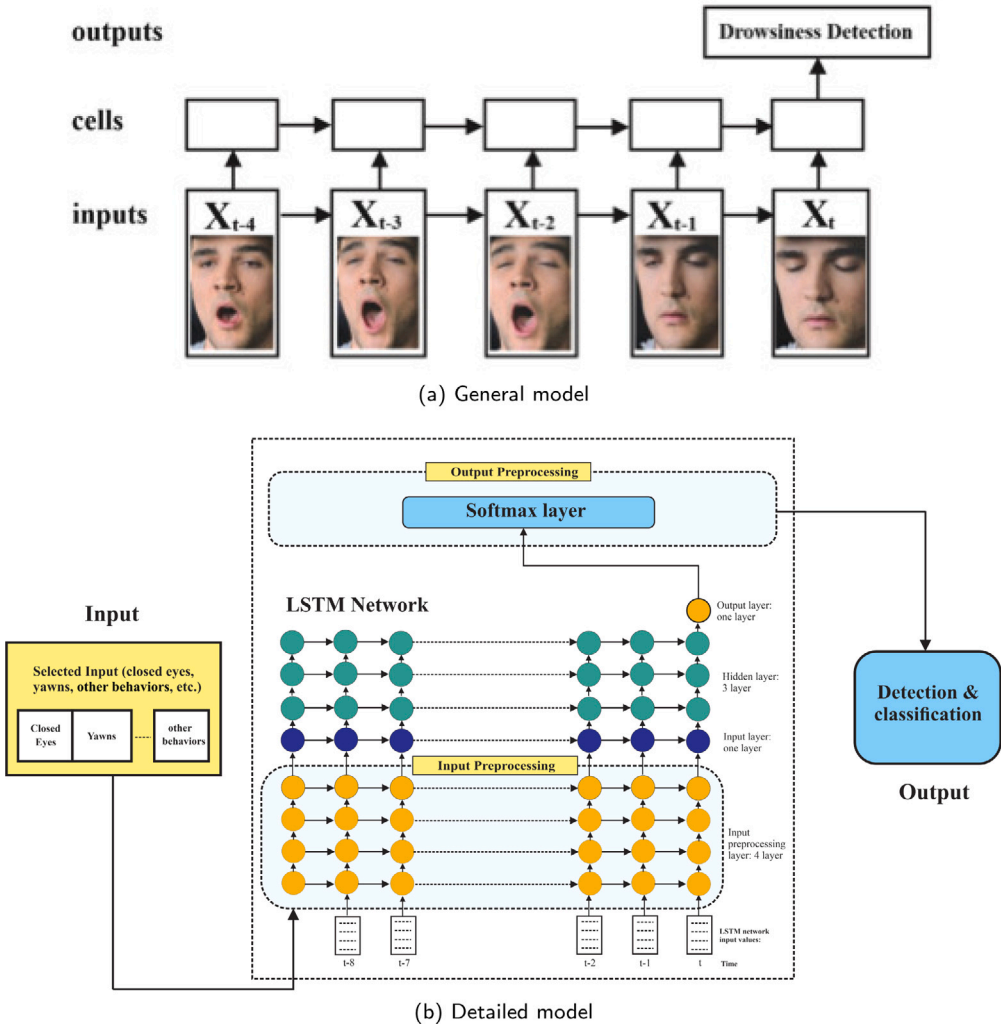


Fig. 10. Model 1: Proposed network architecture improved from LSTM.

3.1.2. Model 2

VGG-16 is a popular neural network architecture commonly used for various applications, especially in image recognition or classification. The network achieved 92.7% top-5 test accuracy on the ImageNet dataset.[2] It significantly outperformed the previous generation of models in both the ILSVRC-2012 and ILSVRC-2013 competitions. Concerning the single net performance, the VGG-16 architecture achieved the best result (7.0% test error). It increases the depth of the network, which enables to learn more complex features, and that too at a low cost. Therefore, the construction of our model architecture is also inspired by the VGG-16. Model 2 is the proposed network improved from VGG-16 architecture by adding layers such as Flatten, Relu (Dense), Dropout, and Dense (Sigmoid). This will help the model to converge faster, avoid rote learning, and provide better results. The added layers are represented by the red dashed lines in the network model shown in Fig. 11.

3.1.3. Model 3

The larger the deep learning model is, the more likely it is to over-fitting and the number of parameters also increases leading to the increased demand for computing resources. In the Inception model [31], these problems are addressed, which allows for increasing the depth and width of the deep learning model while the computational cost is unchanged. With these outstanding advantages, we choose the Inception model with some improvements to be able to accommodate doze detection. Model 3 is the deep neural network model that builds up from the pre-trained Inception-V3 and adds some layers such as Flatten, Relu (Dense), Dropout, and Sigmoid (Dense). The added layers are represented by the red dashed lines shown in Fig. 12.

[2] https://blog.paperspace.com/popular-deep-learning-architectures-alexnet-vgg-googlenet, accessed on 05 March 2022.

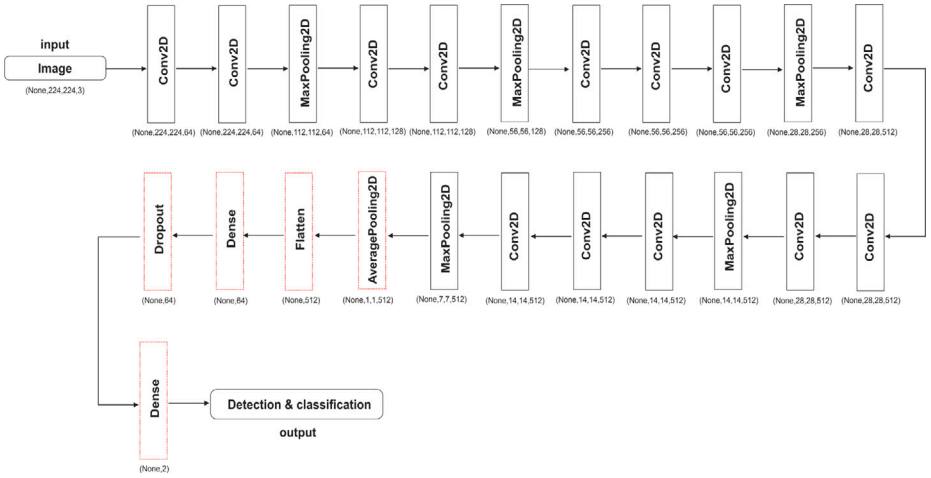


Fig. 11. Model 2: Proposed network architecture improved from VGG-16.

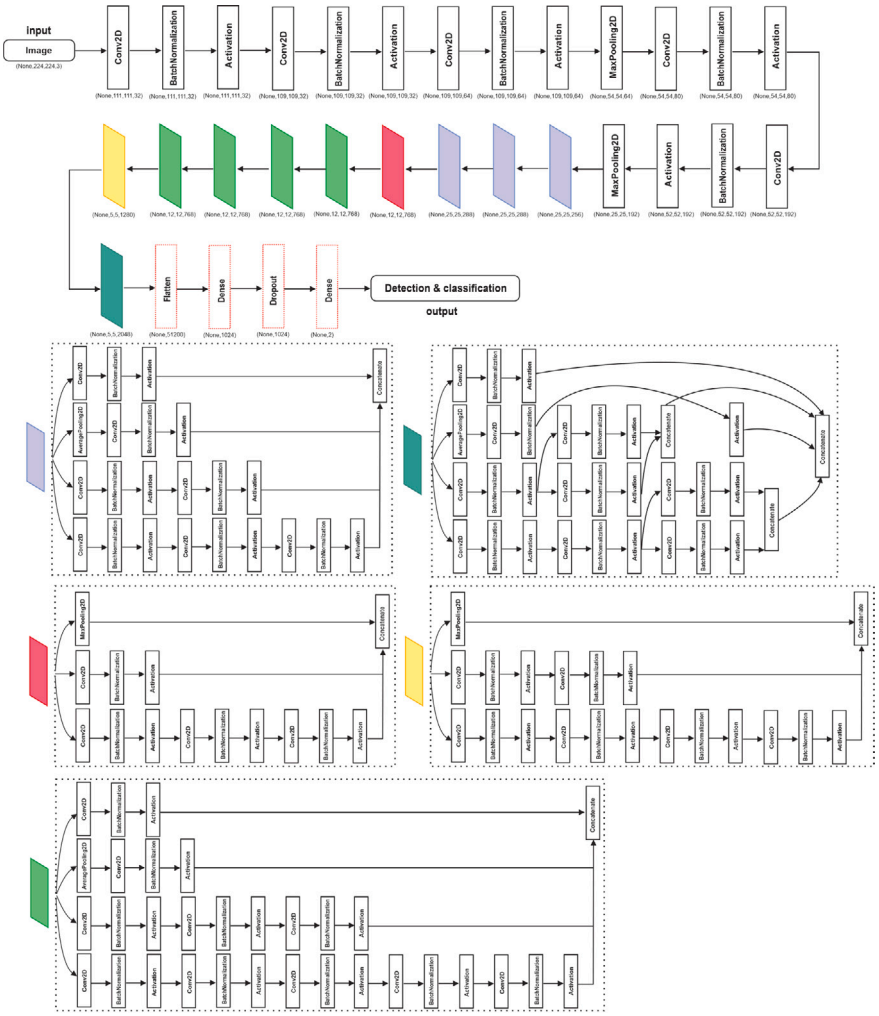


Fig. 12. Model 3: Proposed network architecture improved from Inception-V3.

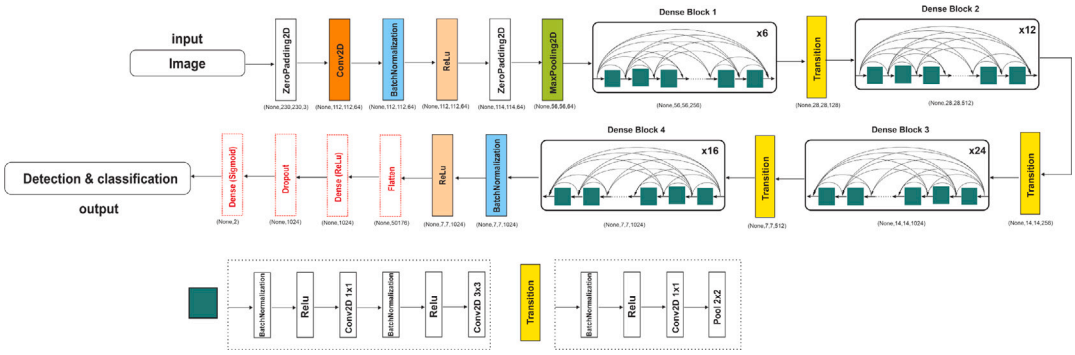


Fig. 13. Model 4: Proposed network architecture improved from DenseNet.

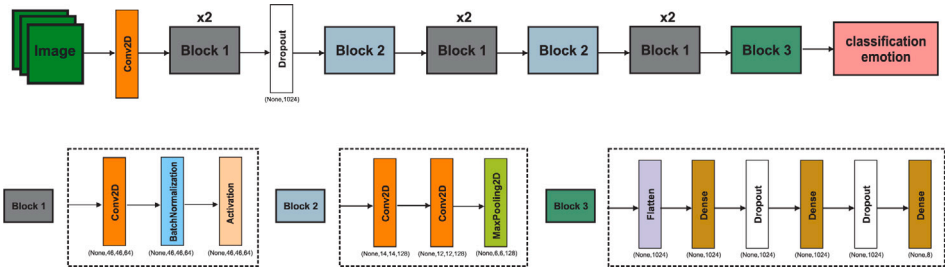


Fig. 14. Emotion detection model proposed by Mostafa et al. [37].

3.1.4. Model 4

Recent studies [32,33] have shown that the dense connectivity pattern help in increasing the performance of convolutional networks. Compared to conventional convolutional networks, DenseNet has outstanding advantages such as the number of parameters being significantly reduced, the features being reused, and the vanishing gradient being mitigated. In order to ensure maximum information flow between layers in the network, we, therefore, connect all layers with matching feature-map sizes directly with each other. Because of the dense connectivity feature, we refer to our approach as DenseNet to develop network model 4 for drowsiness detection in this study.

Model 4 is built from the pre-trained DenseNet and adds some layers of Flatten, Relu (Dense), Dropout, and Sigmoid (Dense). The additional layers are represented by the red dashed lines shown in Fig. 13. It is feasible and suitable for tracking and analyzing time-varying signals of drowsiness in sequential multimedia data by a simple connectivity model. In the dense block, each layer connects to every other layer in a feed-forward mode. The feature maps of all preceding layers and their feature maps are used as inputs to all subsequent layers. This makes the input features of the next layer diversified, reduces the depth of the network, and effectively improves the computation to learn distinct features. The proposed network models are pre-trained on several datasets such as Bing Search API, Kaggle, and RMFD. We use pre-trained weights and re-train these models on our dataset with the transfer learning method to be suitable for drowsiness detection. This approach helps to shorten training time, does not require large training datasets, and provides better learning of all dozy features. During the training phase, the decision to stop the training is made when the Loss and Accuracy values are no longer improved (i.e., the values do not decrease or increase) after some epochs.

3.2. Testing phase

In the test phase, the input data is the images extracted from camera videos. The first step of this phase works exactly the same as in the training phase except that the input data is collected from cameras. These images are then transferred to the pre-trained emotion detection model. After the emotions of the input data are identified, if the emotions fall into the state of Neutral or Sad, the proposed network models are applied to detect drowsy states. That is, if the driver has the level of drowsiness greater than or equal to 3, the detection system will make an alert to wake up the driver. We also analyze and compare the experimental results to evaluate the network models.

4. Experiments

4.1. Dataset description and installation environment

Drowsiness is the natural cyclical rest state of the body and mind [38]. In this state, people often close their eyes and lose consciousness partially or completely, thereby reducing their response to external stimuli. It may accompany other symptoms

Table 5
Experimental datasets.

Status	Type	Number	Num faces	Size
Drowsy	Image	3425	3425	230 MB
Non-drowsy	Image	4388	4388	66.1 MB
Both	Video	24	13,729	1760 MB

Table 6
Scenarios of the proposed methods for experiments.

Scenario	Technique of feature extraction and prediction
1	Model 1: Adaptive deep neural network developed on LSTM
2	Model 2: Adaptive deep neural network developed on VGG-16
3	Model 3: Adaptive deep neural network developed on Inception-V3
4	Model 4: Adaptive deep neural network developed on DenseNet

Table 7
Experimental parameters of scenarios.

Sce	Batch size	Learning rate	Epoch	Score	Image size	Additional layers
1	30	$1e^{-4}$	100	Softmax	64 × 64	Dropout (0.5) Flatten, Densen (Relu) Dense (Softmax)
2	50	$1e^{-4}$	30	Sigmoid	224 × 224	Flatten, Densen (Relu) Dropout (0.5) Dense (Sigmoid)
3	250	$1e^{-4}$	30	Sigmoid	224 × 224	Flatten, Densen (Relu) Dropout (0.8) Dense (Sigmoid)
4	250	$1e^{-4}$	30	Sigmoid	224 × 224	Flatten, Densen (Relu) Dropout (0.8) Dense (Sigmoid)

consisting of yawning, closing eyes, blinking repeatedly and difficulty opening eyes, the inability to concentrate, the inability to keep the head straight, a distracted mind, feelings of tiredness, blurred vision, etc. From these symptoms, we proceed to build datasets of drowsy and non-drowsy by manual classification based on a sequence of image frames extracted from large video databases of Bing Search API, Kaggle, RMFD, iStock, Road Safety Commission, and CJI: Traffic-Safety. The use of recorded images and videos related to drowsy driving is convenient for data collection, cost savings, and safety in drowsy driver experiments. In these images and videos, the driving environment is relatively similar to the actual road experiments. The experimental dataset of 21,542 facial images contains 3425 images of the drowsy state (960 images of level 3, 1240 images of level 4 and 1225 images of level 5), 4388 images of the non-drowsy state (3098 images of level 1 and 1290 images of level 2) for training phase, and 13,729 images of both states including five levels extracted from videos for testing phase. Data contexts include wearing a mask, wearing glasses, yawning, drowsiness, tilting head, head down, and not looking directly at the camera. The training dataset of 7813 images is divided into 80% (6250 images) for training and 20% (1563 images) for validation. The number of images and size of the dataset is described in Table 5.

In order to compare and evaluate the proposed network models, we conduct experiments on the same environment of Visual Studio Code with Windows 10. The configuration of the computer is 32 GB RAM and Nvidia Geforce GPU. The library for training the proposed network models is Tensorflow version 2.2.

4.2. Scenarios and parameters

Experiments of the proposed models are performed on four scenarios as presented in Table 6.

During the training process, we fine-tune the model parameters such as Batch size, Learning rate, Epoch, and Score until the optimal parameter values are reached to obtain the highest accuracy and lowest loss. Details of the parameters are shown in Table 7.

4.3. Experimental results

4.3.1. Training results to evaluate scenarios

To evaluate the performance of network models, we calculate metrics of Precision, Recall, and Accuracy through the confusion matrix as presented in Section 2.5. Besides, we also determine Loss_value and training time to fine-tune and compare the proposed network models (scenarios).

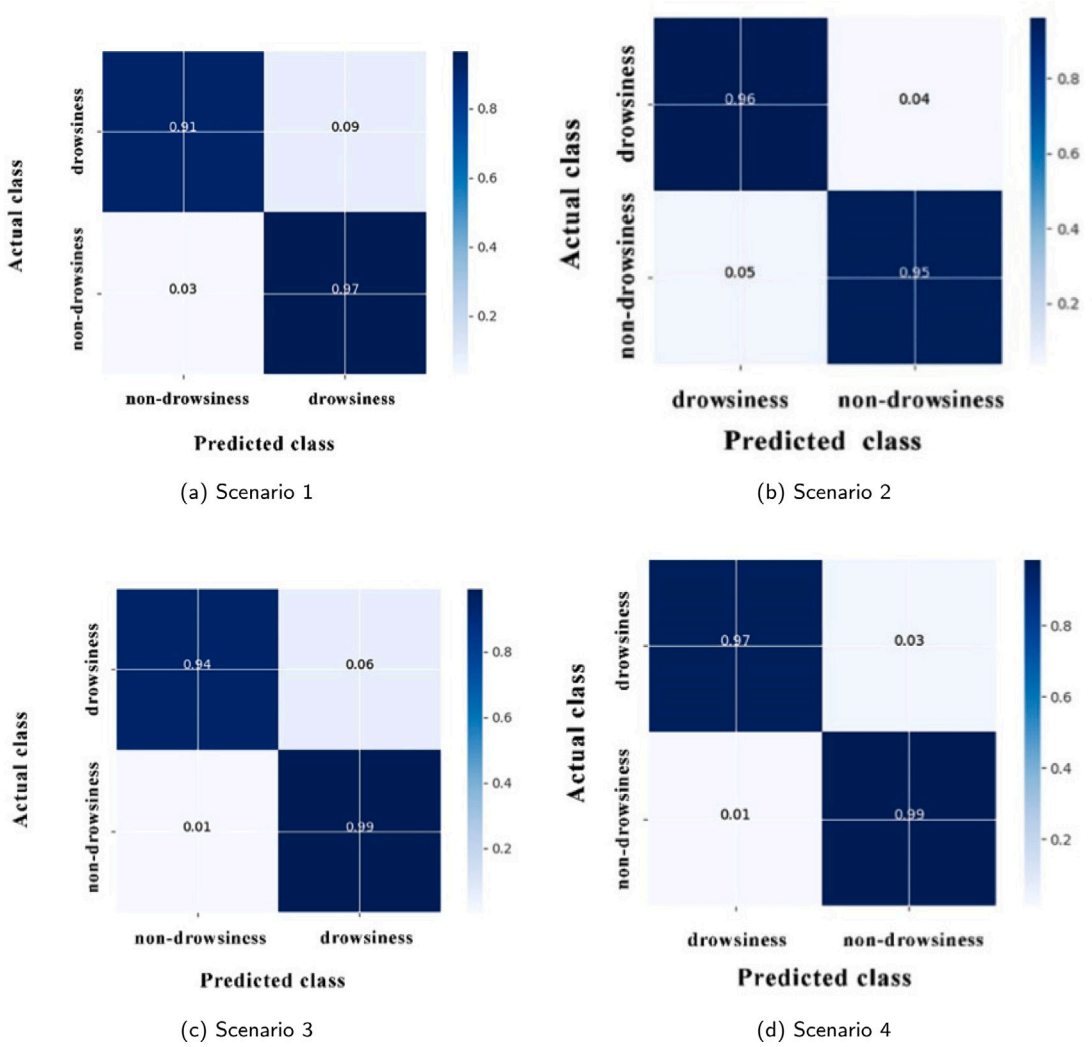


Fig. 15. Confusion matrices for four scenarios.

(a) Confusion matrix Fig. 15 shows the confusion matrices of the four scenarios. Fig. 15(a) shows the confusion matrix for scenario 1, where the values of TP and TN for predicting drowsy and non-drowsy states are 91% and 97%, respectively. On average, this is the model with the lowest values of TP and TN. In contrast, scenario 4 has the highest values of TP and TN, 97% and 99%, respectively (Fig. 15(d)).

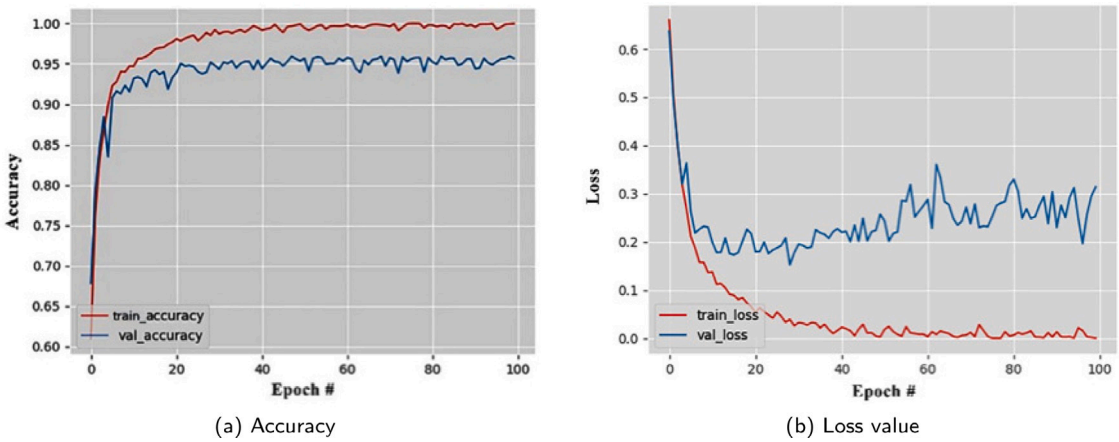
(b) Metrics of the precision, recall, F1, and accuracy The metrics of the Precision, Recall, and Accuracy for four scenarios are shown in Table 8. The training accuracies are 94%, 96%, 97%, and 98% corresponding to the four scenarios 1, 2, 3, and 4. We can see that scenario 4 with the proposed neural network improved from DenseNet provides the highest accuracy compared to the rest of the scenarios because it considers the time-varying features of driver drowsiness. Particularly, each layer of this network has direct access to gradients from the original input data leading to implicit deep supervision.

(c) Metric of Loss_value During the training process of networks, the main purpose is to minimize Loss (in terms of error or cost) observed in the output when training data is sent through them. The results of Loss_value and Accuracy of the four scenarios are illustrated in Figs. 16, 17, 18, and 19, where the y-axis represents the accuracy and Loss_value of the scenarios and the x-axis represents the number of epochs. The accuracy increases and the Loss_value decreases gradually when the number of epochs increases.

Fig. 16 shows the Loss_value and accuracy of scenario 1 after 100 epochs with the learning_rate of 1e-4. We decide to stop the training process when the number of epochs reaches 40 since neither the accuracy of the network nor the Loss_value is further improved. The accuracy increases rapidly in the first 40 epochs (Fig. 16(a)). After 50 epochs, the train_accuracy achieves 100% whereas the val_accuracy reaches 94%. Thus, the average training accuracy is 97%. The Loss value decreases rapidly from epochs

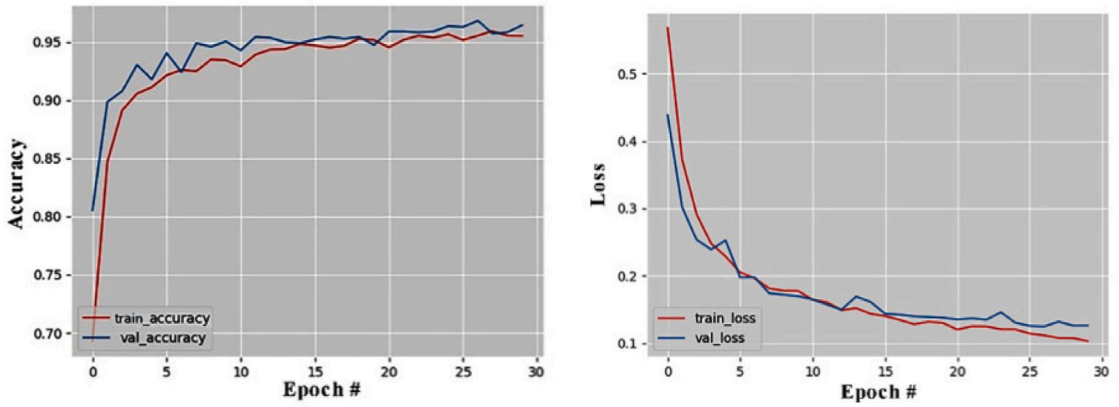
Table 8
Evaluation metrics for four scenarios in training phase.

Scenario	Label	Precision	Recall	F1-score	Support	Accuracy
1	Drowsy	0.96	0.91	0.93	685	0.94
	Non-drowsy	0.93	0.97	0.95	878	
2	Drowsy	0.97	0.96	0.97	685	0.96
	Non-drowsy	0.96	0.98	0.98	878	
3	Drowsy	0.99	0.94	0.96	685	0.97
	Non-drowsy	0.95	0.99	0.97	878	
4	Drowsy	0.99	0.97	0.98	685	0.98
	Non-drowsy	0.98	0.99	0.99	878	



(a) Accuracy (b) Loss value

Fig. 16. Metrics of Accuracy and Loss_value for scenario 1.



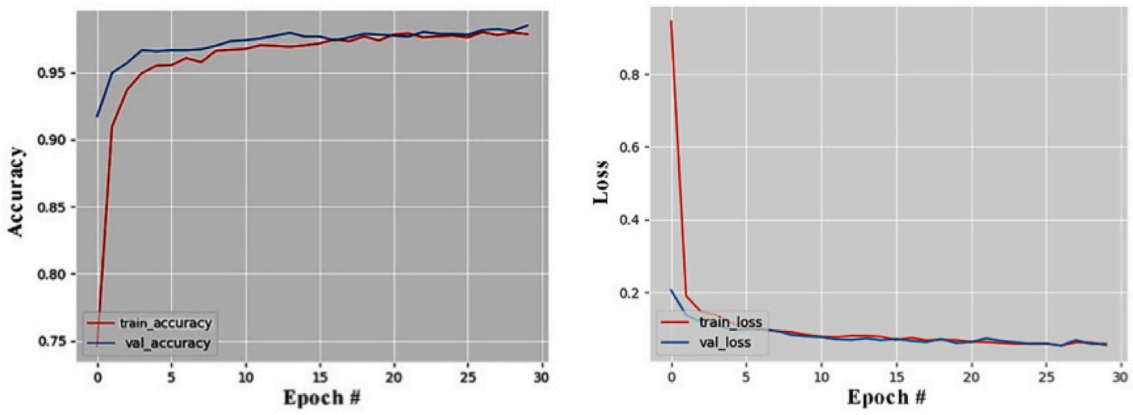
(a) Accuracy (b) Loss_value

Fig. 17. Metrics of Accuracy and Loss_value for scenario 2.

0 to 40 (Fig. 16(b)). As the epoch value reaches 50, the train_Loss approximates to zero and the val_Loss is 0.3, thus the average Loss value is 0.15. The val_Loss for scenario 1 is quite large, which means this model has a high fault prediction rate.

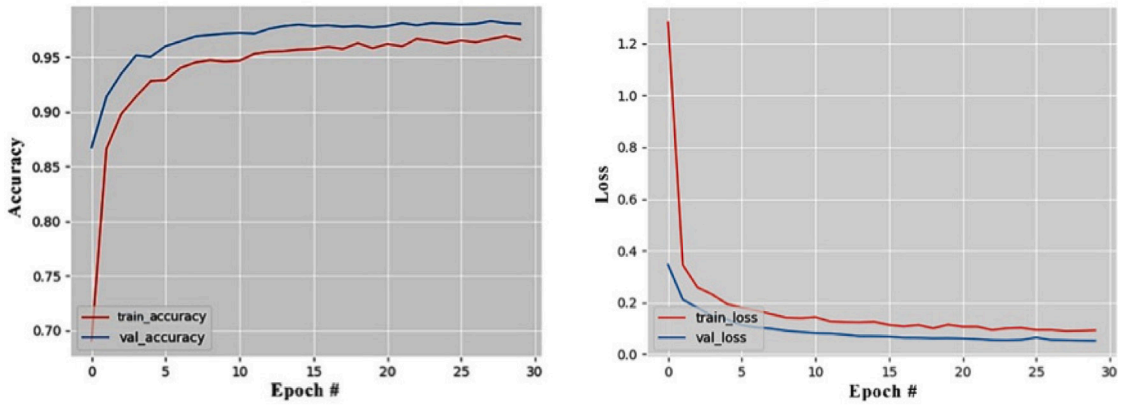
Fig. 17 shows the Loss_value and accuracy of scenario 2 with 30 epochs. The average accuracy in this scenario is 95.5%, which is lower than that of scenario 1. However, scenario 2 has val_accuracy higher than train_accuracy meaning that it tends to learn better than scenario 1. The train_Loss approximates 0.1 and the val_Loss value approximates 0.13 thus we have the average Loss value of scenario 2 is 0.11. Scenario 2 has an average Loss value lower than that of scenario 1, meaning that scenario 2 can predict drowsiness with more stable accuracy than scenario 1.

The Loss_value and accuracy of scenario 3 are presented in Fig. 18. The training phase is stopped after 30 epochs since the accuracy and Loss are not improved. The train_accuracy and the val_accuracy reach 96% and 97% (Fig. 18(a)), respectively, thus the average accuracy in this scenario is 96.5%. This network model tends to learn efficiently. Fig. 18(b) represents the train_Loss



(a) Accuracy (b) Loss value

Fig. 18. Metrics of Accuracy and Loss_value for scenario 3.



(a) Accuracy (b) Loss value

Fig. 19. Accuracy and Loss_value metrics for scenario 4.

and val_Loss values of the training process, approximately 0.06. This is significantly lower than those of scenarios 1 and 2. It means that scenario 3 works better with high accuracy and a low error rate in giving drowsiness predictions.

In scenario 4, the Loss value and accuracy are shown in Fig. 19. Similar to previous scenarios, the accuracy and Loss value are not improving after 30 training steps. The average accuracy is 97.5%, which is higher than those of scenarios 1, 2, and 3. It shows that scenario 4 tends to learn and detect drowsiness effectively. In Fig. 19(b), the train_Loss and val_Loss are 0.08 and 0.02, respectively. The average Loss value is 0.05, lower than those of the remaining scenarios.

In brief, scenario 1 achieves a higher average accuracy than scenarios 2 and 3 but the val_accuracy of scenario 1 is the lowest because its Loss_value is quite high. Meanwhile, the other three scenarios have higher val_accuracy values than train_accuracy values. This is one of the reasons why scenario 1 gives the low drowsiness prediction results in the test phase. This also indicates that all three scenarios 2, 3, and 4 learn and detect well all features of drowsiness. Regarding the Loss_value, scenario 1 has the highest average Loss value among the four scenarios. Although the difference in Loss_value of scenario 2 is not as large as in scenario 1, this scenario has a high error prediction rate due to the average Loss_value of 0.11. The other two scenarios have many outstanding advantages such as a fast computation process, a deeper network architecture that increases efficiency in extracting drowsiness features, and automatic learning of all these features. Therefore, the average Loss and accuracy values in scenarios 3 and 4 have significantly improved compared to scenarios 1 and 2. Moreover, with an effective over-fitting solution and reduction of vanishing gradient, scenario 4 gives better results than the remaining scenarios. In scenario 4, val_accuracy is higher than train_accuracy and val_Loss is lower than train_Loss. It shows that scenario 4 tends to learn drowsiness features efficiently and gives high accuracy results with the lowest error rate when predicting drowsiness. The reason for the good results in scenario 4 is that the proposed network connects all layers in such a way each layer obtains additional inputs from all preceding layers and passes their feature maps to all subsequent layers. It carries out features from all preceding layers to all upcoming layers by feed-forward direct concatenation. Hence, it is suitable for the analysis of time variation of the drowsiness features in sequential multimedia data.

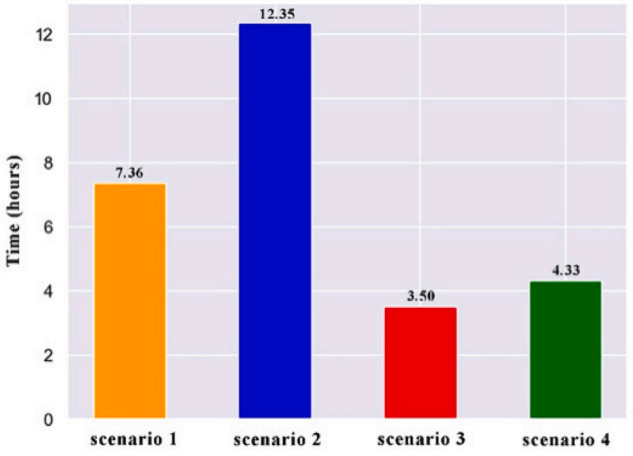


Fig. 20. Training time of the four scenarios.

(d) Training time Fig. 20 shows the training times of four scenarios. It can be seen that, if all scenarios are trained at 30 epochs, scenario 1 will give the fastest training time, about 2 h and 03 min. Scenario 2 is one of the networks with a large number of parameters up to 550 MB in size. This leads to a large model taking a lot of time to complete the training process, i.e., 12 h 35 min.

4.3.2 Test results

(a) Accuracy and time of drowsiness prediction Experiments are conducted on a total of 33 different people from the videos including drowsy and non-drowsy states. Fig. 21 shows the results of the average accuracy and average prediction time when detecting the person falling into a drowsy state. The average accuracy of scenarios 1, 2, 3, and 4 are 93.5%, 96.1%, 97.5%, and 98.5%, respectively. Obviously, scenario 4 outperforms the remaining scenarios in accuracy and the test accuracy is higher than the training accuracy. Besides, the average prediction time is 3.5 min for scenario 1, 4.8 min for scenario 2, 5.7 min for scenario 3, and 5.9 min for scenario 4. Scenario 1 quickly gives prediction results but will easily lead to confusing results between blinking and falling asleep. In contrast, combining all the drowsiness signs to give an accurate prediction result will need more time.

(b) Drowsiness prediction Table 9 presents the experimental results from videos of the four scenarios. In case A, the prediction result of 93.17% of scenario 4 is superior to scenarios 1, 2, and 3. In case C, all four scenarios give the desired results and the highest accuracy at this time also belongs to scenario 4, which is 97.89%. In case D, all four scenarios again give correct results, however, the highest accuracy at this time belongs to scenario 3, which is 88.64%. In case B, scenario 1 gives an incorrect result when the driver has fallen asleep but the predicted state is non-drowsy. On the contrary, the remaining scenarios give correct results and scenario 4 gives the highest prediction accuracy of 66.11%. Besides, in case E, scenario 1 continues to give a wrong prediction of the drowsy state when the driver does not have any sign of drowsiness. Finally, in case F, scenario 1 again gives a false result and scenario 3 gives the highest accuracy of 89.13%. It can be concluded that scenarios 3 and 4 give better results than the remaining scenarios.

Table 10 shows the experiment results from a camera of the four scenarios. Scenarios 2, 3, and 4 give accurate results in all cases including with/without wearing a mask and glasses. In case A, all four scenarios give correct results, but scenario 1 gives lower prediction accuracy than the remaining scenarios. In case B, the predictions of four scenarios are correct that the driver has fallen into sleep, but scenarios 3 and 4 give higher accuracy of 97% and 98.9%, respectively. In case C, all four scenarios give correct results when the driver is wearing glasses, but scenario 1 has an accuracy of 68.81% whereas the other three scenarios have an accuracy of more than 96%. In case D, the driver is falling asleep (wearing glasses) and the highest prediction result belongs to scenario 4 with 83.02%. In case E, all four scenarios give correct results when the driver wears a mask and glasses. Scenarios 2 and 4 give better results of greater than 99.6%. In case F, scenario 1 gives an incorrect result when the driver falls asleep (with mask and glasses). The other three scenarios have correct results and the highest accuracy is in scenario 3 with 97.16%. Table 10 shows that scenarios 3 and 4 give better results than the two remaining scenarios.

5 Comparison and discussion

In this section, we compare the accuracy and Loss measure between the proposed methods and the existing methods on our experimental dataset with or without monitoring the time variation of sleepiness.

Table 11 and Fig. 22 illustrate the comparison results of the accuracy and average Loss measurement of drowsiness detection methods. In Fig. 22(a), we can see that the Loss measure for scenario 1 is the highest while this value for scenario 4 is the lowest compared to the remaining methods. This means that scenario 4 has the optimal Loss measure and the lowest false drowsiness predictions. Indeed, we can see that Table 11 and Fig. 22(b) represent an accuracy comparison of the methods. The training results

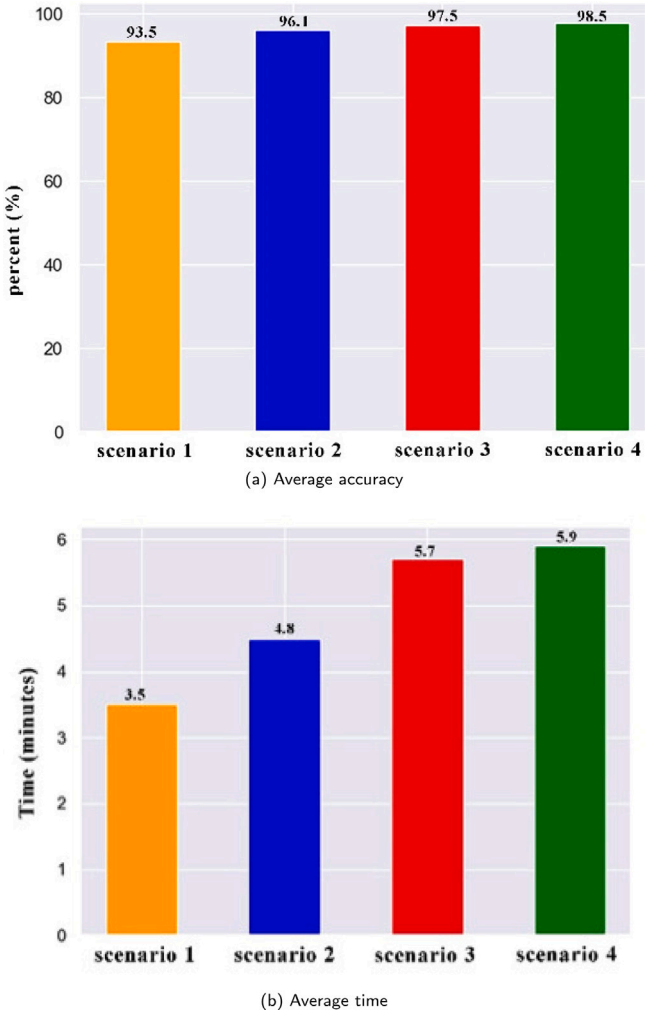
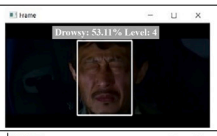
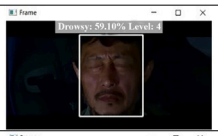
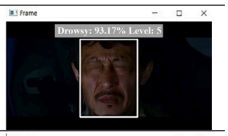
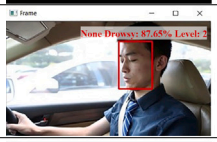
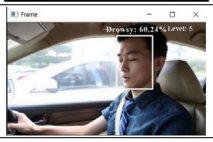
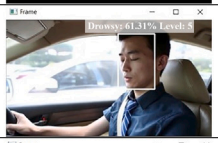
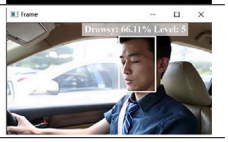
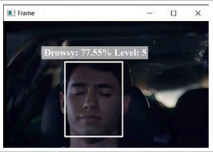
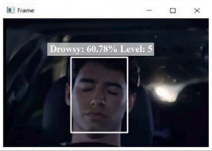
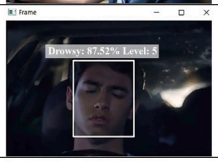
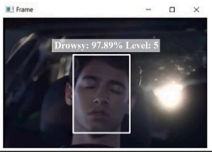
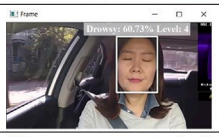
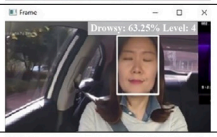
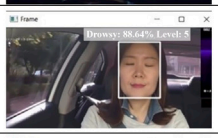
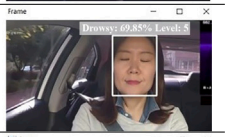
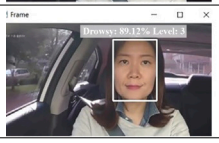
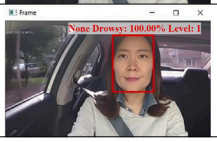
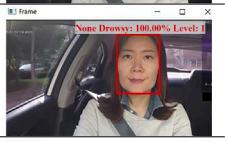
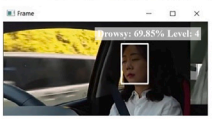
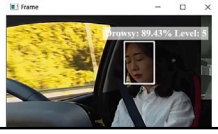
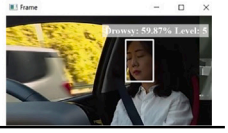


Fig. 21. Average accuracy and time of drowsiness prediction for four scenarios.

in the four scenarios 1, 2, 3, and 4 give the accuracy of 94%, 96%, 97%, and 98%, respectively. Meanwhile, the accuracy of the methods using the existing pre-trained networks ranges from 84% to 97%. It can see that the validation accuracy of scenario 4 is better than that of the remaining methods. Besides, we compare the methods before and after improvement. Obviously, the proposed methods (shown in orange) provide better results than pre-improvement methods (shown in blue) in terms of Loss and accuracy as illustrated in Fig. 22. For example, scenario 1 with the improved LSTM network provides an accuracy value of 94%, which is 5% higher than the method using the LSTM network. Moreover, we also make a comparison of the methods with or without tracking the time variations of dozing. Obviously, scenario 4 achieves the highest accuracy of 98% compared to other methods due to the consideration of the time series data during the drowsiness process. However, scenario 1 using the improved LSTM network provides no better results than scenarios 2 and 3 which do not analyze the time series data of the drowsiness because it is prone to overfitting and cannot completely solve the problem of vanishing gradients. Thus, scenario 1 is not able to keep track of long-term dependencies as scenario 4.

The experimental results have highlighted the advantages of the proposed methods as follows: (1) The accuracy of the proposed methods with four scenarios is higher than the previous research methods. Scenario 4 shows the outstanding advantages such as avoiding the vanishing gradient problem, encouraging feature reuse, analyzing time-varying signals of drowsiness in sequential multimedia data, especially reducing the number of parameters compared to the network model developed from MobileNet-V2 in the study [7]. On the other hand, the proposed network models are good feature extractors, as they can capture the relevant characteristics of the driver's drowsy states through images or videos; (2) The test accuracy of 98.5% provided by scenario 4 is superior to our previous study [7] of 97%. However, the training time of scenario 4 is longer than that of the previous study. This is evident in the field of deep learning that neural network models with high accuracy need to pay the price in terms of training and recognition time; (3) Instead of focusing on the eyes and mouth area as in previous works [13–16,18–22], the proposed deep neural networks analyze all activities of the driver to learn drowsy signs such as inability to keep the head straight, yawning, and blinking

Table 9
Illustration of drowsiness detection and prediction on videos for four scenarios.

Case \ Scenario	1	2	3	4
A.				
B.				
C.				
D.				
E.				
F.				

continuously; These signs are used to predict the drowsy state. Experiments are conducted on large datasets to test the system's accuracy, which is up to 98%; (4) We take advantage of transfer learning to provide fast training time and keep the advantages of deep learning networks; (5) The proposed method gives accurate results on the driver's drowsiness in cases with/without wearing a mask and glasses. This has practical meaning in the epidemic situation when everyone needs to comply with the wearing of masks in public places. The previous studies do not provide accuracy in these cases. The four proposed deep neural networks for drowsiness detection are conducted on scenarios 1, 2, 3, and 4 achieving high accuracy of up to 98%, and thus they can work well to implement as a practical system.

6 IoT-based smart alert system for drowsy driver detection

Drowsiness is one of the main causes of accidents alongside with other causes like drunk driving, distractions, and so on. In order to overcome this issue, we construct a driver drowsiness detection system using deep learning combined with IoT to be able to detect, alert and potentially save a person's life. For our system, a surveillance camera is used to capture the images of the driver's activities and the entire system is incorporated using Jetson Nano. When the driver is drowsy, the IoT module emits a warning message along with impact of collision, location information, and a sound through a Jetson Nano monitoring system.

From the evaluation results, we choose the model in scenario 4 (based on DenseNet) to build an IoT-based drowsiness detection system. Fig. 23 shows the general model of an IoT-based drowsiness detection system. The proposed system is made up of the primary components including a Jetson Nano, an Arduino Micro, a Buzzer, and a surveillance camera (e.g. Logitech Webcam). NVIDIA Jetson Nano[3] is an embedded system-on-module (SoM) and developer kit from the NVIDIA Jetson family, including an integrated 128-core Maxwell GPU, quad-core ARM A57 64-bit CPU, 4 GB LPDDR4 memory, along with support for MIPI CSI-2 and PCIe Gen2 high-speed I/O. There is also the Jetson Nano 2 GB Developer Kit with 2 GB memory and the same processing specs. Useful for deploying computer vision and deep learning, Jetson Nano runs Linux and provides 472 GFLOPS of FP16 compute performance with 5–10 W of power consumption. The Arduino Micro[4] is a micro-controller board based on the ATmega32U4, a low-power CMOS

[3] https://developer.nvidia.com/embedded/jetson-nano-developer-kit.
[4] https://store.arduino.cc/products/arduino-micro

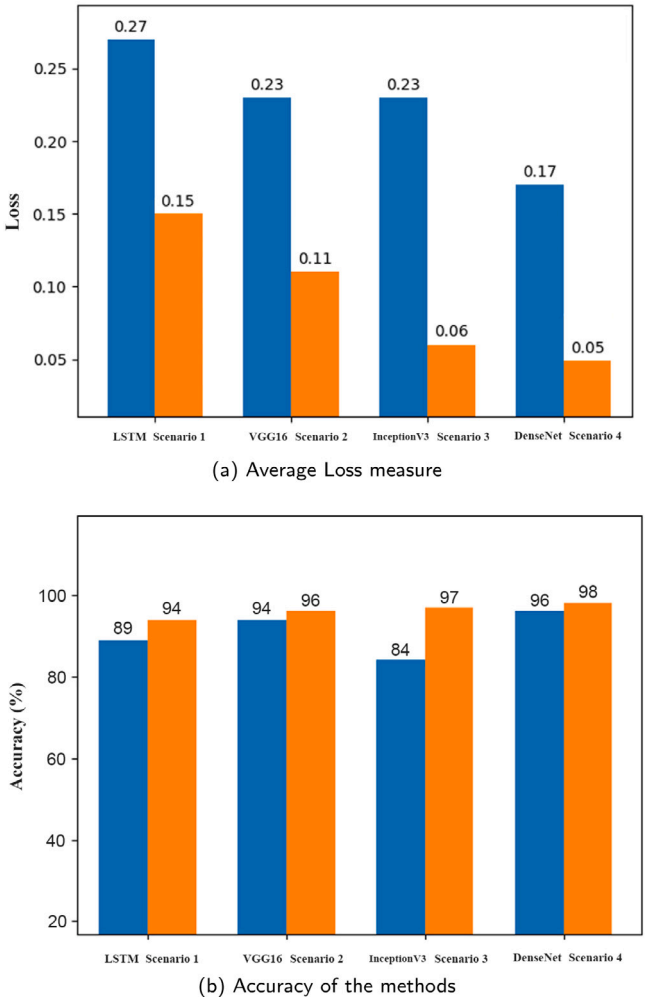


Fig. 22. Comparison chart of the Loss and accuracy measure of the methods before (blue) and after (orange) improvement.

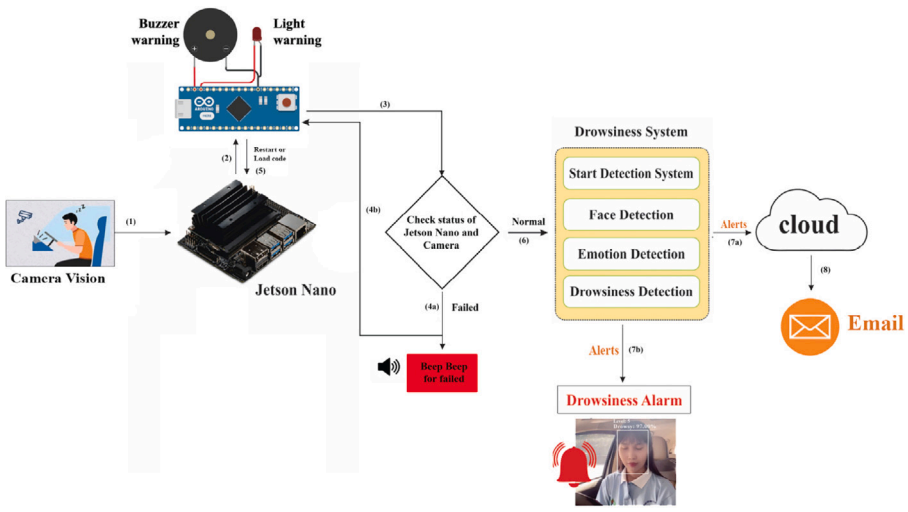


Fig. 23. IoT based driver drowsiness detection system.

Table 10
Illustration of drowsiness detection and prediction on a camera for four scenarios.

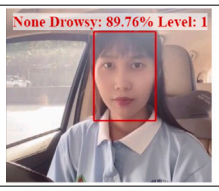
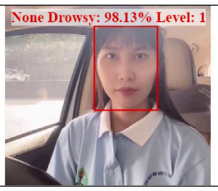
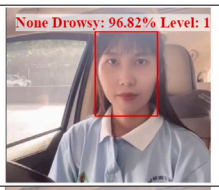
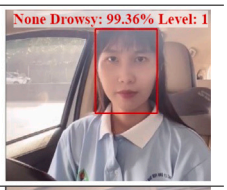
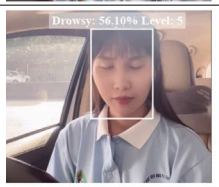
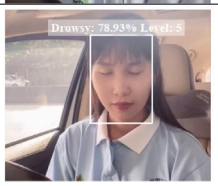
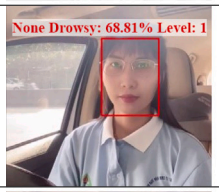
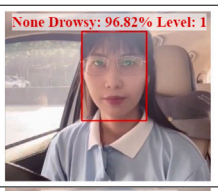
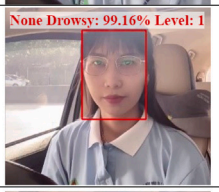
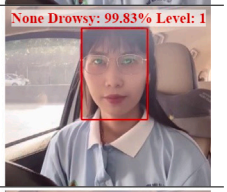
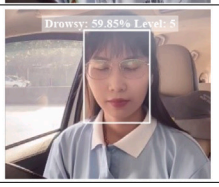
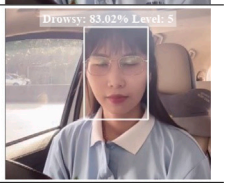
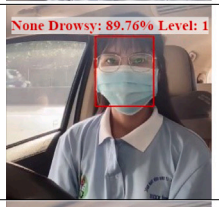
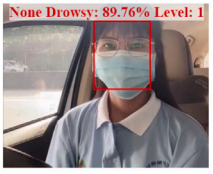
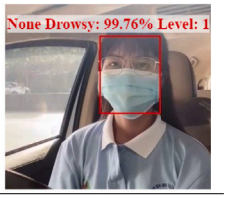
Case \ Scenario	1	2	3	4
A.				
B.				
C.				
D.				
E.				
F.				

Table 11
Accuracy comparison of drowsiness detection methods.

Authors	Method	Accuracy
Szegedy, et al. [13]	The inception architecture for computer vision with Inception-V3	84%
Hochreiter et al. [29]	Long short-term memory (LSTM)	89%
Simonyan et al. [30]	Very deep convolutional networks for large-scale image recognition with VGG-16	94%
Huang, et al. [32]	Densely connected convolutional networks with DenseNet	96%
Phan et al. [7]	An Efficient Approach for Detecting Driver Drowsiness Based on Deep Learning.	83%–97%
The proposed method with scenario 1	Deep learning network improved from LSTM.	94%
The proposed method with scenario 2	Deep learning network improved from VGG-16.	96%
The proposed method with scenario 3	Deep learning network improved from Inception-V3.	97%
The proposed method with scenario 4	Deep learning network improved from DenseNet.	98%

8-bit micro-controller. It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, an ICSP header, and a reset button. It contains everything needed to support the

micro-controller. Buzzer module is used to emit sound when triggering the signal (PWM), applied in signaling systems, burglar alarms. The Logitech C270 HD Webcam gives crisp HD 720p/30 fps video with diagonal 55 degree field of view and auto light correction. The trained detection model through stages (face detection, emotion detection and doze detection) will be integrated into Jetson Nano as shown in Fig. 8. When starting up, the system takes image frames of the driver through the surveillance camera and then passes them through the doze detection model that has been integrated into the Jetson Nano to monitor. If the driver drowsiness is detected, a collision and a warning sound will be played to wake up the driver, meanwhile an email about location and time will be sent to notify the driver's status. Besides, if the system does not function properly, it is dangerous for drivers. Thus, the model is integrated a micro-controller board (Arduino Micro) to repair the system if problems exist. In Fig. 23, the system operation is described as follows:

- (1) The system takes image frames of drivers through a surveillance camera and then passes them through the drowsy detection model that has been integrated into the Jetson Nano to monitor.
- (2) The signals from Jetson nano are passed through an Arduino Micro board to perform a device health check. The Arduino Micro board is connected with a buzzer and a led to warn users when there is a hardware or software problem through lights and sounds.
- (3) Every 5 seconds, the Arduino micro will check the "health" of Jetson nano via the UART protocol (Universal asynchronous receiver-transmitter); The checking results will provide two statuses, normal or failed.
- (4a) If the Jetson nano does not respond or responds with an error code, this means that a device failure occurred. When there is a software problem, it will issue a led and a "beep beep" sound to warn and (4b) at the same time send an error code to the Arduino Micro to notify.
- (5) Based on the error code, the Arduino Micro can control the Jetson nano to restart or recover to fix the error itself. If the Jetson nano still fails, it will use the original docker images to create a new container to run.
- (6) In case the Jetson nano responds, the device is working properly and the system enters the drowsiness detection phase.
- (7a) When the driver is drowsy, the IoT module will issue an alert to the Cloud and from the Cloud will send an alert Email (8); Simultaneously emits a warning sound to wake the driver up (7b).

7 Conclusion

Currently, the rate of vehicle utilization is becoming more and more popular. As a result, the number of traffic accidents is also increasing. Many road accidents happen because of driver fatigue or drowsiness. This is a very serious problem causing in hundred thousands of road accidents each year. Therefore, it is necessary to have a warning system so that whenever the driver feels drowsy, the siren will activate and alert drivers. In this work, we address a drowsy driver alert system that has been developed based on multiple behavioral signs using IoT and deep learning techniques. We propose a deep learning based approach for drowsy detection and prediction of drivers by designing and perfecting four adaptive neural networks developed on LSTM, VGG16, InceptionV3, and DenseNet. We take advantage of the pre-trained networks, add some adaptive layers, and then apply the transfer learning approach to be able to appropriate to our research. This helps to shorten training time, avoid over-fitting, and improve the accuracy of drowsiness prediction. The proposed networks analyze the driver's signs of drowsiness and learn all the characteristics of the drowsy state. They take advantage of the deep learning neural networks to extract all drowsiness features to detect and predict the state of drowsiness accurately. In addition, we perform the experiments on four scenarios. Experimental results show that the training accuracy is up to 98% and these scenarios are feasible and suitable for the development of drowsiness warning applications. Moreover, we provide the test accuracy in the cases of with/without wearing a mask and glasses that have not been provided by previous studies. We also make a comparison of our methods and recent methods. It shows that the proposed networks could be advantageous because they can be trained faster and more efficiently, especially with limited hardware. As a result, scenario 4 gives the highest accuracy of drowsiness prediction compared to the remaining scenarios. Besides better parameter efficiency, a great advantage of the proposed network with scenario 4 is the connection of all layers and passes the feature-maps to all subsequent layers leading to deep supervision. This research demonstrates that the proposed method using deep learning techniques is useful and effective for monitoring drowsy drivers to provide accurate detection of drowsiness in various driving conditions. It can effectively improve the accuracy of drowsiness prediction and help solving real-life problems. Warnings due to drowsiness should be given early to avoid possible unfortunate traffic accidents. We have implemented and examined the effectiveness and feasibility of the system with IoT and machine learning techniques at center of Toyota Technical Education Program, Vinh Long University of Technology Education. Experimental results confirm that the proposed method is completely feasible and can be applied in practice with a high accuracy. In further research, we will deploy the proposed methods on a drowsiness warning system with the help of a voice speaking through an IoT module and the Raspberry Pi monitoring system. Besides, we will focus on analyzing data collected from EEG-ECG devices and applying big data techniques in detecting drowsiness for a centralized management system of buses in the city.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors are unable or have chosen not to specify which data has been used.

References

[1] NHTSA, https://www.nhtsa.gov/risky-driving/drowsy-driving.
[2] SleepFoundation, 2022, https://www.sleepfoundation.org/drowsy-driving.
[3] NSC, https://www.nsc.org/road/safety-topics/fatigued-driver.
[4] B.C. Tefft, Prevalence of Motor Vehicle Crashes Involving Drowsy Drivers, Technical Report, AAA Foundation for Traffic Safety, Washington, DC 20005, 2014.
[5] CDC, 2022, https://www.cdc.gov/sleep/features/drowsy-driving.html.
[6] NHTSA, Early Estimate of Motor Vehicle Traffic Fatalities for the First Quarter of 2021, National Highway Traffic Safety Administration, Washington, DC 20590, 2021, https://www.nhtsa.gov/sites/nhtsa.gov/files/2021-09/Early-Estimate-Motor-Vehicle-Traffic-Fatalities-Q1-2021.pdf.
[7] A.-C. Phan, N.-H.-Q. Nguyen, T.-N. Trieu, T.-C. Phan, An efficient approach for detecting driver drowsiness based on deep learning, Appl. Sci. 11 (18) (2021) 8441.
[8] W.W. Wierwille, S. Wreggit, C. Kirn, L. Ellsworth, R. Fairbanks, Research on Vehicle-Based Driver Status/performance Monitoring; Development, Validation, and Refinement of Algorithms for Detection of Driver Drowsiness. Final Report, Technical Report, US National Highway Traffic Safety Administration, 1994.
[9] N. Haworth, P. Rowden, Fatigue in motorcycle crashes: Is there an issue? in: 2006 Australasian Road Safety Research, Policing and Education Conference Proceedings, Able Video and Multimedia Pty Ltd, 2006, pp. 1–10.
[10] L.R. Hartley, Fatigue and Driving: Driver Impairment, Driver Fatigue, and Driving Simulation, CRC Press, 1995, section 2: the epidemiology of fatigue-related crashes.
[11] A. Čolić, O. Marques, B. Furht, Driver Drowsiness Detection: Systems and Solutions, Springer, 2014, pp. 7–18, http://dx.doi.org/10.1007/978-3-319-11535-1, chapter 2.
[12] M.K. Hussein, T.M. Salman, A.H. Miry, M.A. Subhi, Driver drowsiness detection techniques: A survey, in: 2021 1st Babylon International Conference on Information Technology and Science, BICITS, IEEE, 2021, pp. 45–51.
[13] J.Y. Wong, P.Y. Lau, Real-time driver alert system using raspberry pi, ECTI Trans. Electr. Eng. Electron. Commun. 17 (2) (2019) 193–203.
[14] A.L.A. Ramos, J.C. Erandio, D.H.T. Mangilaya, N.D. Carmen, E.M. Enteria, L.J. Enriquez, Driver drowsiness detection based on eye movement and yawning using facial landmark analysis, Int. J. Simul.–Syst. Sci. Technol. 20 (2019).
[15] S. Shivani, S. Aditya, V. Ramalingam, Driver drowsiness detection system using machine learning algorithms, Int. J. Recent Technol. Eng. (IJRTE) 8 (2020) 990–993.
[16] A.K. Biswal, D. Singh, B.K. Pattanayak, D. Samanta, M.-H. Yang, IoT-based smart alert system for drowsy driver detection, Wirel. Commun. Mob. Comput. 2021 (2021).
[17] N. Sharma, R. Sharma, N. Jindal, Machine learning and deep learning applications-A vision, Glob. Transit. Proc. 2 (1) (2021) 24–28.
[18] H. He, X. Zhang, F. Jiang, C. Wang, Y. Yang, W. Liu, J. Peng, A real-time driver fatigue detection method based on two-stage convolutional neural network, IFAC-PapersOnLine 53 (2) (2020) 15374–15379.
[19] Z. Zhao, N. Zhou, L. Zhang, H. Yan, Y. Xu, Z. Zhang, Driver fatigue detection based on convolutional neural networks using em-cnn, Comput. Intell. Neurosci. 2020 (2020).
[20] P.B. Venkata, C. Suchismitha, Automatic classification methods for detecting drowsiness using wavelet packet transform extracted time-domain features from single-channel EEG signal, J. Neurosci. Methods 347 (2021) 108927.
[21] H.V. Chand, J. Karthikeyan, CNN based driver drowsiness detection system using emotion analysis, Intell. Autom. Soft Comput. 31 (2) (2022) 717–728, URL: http://www.techscience.com/iasc/v31n2/44533.
[22] A. Rajkar, N. Kulkarni, A. Raut, Driver drowsiness detection using deep learning, in: Applied Information Processing Systems, Springer, 2022, pp. 73–82.
[23] A. Quddus, A. Shahidi Zandi, L. Prest, F. Comeau, Using long short term memory and convolutional neural networks for driver drowsiness detection, Accid. Anal. Prev. 156 (2021) 106107, http://dx.doi.org/10.1016/j.aap.2021.106107.
[24] V. Yarlagadda, S.G. Koolagudi, M. Kumar M V, S. Donepudi, Driver drowsiness detection using facial parameters and RNNs with LSTM, in: 2020 IEEE 17th India Council International Conference, INDICON, 2020, pp. 1–7, http://dx.doi.org/10.1109/INDICON49873.2020.9342348.
[25] F. Faraji, F. Lotfi, J. Khorramdel, A. Najafi, A. Ghaffari, Drowsiness detection based on driver temporal behavior using a new developed dataset, 2021, CoRR abs/2104.00125. URL: https://arxiv.org/abs/2104.00125. arxiv:2104.00125.
[26] S. Chaabene, B. Bouaziz, A. Boudaya, A. Hökelmann, A. Ammar, L. Chaari, Convolutional neural network for drowsiness detection using EEG signals, Sensors 21 (5) (2021) http://dx.doi.org/10.3390/s21051734.
[27] G. Geoffroy, L. Chaari, J.-Y. Tourneret, H. Wendt, Drowsiness detection using joint EEG-ecg data with deep learning, in: 29th European Signal Processing Conference (EUSIPCO 2021), Dublin, Ireland, 2021, pp. 955–959, URL: https://hal.archives-ouvertes.fr/hal-03381946.
[28] H. Kitajima, N. Numata, k. Yamamoto, Y. Goi, Prediction of automobile driver sleepiness. 1st report. Rating of sleepiness based on facial expression and examination of effective predictor indexes of sleepiness., Trans. Jpn. Soc. Mech. Eng. Ser. C 63 (1997) 3059–3066, http://dx.doi.org/10.1299/kikaic.63.3059.
[29] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[30] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[32] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[34] M.F. Hashmi, S. Katiyar, A.G. Keskar, N.D. Bokde, Z.W. Geem, Efficient pneumonia detection in chest Xray images using deep transfer learning, Diagnostics 10 (6) (2020) http://dx.doi.org/10.3390/diagnostics10060417.
[35] X. Li, Z. Dong, SizeNet: Object recognition via object real size-based convolutional networks, 2021, CoRR abs/2105.06188. URL: https://arxiv.org/abs/2105.06188. arxiv:2105.06188.
[36] A. Shazia, T.Z. Xuan, J.H. Chuah, J. Usman, P. Qian, K.W. Lai, A comparative study of multiple neural network for detection of COVID-19 on chest X-ray, EURASIP J. Appl. Signal Process. 2021 (1) (2021) 50, http://dx.doi.org/10.1186/s13634-021-00755-1.
[37] A. Mostafa, M.I. Khalil, H. Abbas, Emotion recognition by facial features using recurrent neural networks, in: 2018 13th International Conference on Computer Engineering and Systems, ICCES, 2018, pp. 417–422, http://dx.doi.org/10.1109/ICCES.2018.8639182.
[38] P.S. Kingman, B. Jesse, C. Forrest, G. Kate, K. James, K. Roger, T.M. Anne, L.M. Sharon, I.P. Allan, R. Susan, R. Thomas, S. Jane, W. Pat, W. David, Drowsy Driving and Automobile Crashes, NCSDR/NHTSA Expert Panel on Driver Fatigue and Sleepiness, National Highway Traffic Safety Administration, 1999.