Detecting Driver Sleepiness using Convolutional Neural Networks

Shaik Tousif [1], Abdul Saboor [2], Syed Saffwan Ahmed [3], Sumayya Begum [4]

[1, 2, 3] B.E. Student, [4] Assistant Professor,
Department of IT, Lords Institute of Engineering &Technology,
Hyderabad, Telangana, India.

Published in IJIRMPS (E-ISSN: 2349-7300), Volume 11, Issue 1, (January-February 2023)

Abstract

The development in computer vision has aided drivers in the form of automatic self-driving cars etc. The accidents are caused by driver's exhaustion and drowsiness about 20%. Its carriages a dangerous issue for which numerous methods were proposed. However, they are not appropriate for real-time implementation. The major encounters confronted by these approaches are forcefulness to handle dissimilarity in human face and lightning conditions. Our intention is to implement a smart operating system that can lower the rate of road accidents considerably. This method enables us to find driver's face features like eye closure percentage, eye-mouth aspect ratios, blink rate, yawning, head movement, etc. In this classification, the driver is uninterruptedly observed by using a webcam. The car driver's facial features along with the eye movements are observed using a cascade classifier. Eye images are pull out and fed to Custom designed Convolutional Neural Network for categorizing whether both left and right eye are closed. Based on the sorting, the eye closure score is considered. Upon finding that the driver is being detected drowsy that a high alarm will be raised.

Keywords: Data Augmentation, Deep Learning, CNN Drowsiness

1. Introduction

A lot of safeties associated with carefully driving scheme arrangements abridged the risk of four-wheeler accidents also researches represented fatigue to be a foremost cause of four- wheeler accidents. A reputed four – wheeler company announced an idea that entirely fatal misfortune (17%) would be accredited to drowsy car user. A lot of research shown by a company named Volkswagen AG stipulate that 5-25% of all fatal accidents are occurred by the snoozing of car driver. The absence of attentiveness harm direction-finding actions and decline immediate actions and reconsiderations demonstrated that sleepiness increases danger of clatter our project demands for a reliable smart chauffeur drowsiness detecting system. The purpose is to generate a smart analyzing technique to dodge highway fatal accidents. It could be achieved by detecting tiredness and simultaneously informing the driver so that he or she can avoid fatal accidents on roads. According to previous research conducted, chauffeur's the sleepiness can be noticed by observing the three major factors such as measurement based on vehicles as well behavioral and physiological. However, these researches couldn't be implemented in real time So, the proposed system is intended to used concepts of Deep Learning by implementing CNN. CNN suggests an on-screen and operative way to classify the car driver as sleepy or not sleepy efficiently

2. Literature Survey

A meek tactic to sensing eyes [5] analyze the measurement of the concentration variation in the eye area. However other movements such as eye blinking, continuously yawing, rotation of head is not considered. Hence this method is not applicable for real time scenario. The conclusion was made on enclosing the consumption of two-dimension Conventional Neural Network [2] for distinguishing movement. the three-dimensional-sequential association is ignored in this concept. It uses replicated datasets and movements of head is not considered. Three-dimensional and temporary evidence [8] were integrating the representations of movement directions. A deep residual CNN model [3] is used to categorize the movement of eyes. The CNN is functioned only on a solitary image-based technique. Google Net [9] emphases only on gaping and make use of prediction based on frame by frame. As a result, lots of memory would be consumed. The examination of rotation of head, eyes condition and mouth [4] is done. It does not consider the sequential dependencies. As a substitute of providing ordinary pixel values as data [1], the particular facial features are condensed. There is no explanation of feature Extraction process. The importance of sequential dependencies learning [10] was involved for all electrode and changes three-dimensional input movement. Electro Encephalo Graphy (EEG) signal have scuffle to produce computational algorithms for weariness finding. CNN with data growth [7] and combined datasets stretches a little inferior accurateness.

The primary expertise in finding of sleepiness apprehensions dimension of gestures of muscular, intellectual and cardiovascular state. The following expertise embraces approaches of manipulative whole chauffeur presentation from the vehicle designs. The third expertise fit to vision approaches as a noninvasive method to manage tiredness of driver. The average framework used a dataset for sleepy, and a framework is skilled so that opening abstracts eye movements.

The traditional method uses the positioning of facial appearances for snoozing finding. The worth of these features is known as a time gathering inside a sequential procedure. Weariness detecting procedures are aggressive and not commercially worthwhile. The technique of manipulating the entire deed of chauffeur from the transportation procedures do go well with tiny sleepy movements. The question with the typical framework is that it be subject to an indicator craft wink characteristic. This procedure contempt plentiful informative facial indication that detects sleepiness. Facial innovative features sensed using dlib library grieve from standardization problems.

3. Proposed System

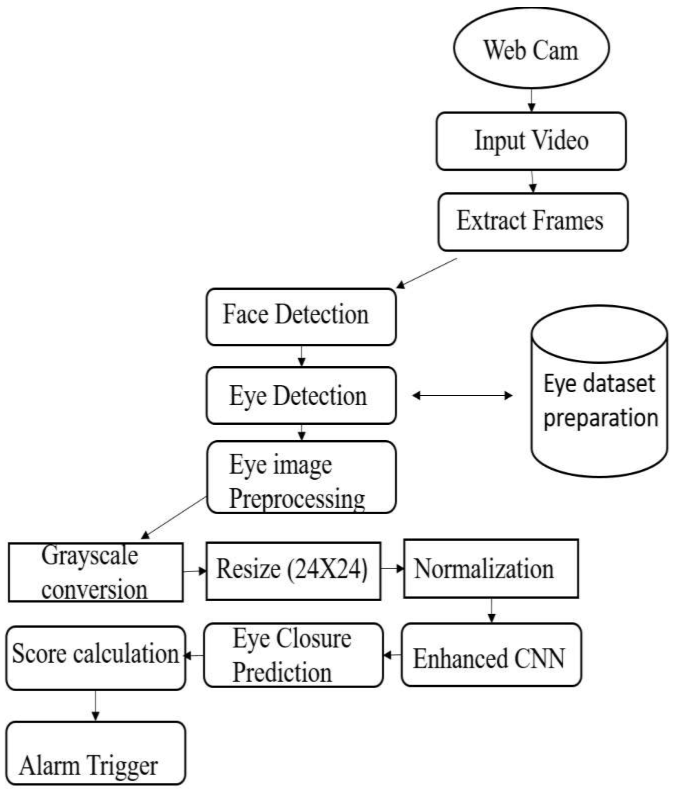
The proposed system will deliver a method for detecting chauffeur's sleepiness. The disadvantages of the old method in pulling out only particular hand-crafted characteristics are overcomed by providing an input of car driver image in custom-designed CNN. Because of this the chauffeur will be endlessly observed by a webcam. The video taken is transformed into a categorization of frames. For each frame, the face and eye are detected using predefined classifiers accessible in OpenCV called Haar cascade classifiers. Eye images are extracted and sent to a series of 2-D CNN layers (5×5, 3×3) kernel valid padding, max-pooling layers (2×2) and finally, the entirely linked condensed layer classifies whether eyes are closed or not. A score is considered built on eye closing movements. The improved CNN complete a computerized and operative cultured characteristic that help us to classify the movements of eyes. After 15 consecutive frames of eye closure is happening then an alarm will be raised to alert and notify the driver. The classification of chauffeur sleepiness is done appropriately and the standardization problems in the existing prototypical are eradicate via custom- designed CNN.

A. System Architecture

Figure 1 portrays the flow of the scheme to be created. In the preliminary step, the chauffeur is observed by using a webcam. After that the video input is transformed into a categorization of frames. In each frame, chauffeur's face and eyes are perceived by using Haar cascade classifiers. The perceived eyes are kept as images to formulate a data set for CNN. The scheme also delivers establishment for the preparation of the eye data set to train CNN model. To train the image and to increase the number of data sets Data augmentation is done. The pictures of right and left eyes are then subjected to a series of image pre-processing steps such as grayscale conversion, re-sizing and normalizing, etc. the next step is those images are now given to a pre-trained CNN model consisting of convolution layers, max- pooling layers, and dense layers to predict eye closure. As per calculation, a score is premeditated. If the proposed system detects the chauffeur as sleepy, then an alarm will be activated to vigilant the chauffeur.

B. Implementation Description

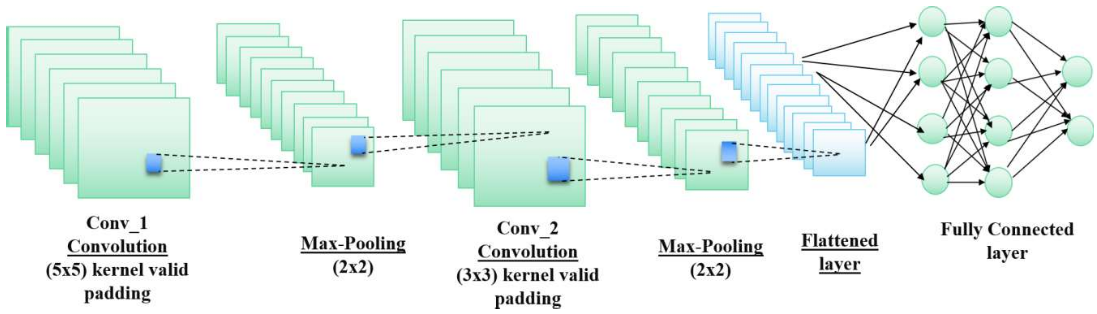
The components are Face and Eye Detection, Data Augmentation, Enhanced CNN, and Triggering Alarm, Pre-processing and Labeling.



OpenCV delivers pre-trained models in the face and eye detection parts via a load method.

The pre-trained forms are positioned in the folder of OpenCV evidence. The proposed system will use previous trained Haar models to detect the eye images. Cascade Classifier is molded and heaps the vital XML file by means of its method. The next step is the multi-scale method delivers bound rectangle for the experimental eyes. Our dataset is being prepared by pre-processing and labeling the eyes images found through a webcam as both the eye movements open and closed. The pull out eye images were rehabilitated into grayscale images, resized to 24 × 24 pixels, and normalized. No new data is being gathered for Data augmentation module, already present data id being transformed. We achieve data augmentation using Keras in which it receives the ranges for rotation, brightness, shear, zoom, etc. as parameters. Each of CNN layers has various parameters that can be altered and implement various tasks on the input data as shown in Figure 2 ReIU activation function in convolutional layers with kernel size of 3 × 3. Softmax activation function is utilized in fully connected layers to give the consequence as either open or closed eyes.

Figure 2: Improved CNN Architecture



Our Technique uses Pygame library to provide indications to car driver while he or she is starting to fall asleep. The grade point or a score value has been found depending upon how long a chauffeur is closing an eye. The score would be increases If the driver both eyes are closed, and score would be decreased when eyes are open. Our proposed system conscripting the consequence to show the definite time state of the chauffeur. The prototypical summary facts the layers involved in CNN, input shape, output shape, and several trainable parameters as shown in Figure 3. Dropout, dense and activation layers are extra to create learning effectiveness. This system is accomplished in 10 epoch's, training accuracy and losses as portrayed in Figure 4.

Figure 3: Summary of Improved CNN Model

```
In [12]:  model.summary()
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 24, 24, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 24, 24, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 24, 24, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 24, 24, 64)        18496
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 24, 24, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 24, 24, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 36864)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               9437440
_________________________________________________________________
activation_1 (Activation)    (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 514
=================================================================
Total params: 9,466,594
Trainable params: 9,466,594
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: Outcomes of Improved CNN

```
In [17]: History = model.fit_generator(datagen.flow(x_train,y_train, batch_size=batch_size),
                                       epochs = epochs, validation_data = (x_test,y_test),
                                       verbose = 1, steps_per_epoch=x_train.shape[0] // batch_size)

Epoch 1/10
187/187 [==============================] - 78s 419ms/step - loss: 3.3121 - accuracy: 0.6592 - val_loss: 0.4767 - val_accurac
y: 0.7720
Epoch 2/10
187/187 [==============================] - 75s 400ms/step - loss: 0.4647 - accuracy: 0.7842 - val_loss: 0.5745 - val_accurac
y: 0.7150
Epoch 3/10
187/187 [==============================] - 75s 403ms/step - loss: 0.6765 - accuracy: 0.5834 - val_loss: 0.6164 - val_accurac
y: 0.7560
Epoch 4/10
187/187 [==============================] - 75s 402ms/step - loss: 0.6637 - accuracy: 0.5942 - val_loss: 0.6605 - val_accurac
y: 0.5700
Epoch 5/10
187/187 [==============================] - 75s 401ms/step - loss: 0.6401 - accuracy: 0.6418 - val_loss: 0.4994 - val_accurac
y: 0.8010
Epoch 6/10
187/187 [==============================] - 75s 400ms/step - loss: 0.5622 - accuracy: 0.7162 - val_loss: 0.2739 - val_accurac
y: 0.9070
Epoch 7/10
187/187 [==============================] - 75s 401ms/step - loss: 0.4265 - accuracy: 0.8190 - val_loss: 0.2794 - val_accurac
y: 0.8710
Epoch 8/10
187/187 [==============================] - 75s 401ms/step - loss: 0.2384 - accuracy: 0.9209 - val_loss: 0.1446 - val_accurac
y: 0.9630
Epoch 9/10
187/187 [==============================] - 74s 398ms/step - loss: 0.1587 - accuracy: 0.9601 - val_loss: 0.0932 - val_accurac
y: 0.9760
Epoch 10/10
187/187 [==============================] - 74s 395ms/step - loss: 0.1139 - accuracy: 0.9729 - val_loss: 0.0321 - val_accurac
y: 0.9910
```

4. Performance Measures

The examination of the appearance finding methods is shown in Table 1.

Table 1: Face Recognition Systems Investigation

Techniques	Features Count	Dataset	Accuracy
Geometric	38400	47	90
Mixture-Distance	23800	685	94
Eigen Faces	26400	860	95
CV_DNN	22500	880	97.05
Enhanced DNN	30800	1000	99.10

The precision swiftness of Figure 5 and loss of Figure 6 is faster than expectable training. This paper uses amplified database in order to get train and for open and close eye images it uses original database that is available for testing.

Figure 5: Accuracy of Model Improved CNN

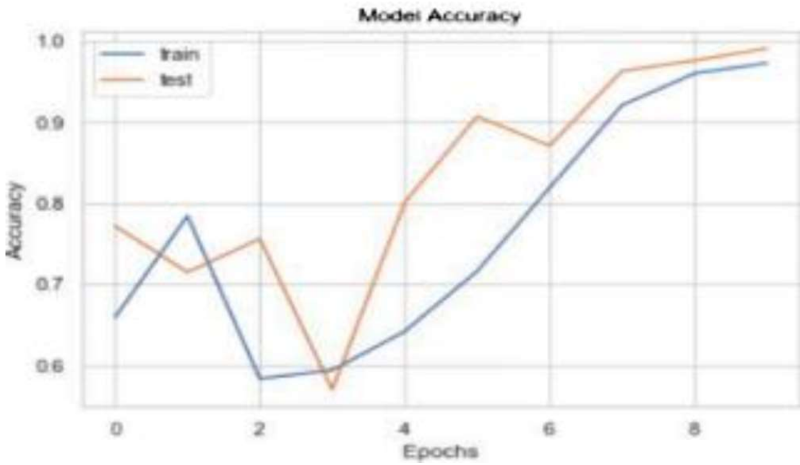


Figure 6: Loss of Model Improved CNN

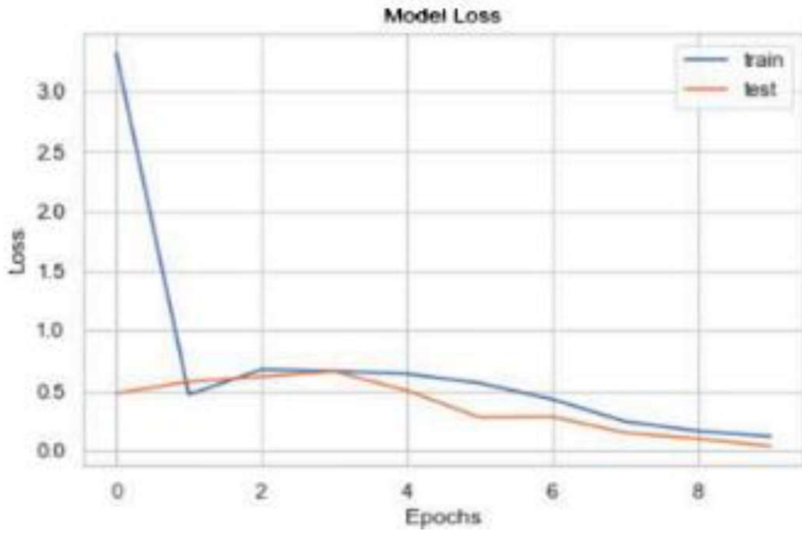


Table 2 shows the count of pictures in closing the eye is less than the count of opening the eye.

Table 2: Databases - Original and Augmented Images

Image Types	Original Database	Augmented Database
Eye Opened	4563	24680
Eye Closed	3867	20430

5. Testing

Once the CNN is being trained, we have started testing with a dataset comprising of 1000 images of which indicates both open eye and close eye. A sorting statement is produced as shown in Table III. This paper suggested sleepiness detection technique by parameters. Precision is nothing but the amount of actual affirmative considerations to the entire count of considerations.

Table 3: Organization Statement

Average	Precision	Recall	f1-score	Support
0.0	1.0	0.34	0.51	500
1.0	0.60	1.00	0.75	500
Accuracy	-	-	0.67	1000
Macro Avg.	0.80	0.67	0.63	1000
Weighted Avg.	0.80	0.67	0.63	1000

Table 4 displays the investigation completed on ResNet, AlexNet, VGGNet and ProposedNet of images for Epoch-10.

Table 4: Epoch-10 Precision

Techniques	Epoch-10
ResNet	20.42
AlexNet	23.17
VGGNet	20.85
ProposedNet	11.39

The confusion matrix of real and untrue-positive ratings:
Array([[169, 331], [0, 500]], dtype = int64)

The Receiver Operating Characteristic curve is plotted as shown in Figure 7 denote testing information with the suggested technique built on the quantity of training periods.

Figure 7: ROC Curve of Improved CNN

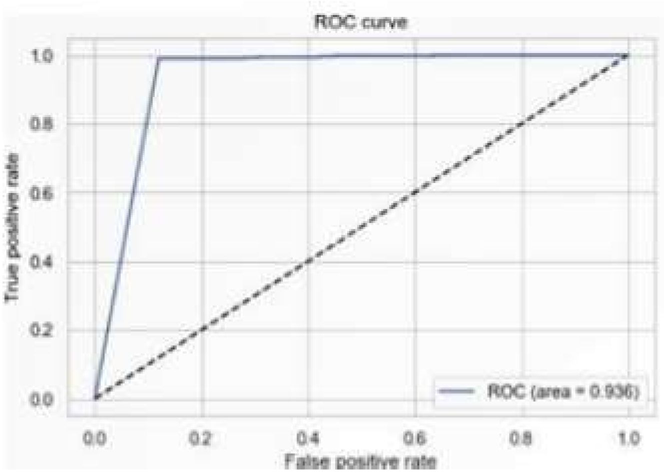


Figure 8 Shows the face and eye identification using Haar classifier.

Figure 8: Detection of Face and Eye

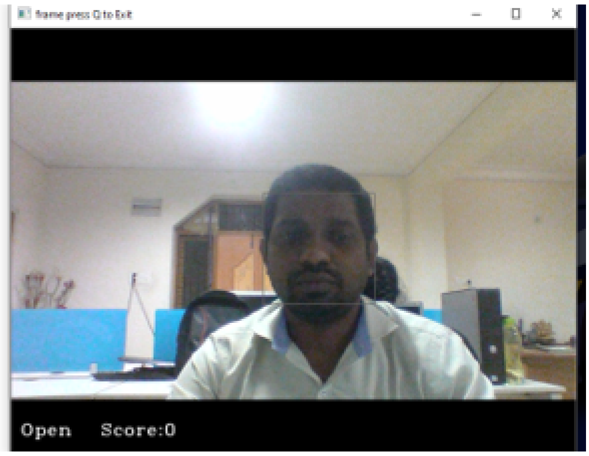


Figure 9 shows that both right and left eyes are fully open. Hence, it is concluded that the person is in alert state.

Figure 9: Alert State

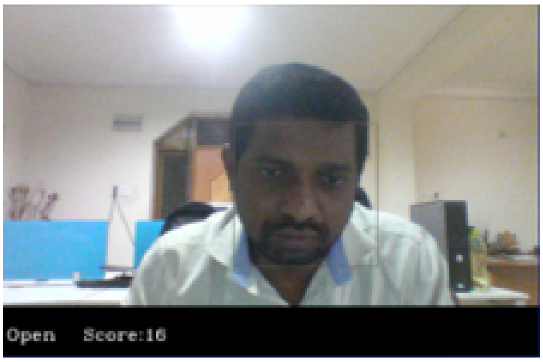
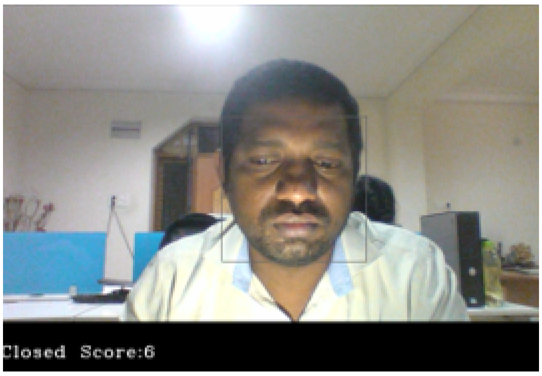


Figure 10 shows that both right and left eyes are partially closed. Hence we can say that the person is in drowsy state.

Figure 10: Drowsy State



6. Conclusion

A prototypical model for sleepiness recognizing be contingent on operative CNN architecture, strategic to detect sleepiness created on eye closing. The execution started formulating image datasets for both open and closed eyes. 75% of the data set is used for the tradition-considered CNN training and the sense of balance 25% of the dataset is exploited for test determinations. First, the data video is altered into frames and in each frame, the face and eyes are distinguished. The improved CNN complete a computerized and operative cultured characteristic that help us to classify the movements of eyes. After 15 consecutive frames of eye closure is happening then an alarm will be raised to alert and notify the driver. The anticipated CNN provide an accuracy rate of about 97% and a testing precision of 67%.

References

[1] Priya Gupta, Nidhi Saxena, Meetika Sharma, Jagriti Tripathi, "Deep Neural Network for Human Face Recognition", International Journal of Engineering and Manufacturing, vol. 8, no. 1, pp. 63-71, January 2018.

[2] Jeyasekar A., Vivek Ravi Iyengar, "Driver Drowsiness Detection Based on Behavioural Changes using ResNet", International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 3, pp. 25-30, 2019.

[3] Mandalapu Sarada Devi, Preeti R. Bajaj, "Driver Fatigue Detection Based on Eye Tracking", First International Conference on Emerging Trends in Engineering and Technology, vol. 1, pp. 649-652, 2008.

[4] Sanghyuk Park, Fei Pan, Sunghun Kang, Chang D. Yoo, "Driver drowsiness detection system based on feature representation learning using deep networks", Springer International Publishing, Computer Vision ACCV 2016 Workshops.

[5] Tawsin Uddin Ahmed, Sazzad Hossain, Mohammad Shahadat Hossain, Raihan Ul Islam, Karl Andersson, "Facial Expression Recognition using Convolutional Neural Network with Data Augmentation 8th International Conference on Informatics, Electronics & Vision (ICIEV), Washington, USA, 2019.

[6] Upasana Sinha, Kamal K. Mehta, A.K. Shrivastava, "Real Time Implementation for Monitoring Drowsiness Condition of a Train Driver using Brain Wave Sensor", International Journal of Computer Applications, vol. 139, no. 9, pp. 25-30, 2016.

[7] Xiaoxi Ma, Lap-Pui Chau, Kim-Hui Yap, "Depth Video-based Two- stream Convolutional Neural Networks for Driver Fatigue Detection", IEEE International Conference on Orange Technologies (ICOT), pp. 155-158, 2017.

[8] Weiwei Zhang, Jinya Su, "Driver Yawning Detection based on Long Short Term Memory Networks", IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 2017.

[9] Zhongke Gao, Xinmin Wang, Yuxuan Yang, Chaoxu Mu, Qing Cai, Weidong Dang, Siyang Zuo "EEG-based Spatial-Temporal Transactions on Neural Networks and Learning Systems", IEEE, vol. 30, no. 9, pp. 2755-2763, 2019.