International Journal of Advanced Research and Review

www.ijarr.org

The Use of Convolutional Neural Networks for the Monitoring of Driver Drowsiness

[1] S. Venkateswara Rao, [2] V.Deepthi,

[1] Assistant Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.
[2] MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

Abstract

Automated self-driving automobiles, etc., are a result of computer vision advancements that aid drivers. About 20% of accidents occur because drivers are too sleepy or exhausted to react properly. A number of solutions have been suggested in response to this major issue. On the other hand, real-time processing is not something they excel at. These approaches suffer from a lack of robustness when it comes to dealing with lighting circumstances and variations in human faces. Our goal is to install a smart processing system that will significantly cut down on traffic accidents. The method allows us to detect the driver's facial features, such as the frequency of blinking, the angle of the eyes relative to the lips, the amount of yawning, the amount of head movement, and the proportion of closed eyes. A camera is used in this system to continually observe the driver. When a motorist looks into the camera, haar cascade classifiers can identify their face and eyes. To determine whether the eyes are closed or open, we use a custom-designed convolutional neural network to categorize photos of the eyes. It is the categorization that determines the ocular closure score. There will be an alert that goes off if the driver is determined to be too sleepy.

Keywords

Network for Convolutional Neural Systems; Data Augmentation; Learning for Deep Learning; Fatigue;

I. INTRODUCTION

Investigations revealed fatigue to be a key cause of four-wheeler accidents, and several safety-connected driving assistance programs reduced the likelihood of such incidents. The assumption that sleepy drivers were responsible for one-fifth of all fatal accidents was proposed by an automotive group. According to many revisions shown by Volkswagen AG, driver sleepiness is the cause of 5-25% of all accidents. A trustworthy intelligent driver sleepiness monitoring system is necessary since inattention impairs steering actions and slows reaction time, and updates have shown that drowsiness increases the risk of accidents. The end goal is to develop a system of intelligent processing that may prevent traffic accidents. A length of time spent monitoring the driver's level of sleepiness may help achieve this goal and alert them when they are becoming too distracted to drive safely. Three criteria, including physiological, behavioral, and vehicle-based measures, may be used to identify driver tiredness, according to the literature review. However, in certain real-time situations, these methods do have significant drawbacks. Consequently, we want to tackle this issue statement by using Deep Learning techniques. A CNN is the tool we'll be using in our technique. Computerized neural networks (CNNs) provide a reliable method for accurately classifying drivers as sleepy or not.

II. LITERATURE SURVEY

One easy way to find eyes is to measure the duration of the intensity change in the area around the eyes [5]. Other physiological parameters like blink rate, yawning, head movement, etc. are not included. Processing in real-time is not possible with this. The use of a two-dimensional convolutional neural network (CNN) [2] for distinctive activity was decided upon. The spatial-temporal link is ignored. It ignores characteristics like head movement in favor of simulated datasets. In order to classify driver sleepiness, the characteristic state learning conceptualizations [6] offered a useful learnt feature. It extracts features using a complicated framework. The hand-operated tagging of the chosen photo series was done on the video data set. The representation of movement vectors was being included into spatial and transient data [8]. For the purpose of classifying eye movement, a deep residual CNN model [3] is used. A single image-based approach is used to run the CNN. Using frame-by-frame prediction, Google Net [9] zeroes focused on yawning as its only target. This means that a lot of RAM is needed. The status of the head, lips, and eyes are examined [4]. It disregards the interdependencies throughout time. Instead of using raw pixel values as input, the chosen face features are shown. There is a lack of clarity on the feature extraction technique. All electrodes and changes in spatial input activity were linked to the significance of learning about time relationships [10]. The development of computer techniques for the identification of weariness using electroencephalography (EEG) signals has been a challenge. The accuracy of CNN with information increase [7] and connected datasets is somewhat worse than that of earlier methods. Measuring signals of muscle, brain, and cardiovascular states is the first technological advancement in the identification of sleepiness. The following technological development relates to techniques for deriving overall driver performance metrics from vehicle blueprints. Thirdly, vision approaches provide a noninvasive way to monitor driver sleepiness. By taking the duration, opening speed, and amplitude of each eye blink into account, the typical framework is trained to first extract drowsy-related traits from a dataset. For drowsiness detection, the current approach makes advantage of the orientation of face features. A time ordering within a successive form describes the importance of these features. Unfortunately, fatigue detection methods are both intrusive and economically unfeasible. When there is little sleep, the approach of determining the driver's overall activity from the transportation forms fails. The traditional framework has a flaw in that it is dependent on the hand-crafted wink feature. This task disregards a plethora of informative face cues that influence lethargy. The dlib library has problems with normalizing facial landmark characteristics.

III. PROPOSED SYSTEM

An answer to the problem of keeping tabs on drivers' sleepiness is our suggested approach. By feeding an input driver picture into a custom-designed CNN, the current system's drawback of extracting just chosen hand-crafted features is circumvented. Currently, a camera will be always watching the driver. The recorded video is then rendered as a series of still images. Using openCV's default classifiers, namely the haar cascade classifiers, we can identify the face and the eye in every frame. In order to determine whether the eyes are closed or not, the pictures of the eyes are first extracted and then sent through a sequence of 2D convolutional neural network (CNN) layers, including max-pooling layers (2x2), 2x5 kernel valid padding, and lastly, a fully connected dense layer. Eye closure is used to calculate a score. If the driver's eyes stay closed for 15 frames in a succession, the system will interpret it as drowsiness and raise a warning. Using a custom-designed CNN, we are able to accurately classify driver tiredness and remove the normalization difficulties in the old model. A. Architecture of the System The proposed system's flow is shown in Figure 1. First, a camera is used to keep an eye on the driver. The input video is transformed into a series of frames. The driver's eyes and face are identified in every frame by using haar cascade classifiers. Images of the identified eyeballs are added to a CNN data set. In addition, the system includes tools to help get the ocular data set ready for CNN model training. In order to train the picture and expand the dataset Data is supplemented. Next, a number of picture preparation operations are applied to the two eye images, including grayscale conversion, resizing, normalizing, and so on. After that, it is input into a convolutional neural network (CNN) model that has already been trained to anticipate when the eyes would close using convolution layers, max-pooling layers, and dense layers. A grade is determined by using the forecast. The device will sound an alarm to wake the driver up if it detects that they are too sleepy. B. Example of an Execution Face and eye detection, data augmentation, enhanced convolutional neural networks (CNN), preprocessing and labeling, and triggers are the components involved. Red flag. Using a load approach, OpenCV's Face and eye identification module delivers pre-trained models.

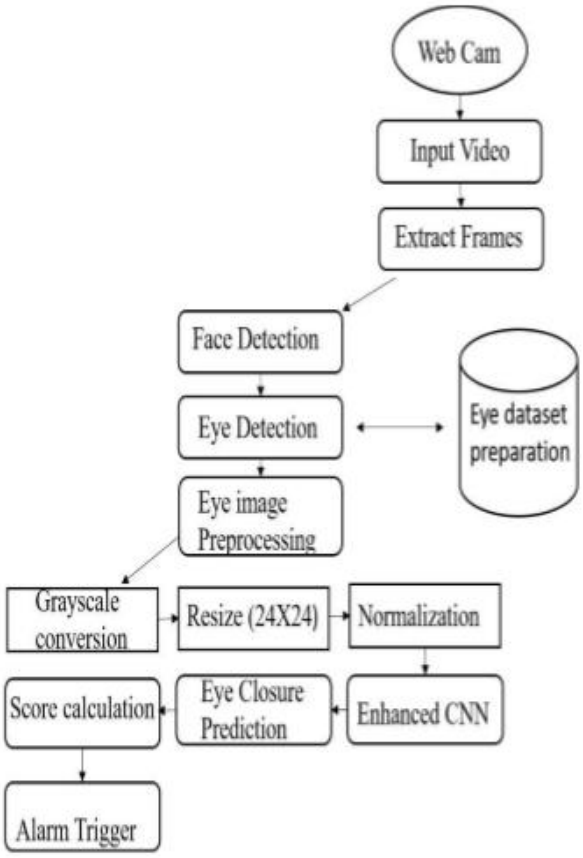

Fig. 1 System Architecture of CNN

In the OpenCV data folder, you may find the pre-trained forms. After training Haar models, the algorithm will analyze the eye image. After forming, Cascade Classifier uses its way to load the necessary XML file. After that, the observed eyeballs are given a bound rectangle using the multiscale approach. In order to get our dataset ready, we labeled and preprocessed webcam photos of people's eyes as either open or closed. After being extracted, the pictures of the eyes were normalized, reduced to 24x24 pixels, and turned into grayscale. For the Data augmentation module, we transformed the existing data rather than collecting new data. Applying Keras, which takes as parameters the rotation, brightness, shear, zoom, etc. ranges, allows us to do data augmentation. Figure 2 shows that each CNN layer contains a number of parameters that may be used to modify the input data and perform different tasks. We have implemented the relu activation function in 3x3 kernel size convolutional layers. An output of "open eyes" or "closed eyes" is produced by using the Softmax activation function in completely linked layers. The adam optimizer is used to train the CNN.

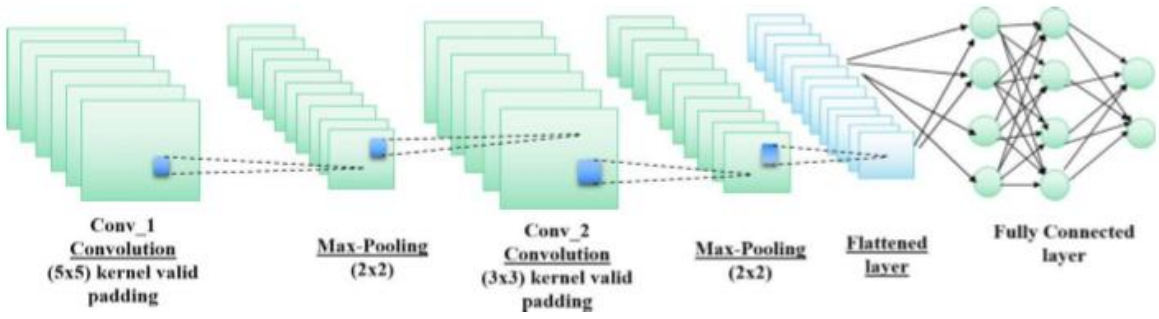

Fig 2. Enhanced CNN Architecture

While initiating sleep, we alert the driver using the Pygame library. To determine the duration of the driver's eye closure, the score value is used. When both eyes are closed, the score goes up, and when they're open, the score goes down. The result will show the driver's current time status, which we are currently working on. Layers are described in the model summary.

```
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
```

Fig 3. Model summary of Enhanced 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
```

Fig. 4. Output of Enhanced CNN

participation in convolutional neural networks (CNN), input/output shape, and a number of trainable parameters (Figure 3). Additional levels for creating successful learning are dropout, dense, and activation. As shown in Figure 4, this system is completed after 10 epochs, with respect to training accuracy and losses.

IV. PERFORMANCE MEASURES

The analysis of the face detection schemes is shown in table. I.

TABLE I. FACE DETECTION SCHEMES ANALYSIS

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

Figure 5's accuracy and Figure 6's loss are faster than predicted training. For the purpose of applying open and closed eye pictures, we trained using the augmented database and tested using the original database.

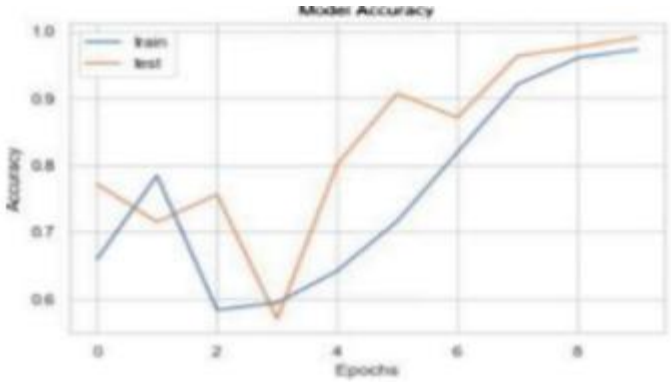

Fig. 5. Model accuracy of Enhanced CNN

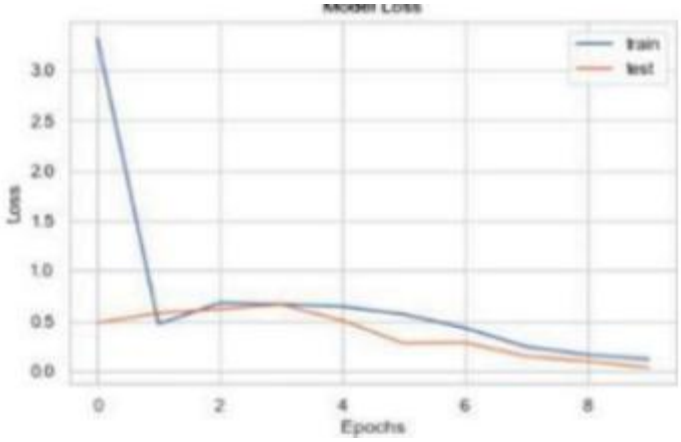

Fig. 6. Model loss of Enhanced CNN

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

TABLE II . DATABASES - ORIGINAL AND AUGMENTED IMAGES

Image Types	*Original Database*	*Augmented Database*
Eye opened	4563	24680
Eye closed	3867	20430

V. TESTING

Once the CNN has been trained, it is tested using a dataset that includes 1000 photos of open and closed eyes. Table III displays the results of the created categorization report. We use parameters to estimate the proposed method for sleepiness recognition. The accuracy rate is defined as the percentage of correct interpretations relative to the total number of interpretations. The percentage of correctly predicted interpretations relative to total interpretations in actual eye yawn is called recall. The f1-score is the average of the recall and precision.

TABLE III: CLASSIFICATION REPORT

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

TABLE IV: EPOCH-10 ACCURACY

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

The false-positive and real-life confusion matrix: Array of data type int64, consisting of elements from 169,331 to 500. Testing data using the proposed method based on the amount of training sessions is shown by the Receiver Operating Characteristic curve in Figure 7.

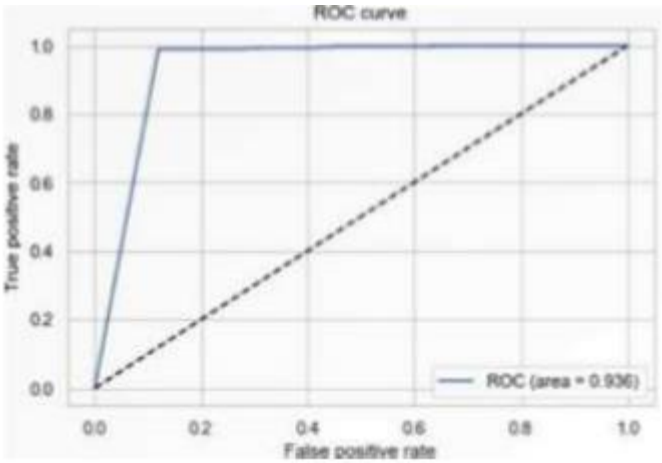

Fig. 7. ROC curve of Enhanced CNN

Figure 8 shows the haar classifier's face and eye sensing in action.

Fig. 8.Detection of face and eye

It is seen in Figure 9 that both eyes are open. Consequently, they are on high alert.

Fig. 9. Alert state

Both eyes are closed, as seen in Figure 10. As a result, they are in a condition of sleepiness.

Fig. 10. Drowsy state

VI. CONCLUSION

An efficient convolutional neural network (CNN) architecture is crucial to a sleepiness sensing model that aims to detect tiredness via eye closure. The process of creating picture datasets for open and closed eye scenarios began with the installation. Training the custom-designed CNN uses 75% of the dataset, while the remaining 25% is utilized for other purposes. The dataset is used for testing using 25% of it. To begin, the video data is split into individual frames, and then, in each of those frames, the eyes and face are identified. In order to classify eye openings and closings, the improved CNN provided an automatic and effective learning characteristic. An alarm will sound to notify the motorist if their eyes close for fifteen consecutive frames. With the suggested CNN, we can get a 97% training accuracy and a 67% testing accuracy. To improve detection accuracy in future studies, other facial attributes may be incorporated. The face traits that were retrieved may also be combined with data on the vehicle's driving patterns that were collected from the On-Board Diagnostics sensors.

REFERENCES

[1] Dr. 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, Based on Behavioural Changes using ResNet , International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 3, pp. 2 5-30, 2019.

[3] Conference (IACC), Gurgaon, pp. 995-999, 2014.

[4] Ki Wan Kim, Hyung Gil Hong, Gi Pyo Nam and Kang Ryoung Park, . [5] Luigi Celona, Lorenzo Mammana, Simone Bianco, Raimondo Schettini, -Task CNN Framework for Driver Face

Berlin, 2018.

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

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

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

[9] 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.

[10] Xiaoxi Ma, Lap-Pui Chau and Kim- -based Two- stream Convolutional Neural Networks for Driver Fatigue IEEE International Conference on Orange Technologies (ICOT), pp. 155-158, 2017.

[11] Weiwei Zhang, Jinya Su Driver Yawning Detection based on Long , IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA,2017.

[12] Zhongke Gao , Xinmin Wang, Yuxuan Yang, Chaoxu Mu , Qing Cai, Weidong Dang , and Siyang Zuo EEG-Based Spatial-Temporal , IEEE Transactions on Neural networks and learning systems, vol. 30, no. 9, pp. 2755-2763, 2019.