Driver Drowsiness Monitoring using Convolutional Neural Networks

Dr. D. Rosy Salomi Victoria
Associate Professor in CSE Department,
St. Joseph's College of Engineering,
Chennai, Tamil Nadu, India
drosysalomi@gmail.com

Dr. D. Glory Ratna Mary
HOD in Computer Science Department,
Women's Christian College,
Nagercoil, Tamil Nadu, India
gloryrmary@gmail.com

2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) | 978-1-7281-8529-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCIS51004.2021.9397070

***Abstract*—The advancement in computer vision has assisted drivers in the form of automatic self-driving cars etc. The misadventure are caused by driver's fatigue and drowsiness about 20%. It poses a serious problem for which several approaches were proposed. However, they are not suitable for real-time processing. The major challenges faced by these methods are robustness to handle variation in human face and lightning conditions. We aim to implement an intelligent processing system that can reduce road accidents drastically. This approach enables us to identify driver's face characteristics like eye closure percentage, eye-mouth aspect ratios, blink rate, yawning, head movement, etc. In this system, the driver is continuously monitored by using a webcam. The driver's face and the eye are detected using haar cascade classifiers. Eye images are extracted and fed to Custom designed Convolutional Neural Network for classifying whether both left and right eye are closed. Based on the classification, the eye closure score is calculated. If the driver is found to be drowsy, an alarm will be triggered.**

Keywords—Convolutional Neural Network; Data Augmentation;Deep Learning; Drowsiness;

I. INTRODUCTION

Many safety connected driving supporter schemes decreased the danger of four-wheeler accidents, and investigations depicted weariness to be a major reason of four-wheeler accidents. A car organization announced an idea that whole deadly accidents (17%) would be attributed to weary drivers. Many revisions showed by Volkswagen AG specify that 5-25% of all accidents are produced by the sleeping of driver. The lack of concentration damage steering actions and decrease response period, and revisions illustrated that sleepiness raises threat of crashes. This figure point the demand for a dependable intelligent driver sleepiness sensing system. The aim is to create an intelligent processing scheme to avoid road accidents. This can be done by period of time monitoring the drowsiness and warning driver of inattention to prevent accidents. Based on the literature survey, the driver's drowsiness can be detected based on three factors such as physiological, behavioral, and vehicle-based measurements. But these approaches pose some disadvantages in certain real-time scenarios. So, we aim to apply Deep Learning algorithms to this problem statement. Our methodology is to use a Convolutional Neural Network (CNN). CNN offers a computerized and effective way to categorize the driver as drowsy or non-drowsy correctly.

II. LITERATURE SURVEY

A simple approach to detecting eyes [5] calculate the length of the intensity alteration in the eye region. It does not take into account other physiological factors such as blink rate, yawning, head movement, etc. It is not suitable for real-time processing. The decision was made on framing the utilization of two dimension CNN [2] for characteristic activity. It does not take into account the spatial-temporal relationship. It uses simulated datasets and does not consider features such as head movement. The characteristic state learning conceptualizations [6] provided an effective learned characteristics to categorize the driver drowsiness. It uses a complex structure for feature extraction. The selected picture series the rating of video data set were tagged hand-operated.

Spatial and impermanent information [8] were incorporating the represention of movement vectors. A deep residual CNN model [3] is used to classify the opening or closing of the eyes. The CNN is operated only on a single image-based method. Google Net [9] focuses only on yawning and uses frame by frame prediction. So, it requires a lot of memory. The analysis of head position, mouth and eyes condition [4] is done. It does not take into account the temporal dependencies. As an alternative of supplying natural pixel values as information [1], the selected facial characteristics are rendered. The feature Extraction procedure is not explained clearly. The importance of temporal dependencies learning [10] was attached for all electrode and changes spatial input activity. Electro Encephalo Graphy (EEG) signal have struggle to create computational algorithms for fatigue detection. CNN with information increase [7] and joined datasets gives a slightly lower accuracy when compared to previous approaches.

The first technology in detection of drowsiness concerns measurement of signals of muscular, cerebral and cardiovascular state. The next technology comprises methods of calculating total driver performance from the automobile designs. The third technology belong to vision methods as a noninvasive manner to supervise drowsiness of driver. The standard framework used a Data set for drowsy and a framework is trained so that initial extracts eye blink

ISBN: 978-1-7281-8529-3/21/$31.00 ©2021 IEEE

connected characteristics, considering eye blink period, opening speed and amplitude. The existing system uses the orientation of facial characteristics for drowsy detection. The value of these characteristics is known as a time ordering inside a successive form. Fatigue sensing techniques are invasive and not commercially viable. The method of calculating the whole action of driver from the transport forms do not work with small sleeping. The issue with the standard framework is that it depends on hand craft wink characteristic. This activity disregard numerous enlightening facial evidence that affect drowsy. Facial landmark features detected using dlib library suffer from normalization issues.

III. PROPOSED SYSTEM

Our proposed system will provide a solution for monitoring driver's drowsiness. The cons of the existing system in extracting only selected hand-crafted features is overcome by using custom-designed CNN by giving an input driver image. Now the driver will be continuously monitored by a webcam. The video captured is converted into a sequence of frames. For each frame, the face and eye are detected using predefined classifiers available in opencv called haar cascade classifiers. Eye images are extracted and sent to a series of 2D CNN layers (5x5, 3x3 kernel valid padding), max-pooling layers(2x2) and finally, the fully connected dense layer classifies whether eyes are closed or not. A score is calculated based on eye closure. If both eyes are closed consecutively in 15 frames then the system predicts as drowsy and an alarm sound is triggered to alert the car operator. The categorization of driver drowsiness is done correctly and the normalization issues in the existing model are eliminated by using custom-designed CNN.

A. System Architecture

Figure 1 depicts the flow of the system to be created. In the initial step, the driver is monitored by using a webcam. The video input is converted into a sequence of frames. In each frame, driver's face and eyes are detected by using haar cascade classifiers. The detected eyes are stored as images to form a data set for CNN. The system also provides provision 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 images of both eyes are then subjected to a series of image preprocessing steps such as grayscale conversion, re-sizing and normalizing, etc. It is then fed to a pre-trained CNN model consisting of convolution layers, max- pooling layers, and dense layers to predict eye closure. Based on the prediction, a score is calculated. If the system finds the driver as drowsy, then an alarm will be triggered to alert the driver.

B. Implementation Description

The modules are Face and eye detection, Preprocessing and labeling, Data Augmentation, Enhanced CNN, and Triggering Alarm. OpenCV provides pre-trained models in the Face and eye detection module, using a load method.

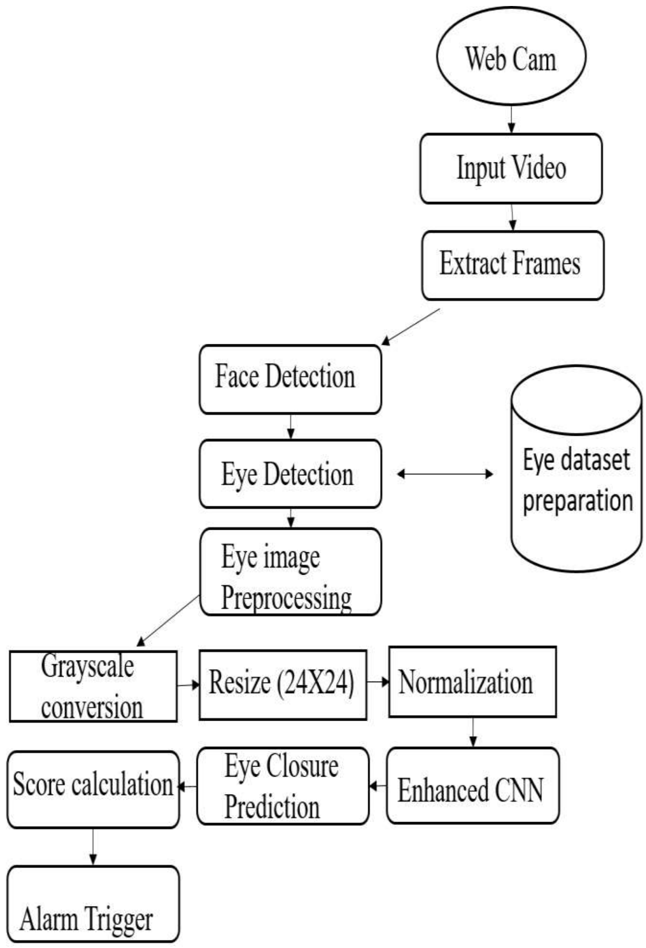

Fig. 1 System Architecture of CNN

The pre-trained forms are placed in the folder of OpenCV information. The system will use prior trained Haar models to observe the picture of eyes. Cascade Classifier is formed and loads the essential XML file using its method. Afterward, the multiscale method provides bound rectangle for the observed eyes. We prepared our dataset by preprocessing and labeling the eyes images obtained through a webcam as open and closed eyes. The extracted eye images were converted into grayscale images, resized to 24x24 pixels, and normalized. We did not collect new data for the Data augmentation module, rather we transform the already present data. We achieve data augmentation using Keras in which it accepts the ranges for rotation, brightness, shear, zoom, etc as parameters. Each of CNN layers has various arguments that can be modified and execute various tasks on the input information as shown in Figure 2. We have used relu activation function in convolutional layers with kernel size of 3x3. Softmax activation function is used in fully connected layers to give the outcome as either open or closed eyes. CNN is trained with adam optimizer.

.

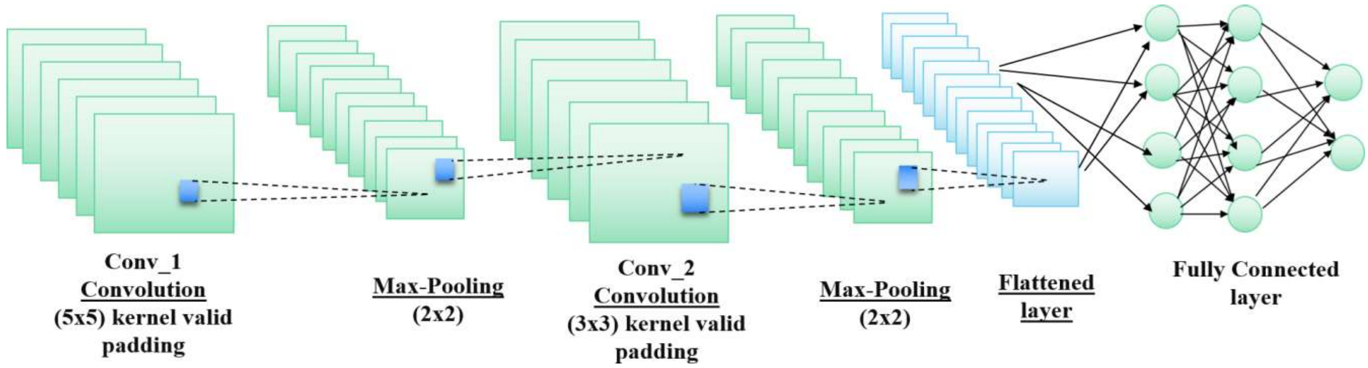

Fig 2. Enhanced CNN Architecture

We use Pygame library to give warning to the driver while starting sleep. The score value is used to find how long the driver has closed eyes. If the eyes are both closed, we increase the score and when eyes are open, we decrease the score. We are drafting the outcome to display the actual time condition of the driver. The model summary details 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 depicted in Figure 4.

```
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
y: 0.9910
```

Fig. 4. Output of Enhanced CNN

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

The accuracy speed of Figure 5 and loss of Figure 6 is quicker than predictable training. We used the augmented database in order to train and the original database for test as presented for applying open and closed eye images.

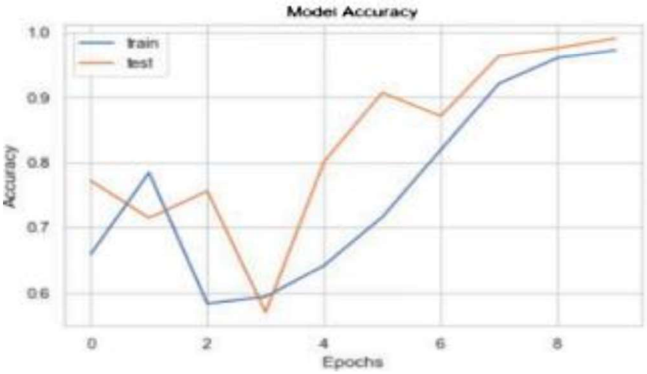

Fig. 5. Model accuracy of Enhanced CNN

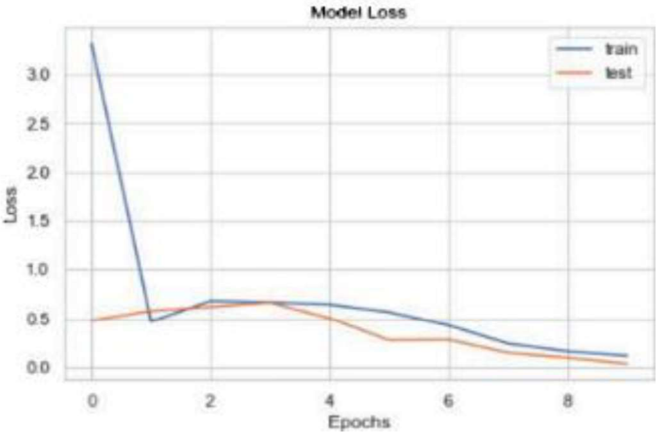

Fig. 6. Model loss of Enhanced CNN

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

After training the CNN, testing is done with a data set consisting of 1000 images of both closed and open eyes. A classification report is generated as shown in Table III. We estimate the suggested drowsiness recognition technique by parameters. Precision means the proportion of actual affirmative interpretations to the entire count of interpretations. Recall is the proportion of properly expected interpretations to whole interpretations in real eye yawn. The mean of precision and recall is f1-score.

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 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.

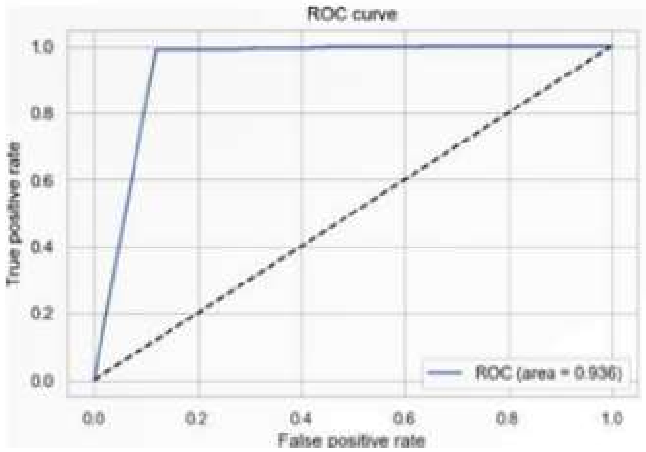

Fig. 7. ROC curve of Enhanced CNN

Figure 8 depicts the face and eye sensing using haar classifier.

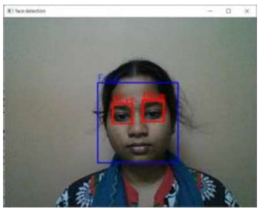

Fig. 8.Detection of face and eye

Figure 9 depicts that the left and right eyes are both open. Therefore, the person is in alert state.

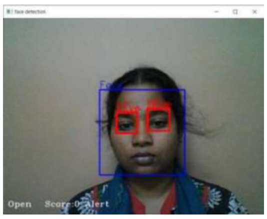

Fig. 9. Alert state

Figure 10 shows that both left and right eyes are closed. Therefore, the person is in drowsy state

Fig. 10. Drowsy state

VI. CONCLUSION

A model for drowsiness sensing depends on effective CNN architecture, planned to observe drowsiness based on eye closure. The implementation started preparing image datasets for both open and closed eyes. 75% of the data set is used for the custom-designed CNN training and the balance 25% of the dataset is utilized for test purposes. First, the information video is transformed into frames and in each frame, the face and eyes are detected. The enhanced CNN supplied an automated and effective learned characteristics that aid us to categorize the opening or closing of eyes. If the closing of eyes occur in 15 successive frames, an alarm is triggered to alert the driver. The proposed CNN gives a training accuracy of 97% and a testing accuracy of 67%. For future works, extra face characteristics can be added to give more accuracy in detection. We can also combine vehicle driving pattern information obtained using On-Board Diagnostics sensors with the facial features extracted.

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, "Driver's Drowsiness Detection Based on Behavioural Changes using ResNet", International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 3, pp. 2 5-30, 2019.

[3] Kartik Dwivedi, Kumar,Biswaranjan and Amit Sethi "Drowsy Driver Detection using Representation Learning", IEEE Advance Computing Conference (IACC), Gurgaon, pp. 995-999, 2014.

[4] Ki Wan Kim, Hyung Gil Hong, Gi Pyo Nam and Kang Ryoung Park, "A Study of Deep CNN-Based Classification of Open and Closed Eyes Using a Visible Light Camera Sensor", Sensors, 2017.

[5] Luigi Celona, Lorenzo Mammana, Simone Bianco, Raimondo Schettini, "A Multi-Task CNN Framework for Driver Face Monitoring", IEEE International Conference on Consumer Electronics, Berlin, 2018.

[6] Mandalapu Sarada Devi and Dr. 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.

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

[8] 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,

[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-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.

[11] 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.

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