Deep Stacked CNN, A Deep Learning Approach for Driver Drowsiness Detection

Deep Patel[a]*, Keyur Modi[b], Om Pandya[c],Dhruv Patel[d],Hetal Shah[e]

*[a*bcd] Student, Computer Engineering Department, Gandhinagar Institute of Technology, Moiti Bhoyan, Gandhinagar, Gujarat, 382721, India*
[e]Assistant Professor, Computer Engineering Department, Gandhinagar Institute of Technology, Moiti Bhoyan, Gandhinagar, Gujarat, 382721, India

Abstract

Driver drowsiness is turn out to be one of the most common reasons for road accidents these days. To overcome this, with the help of advanced technologies like Computer Vision and Deep Learning a system can be designed which alerts the driver quickly if he encountered to be sleeping. Here in this work, a new framework is proposed using deep-stacked convolution Neural Networks to classify the real-time status of a driver into active or sleeping. Dlib and OpenCV libraries are used to extract the face and eye region of the driver from continuously captured real-time images of a driver. A Sigmoid activation function in the output layer of CNN Classifier is used to detect driver status. The proposed method is evaluated on a collected open-source dataset from Kaggle and real-time camera capture images. It has delivered an accuracy of 97.98% when compared to traditional CNN. The limitations of traditional CNN such as pose accuracy and different lightning conditions are overcome with the proposed Deep Stacked CNN model.

Keywords: Deep stacked CNN, Sigmoid Activation, Dlib, OpenCV

1. Introduction

In the modern era, living standards are very high and automobiles are a part of it. Automobiles are utilized as a means of transportation as it is comfortable and save time. Due to this, the amount of automobiles has surged significantly on the road. It is only natural that the number of road accidents also peaked. There are various reasons for a road accident but one of them is driver drowsiness. According to a survey conducted by the Times Of India (TOI), around 1.5 lakhs of accidents have a major reason for driver drowsiness [1]. We can only assume the damage caused due to driver drowsiness worldwide. Police personnel patrolling the highways revealed that most accidents are caused between 2 am to 5 am as drivers are sleep-deprived. Also, the most common age group is 18 years to 40 years who died in these accidents [2][3]. To prevent such accidents to endanger more lives a driver drowsiness alerting system is built. Such a system is a challenge at both the research level as well as at Industrial level.

To detect drowsiness, there are various approaches and signs to observe like the ability to keep eyes open, heart rate of the driver, yawning frequency, head forwarding, etc. but, on a large scale, these approaches are turn out less effective as the behavior of every person is different.

Far now, there are three most common measures for the detection of driver drowsiness as listed below.

- Physiological measures
- Behavioral measures
- Vehicle behavior

In physiological measures, various approaches like Electroencephalography (EEG), Electrocardiography (ECG), and Electrooculogram (EOG) are used to access the driver's conditions, but they turn out to be less practical and not accepted on a daily basis. In-Vehicle behavior, vehicle steering, and braking frequency are monitored but it is more sensitive to road conditions than drivers. In the proposed method, behavioral measures captured via camera and examined by deep-stacked CNN algorithm used which is easy to adapt and not affected by external road and vehicle conditions.

In this method, real-time images of the driver are captured via camera, the captured camera frames are then processed by dlib and OpenCV libraries which extract the eye region and give input to our CNN model. The model classifies the status of the driver and if it encounters the driver to be sleeping for a specified time frame then it will quickly sound an alerting warning alarm, resulting

* Deep Patel
E-mail address: 180120107100@git.org.in

in the driver being alerted and suggested to take a rest or driver carefully. This method can prevent a lot of accidents from happening on the roads.

2. Related Works

Sukrit Mehta et all., has proposed an android-based Driver Drowsiness Detection system. By using the camera, the images of the driver have been captured and passed to a machine-learning algorithm to detect driver drowsiness. The proposed system was tested on a dataset and it showed an accuracy of 84% [4].

Dricare, a system demonstrated by Wanghua Deng1 Ruoxue Wu12 for Drowsiness Detection. The author has used MC-KCF to detect the facial area based on facial key points. Additionally, a method based on the moments of eyes was also introduced and because of this, his model has a high computational speed. It was observed from the experimental results that this model was capable of detecting at different lighting conditions [5].

Rukhsar Khan and Shruti Menon have suggested a drowsiness detection system. The authors have examined various criteria like the position of the head, facial detection, eye blinking moments, etc. The system was made to capture the real-time videos and the video was converted into image frames and each frame was given input to the algorithm for classification. Once the driver shows the sign of drowsiness, it sounds the alerting alarm to warn the driver. Additionally, in this approach, GPS tracking was installed in the cars, and vehicle movements were also captured [6].

C.M Sheela Rani et al and B.Mohana also have demonstrated a drowsiness detection system. In their approach, the authors had used classifier based on Haar in their research to detect the face and eye closure. The proposed model has delivered the accuracy of 85% on the test cases. It is observed that in better lighting condition environment the model performs better [7].

Shivani Sheth, V.V Ramalingam, and Aditya Singhal proposed a drowsiness detection approach. They have used Haar based classifier to recognize the face of the driver in their study. The eye aspect ratio was administered to detect the state of drowsiness of the driver. They have implemented this model on a raspberry pi computer attached with an alarming system. The warning alarming noise by alarming system will alert the driver, which alerts the driver and prevent any unwanted accidents [8].

3. Proposed Workflow of Model

1. Dlib's frontal face detector library is used to detect the face of the driver and also to extract the Region of Interest (ROI) from the detected face which will be given as input to OpenCV's CascadeClassifier for eye region detection.
2. With the help of OpenCV's Cascade Classifier the eye region of the driver's face (Fig. 4(a) & Fig. 5(a)) is extracted and given as input to the CNN model.
3. Two Convolution blocks having 3 and 2 convolution layers respectively are used to extract the features. A single MaxPooling2D layer is used after each block to reduce the dimensions of the feature maps.
4. Three fully connected layers follow convolution layers with ReLU activation at a decreasing rate of neurons (256,128,64) are used for classification. Also, a 30% dropout after each layer is used to avoid over fitting.
5. Sigmoid activation function at output layer plays the role of classifying images into sleepy or active status.
6. The system will monitor if, for a continuous specified time frame, the sleeping state encountered then a warning alarm will be triggered instantly to alert the driver.

4. The Data

For our research work, we have used the Drowsiness_dataset present on the Kaggle platform on this link. There are four classes present in the original dataset (Open Eyes, Closed Eyes, Yawning, or No-Yawning) [9]. However, for this project we have used only two classes (Open Eyes & Closed Eyes) as this projects' scope is to classify drowsiness based on driver's eyes only. So, for this approach the characteristics of the dataset are as follows:

- There is a total of 1452 images classified into two categories (726 images in each category).
- The dataset is distributed equally among both categories, so no need to balance it further.
- Categories (class labels): 'Open Eye' and 'Closed Eye'
- Labels are encoded into 0 and 1.0 for 'Open Eye' and 1 for 'Closed Eye'.

5. Eye Region Extraction

The CNN model does not require the whole image of the driver to detect his status, only the eye region is sufficient. The first step is to detect the face of the driver and for that dlib's frontal face detection library is used. In order to avoid false positives, we first detect the face Region of Interest (ROI), and then with the help of OpenCV, Haar Cascade Classifiers are used to extract the eye region of the driver which are given as the input to our classifier model.

6. Deep Learning Model

Convolution Neural Networks (CNNs) are used to detect the status of the driver into active and sleeping [10]. Generally, CNN required a fixed size input so data preprocessing is required. At first, all the images of the dataset are resized to (32,32,3) which is the standard input for our proposed model. Also, data augmentation is performed on the training data before passing it to the CNN model. The deep CNN model is comprised of various layers like convolution layers, activation layers, pooling layers, dense layers, and dropout layers. The Convolution layer is having kernels (filters) and each kernel has width, depth, and height. This layer produces the feature maps as a result of calculating the scalar product between the kernels and local regions of the image. CNN uses the pooling layers to reduce the dimensions and boost the calculation process. Our model use MaxPooling2D (with pool_size of 2) as a pooling layer which reduces the dimensions of the image by half. In Max pooling, for each region, the maximum value is selected and given as output. ReLU (Rectified Linear Units) is a non-linear function that will return the input unchanged if it is positive, else, zero for all non-negative inputs.

In this method, there are two Convolution blocks having 3 and 2 convolution layers followed by three totally connected dense layers are used. Images with the shape of (32,32,3) are passed as the input to the first convolution layer(conv2d) of the first convolution block. The first block comprises 3 convolution layers each having 32 filters of size 3x3(kernel size) and a non-linear activation function, ReLu. After the convolution layers, Max pooling over 2 x 2 cells with strides of 2 is placed in the architecture which reduces the dimensions in half. Conv2d requires 896 parameters. The output of conv2d is fed into the convolution layer-2(Conv2d_1) which requires 9248 parameters. Convolution layer-3(Conv2d_2) also requires 9248 parameters.At the end of the first convolution block, MaxPooling2D(max_pooling2d), a pooling layer is used which half the input dimensions. It gives an output dimension of (13x13) from an image of shape 26x26.

The second convolution block has 2 convolution layers followed by a Max Pooling layer. Both convolution layers have 32 filters with a kernel size of 3. Both layers (Conv2d_3 and Conv2d_4) requires 9248 parameters and has a ReLu activation function. At the end again a MaxPooling2d layer over 2 x 2 cells with strides of 2 is used which half the input dimension from 9x9 to 4x4. After the convolution blocks, before passing the output to the dense layers, a Flatten layer is used which transforms a multi-dimensional vector of shape (None,4,4,32) into a linear 1-dimensional vector of shape (,512). Here, three connected dense layers with ReLU activation functions and neurons at decreasing rates (256,128,64) are added. Between each dense layer, a dropout layer with 30% is placed to avoid overfitting.

In the end, the output layer having a sigmoid activation function is used to classify the status of images into active and sleeping states. The output layer has only 2 outputs, which turn out to be a binary classification problem so a sigmoid activation function is used. Also, we have set the losses to 'binary_crossentropy' as we have only 2 outputs to deal with. Adam optimizer with a default learning rate (0.0001) is used as it gives the best possible results. Overall, the proposed model has a total of 210,498 parameters with 210,498 trainable parameters and 0 non-trainable parameters.

The model architecture is shown in figure 1.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 conv2d_1 (Conv2D)           (None, 28, 28, 32)        9248

 conv2d_2 (Conv2D)           (None, 26, 26, 32)        9248

 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0
 )

 conv2d_3 (Conv2D)           (None, 11, 11, 32)        9248

 conv2d_4 (Conv2D)           (None, 9, 9, 32)          9248

 max_pooling2d_1 (MaxPooling  (None, 4, 4, 32)         0
 2D)

 flatten (Flatten)           (None, 512)               0

 dense (Dense)               (None, 256)               131328

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 128)               32896

 dropout_1 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 64)                8256

 dropout_2 (Dropout)         (None, 64)                0

 dense_3 (Dense)             (None, 2)                 130

=================================================================
Total params: 210,498
Trainable params: 210,498
Non-trainable params: 0
```

Fig. 1. Convolution Neural Network model architecture.

7. Testing

For testing purposes, we have tested this model on various types of datasets. For example, On the video, image dataset, and on real-time webcam. And in all these types we get accurate results. This model delivers a training accuracy of 98.78% and a validation accuracy of 97.98% when tested on the dataset. The alarm system is designed in such a way that if the driver is found in the sleeping status for more than a specified period of time then it will identify the driver as drowsy and sound the alerting alarm [11] to warn the driver and avoid any accident.
The confusion matrix is shown in figure 2. (0 represents open eye and 1 represents closed eye).
The training and validation losses and accuracies curves are shown in the figure 3.

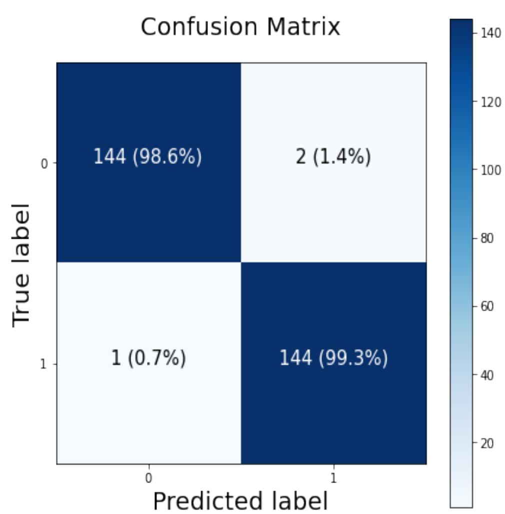


Fig.2. Confusion Matrix.

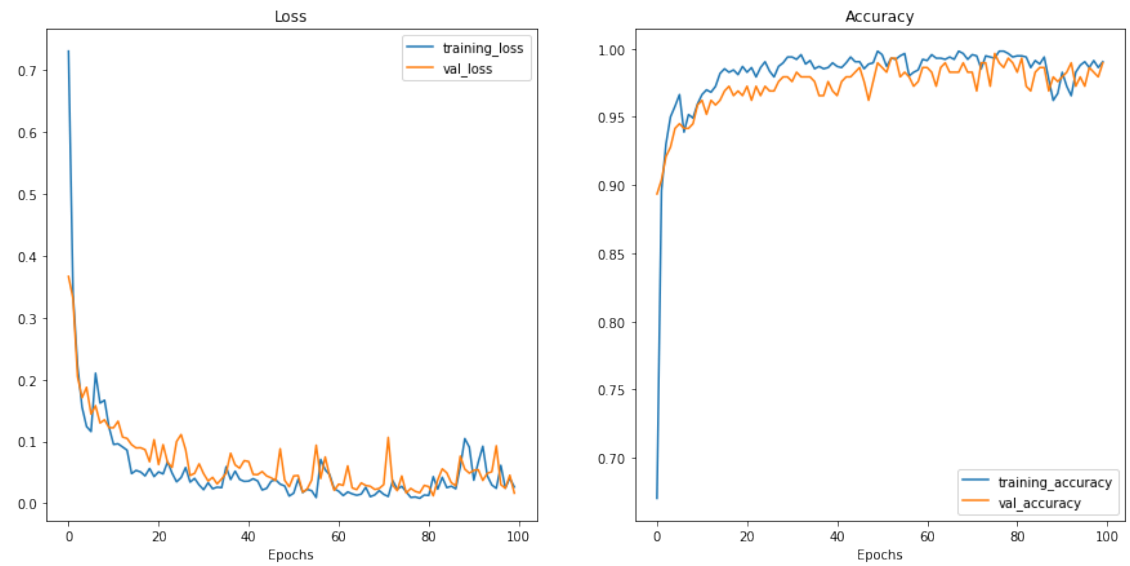
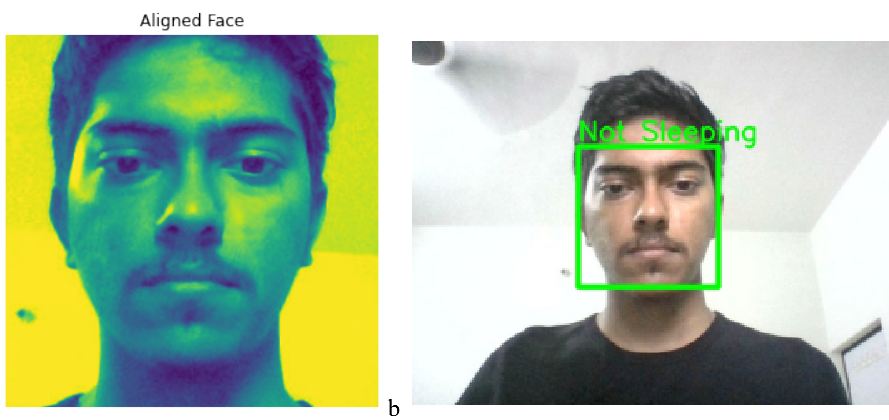


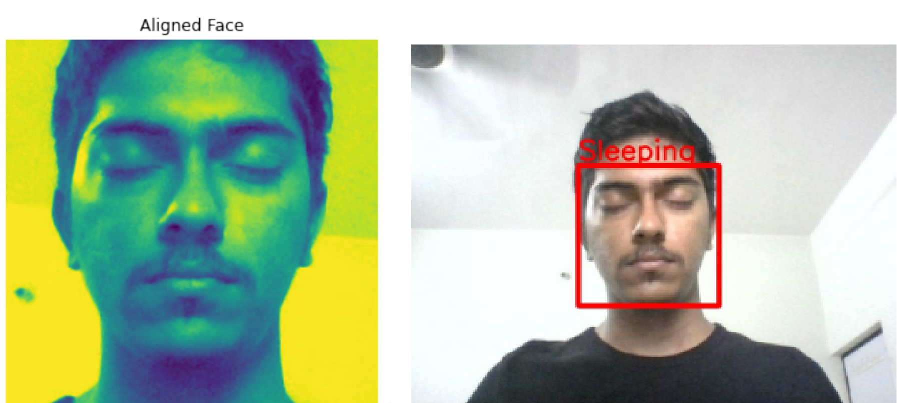
Fig.3. Losses and accuracies curves against no of epochs for training and validation data

Experimental Outcomes:



a b

Fig.4. (a) shows the region of interest from the full image of driver and (b) shows the prediction of the driver's status (active status) by our classifier model.



a b

Fig. 5. (a) shows the region of interest from the full image of driver and (b) shows the prediction of the driver's status(Sleeping) by our classifier model.

8. Conclusion

Here, in this approach, a new method is proposed to detect the drowsiness of the driver. To accurately extract the eye region from the driver's face, dlib and OpenCV library are used. A deep-stacked CNN architecture is used to extract the features from the input images and the sigmoid activation function at the output layer is responsible to detect the current status of the drive. This method delivers an accuracy of 97.98%. The proposed method precisely detects the status of the driver into an active or sleeping state and alert with a warning alarm if the driver encounters to be sleeping for a certain period of time (adjustable as required). To further improve this model, we will focus on using transfer learning for more accurate performance.

References

1. https://timesofindia.indiatimes.com/city/agra/sleep-deprived-drivers-reason-behind-road-accidents/articleshow/64606900.cms

2. https://www.ibef.org/industry/automobiles-presentation

3. https://www.thehindu.com/news/national/kerala/sleep-deprived-drivers-responsible-for-40-of-road-accidents-say-transport-officials/article61629032.ece

4. Sukrit Mehta, Sharad Dadhich, Sahil Gumber, Arpita Jadhav Bhatt" Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio ", International Conference on Sustainable Computing in Science, Technology, and Management,2019

5. Wanghua Deng1 Ruoxue Wu12" Real-Time Driver-Drowsiness Detection System Using Facial Features", IEEE Access

6. Rukhsar Khan, Shruti Menon, Shivraj Patil, Suraj Anchan, Saritha L. R." Human Drowsiness Detection System", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-4, April 2019.

7. B. Mohana and C. M. Sheela Rani" Driver Drowsiness Detection Based on Yawning Detection and Eye Closure", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-4, November 2019

8. Shivani Sheth, Aditya Singhal, V.V. Ramalingam" Driver Drowsiness Detection System using Machine Learning Algorithms", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277- 3878, Volume-8 Issue-6, March 2020

9. https://www.kaggle.com/dheerajperumandla/drowsiness-dataset

10. https://www.tensorflow.org/resources/learn-ml?gclid=CjwKCAiAvaGRBhBlEiwAiY-yMDo_9Z2liJ0BAqwHBMsugJvz7lGQvZI8qV87TGjSR2VA-jyE4PGojxoCwdsQAvD_BwE

11. https://commons.wikimedia.org/wiki/Category:Audio_files