International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

CNN Model to Build Drowsiness Detection System

Prof. D. R. Uike,
Anurag Shrirao, Kunal Raut, Pratik Moharkar, Yash Lanjewar

(Department of Information Technology, Government College of Engineering, Amravati – 444604, Maharashtra, India)
(dilip.uike100, anuragshrirao, kunalraut2110, pratikmoharkar, yashlajewar01)@gmail.com

ABSTRACT: Travel during rush hour or on holiday, uncover the driver to a traffic jam for an hour, thus causing the driver to feel drowsy easily due to high concentration and lack of rest. In this situation, it increases the percentage of a car accidents due to car driver overtiredness is the primary origin of the car accident. So, this drowsiness detection system is proposed to detect the state of the car driver using techniques like face detection, drowsiness is defined to be a condition of exhaustion, where the expression of the face is different from usual. The important steps in detecting drowsiness are face detection, eye detection, and expression detection. The light and position of the camera is the major problems. In this project, different architectures such as haarcascade classifier and CNN were used to analyze the performance of face and drowsiness detection. Also, we have proposed new detection methods using deep learning techniques. To detect the driver's state, we use facial regions corresponding to the entire face. For the Classification, The CNN (Convolutional Neural Network) architecture integrated in the drowsiness detection.

Keywords: Convolutional Neural Network (CNN), Drowsiness detection, Face Detection, Deep Learning, multi-modal analysis

I.INTRODUCTION

Automobiles have become an essential mode of transportation for people rather than other mode of transports. Nowadays, statistic shows that road accidents are the primary origin of the number of people's death, compared to other root causes all over the world. There are a lot of problems which cause road accidents, like (i) the situation of the road such as slippery and flood on road (ii) the condition of the vehicle, the braking system problem and the main problem is (iii) the lack of rest of the driver, causes to drowsiness. The driver may contribute to the drowsiness effect if the driver does not have enough rest and thus may cause a serious road accident. Drowsy driving is the most dangerous aspect of road accidents.

The methods for drowsiness detection are classifiedinto subjective and objective detection methods. In objective detection, no feedback is given to drivers and detection takes place according to the drivers' physical aspects but subjective detection is all about the physical aspects of drivers. The objective type of detection is furthergrouped into contact and non-contact types. The proposed system is based on a non-contact method since it is low cost as compared to the contact method.

The main goal of this project is to determine whether a driver is drowsy or not. The algorithmic pipeline analyses each frame image of the video stream and detects the driver's condition – whether feeling drowsy or not. The proposed system is based on a non-contact method since it is low cost as compared to the contact method. After detection, the next step is giving an alert alarm to the driver so that he can take necessary action. Here deep learning technique is used with the help of a Convolutional Neural Network (CNN).

The drowsiness can be detected through some facial expressions such as blinking of eyes, eye closure, and head pose. This can be done by installing a camera in front of the driver to capture real-time images of the driver. The driver's images are then further processed to detect the drowsiness of the driver. This can be done byperforming live monitoring by application of image processing using opencv.

This explains why it is important to do more work in this field to reduce the chances of such serious accidents related to drowsiness and motivate ourselves to develop a driver drowsiness detection system.

II. METHODOLOGY

In the drowsiness detection system, the whole process is carried down according to image processing which is a method to perform some operations on an image. Figure 1 shows the Conventional Drowsiness Detection System.

The system flow has five steps are as follows:

i) Video Capturing

ii) Detection of Face

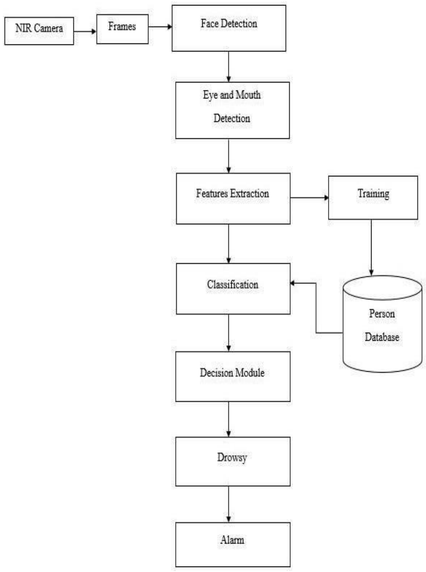
iii) Detection of Eye (alternate approach)

iv) Assessment of State

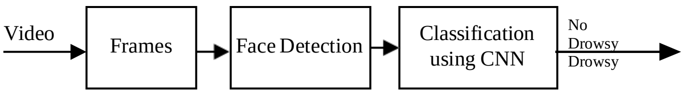
v) Categorization into drowsiness or non-drowsiness state.

The proposed system has these basic steps: the first video is captured through a Web camera then frames are converted, the second face is detected, the third feature detection, and the fourth is drowsiness detection using the deep learning algorithm CNN. The proposed system is based on a non-contact method since it is low cost as compared to a contact scheme. The paper uses deep learning techniques for feature expression, keeping away from manual feature extraction. Feature extraction is a type of dimensionality reduction where a large number of pixels of the image are efficiently represented in such a way that interesting parts of the image are captured effectively. The reduction of data helps to build the model with less machine effort and increases the speed of learning and generalization steps in the machine learning process.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930



(a)



(b)

Figure 1: (a) Conventional Drowsiness Detection System (b) Proposed System

A. Video Capturing

The images are collected using the webcam live. Each image has a resolution of 512 x 424 pixels and grayscale values of 8 bits. This data contains different subjects, but we use only drowsy and non-drowsy conditions. We want to classify drowsiness conditions.

B. Face Detection

Haar features are extracted here to detect markers like face and eye. This method is similar to convolution operation which can detect a feature in a given image. The sum of pixels that fall under the white rectangle is subtracted from the sum of pixels within the black rectangle to get the results of each feature.

Haar -Cascade algorithm:

Step 1: Load the required frontal face, left eye, and right eye XML classifiers

Step 2: Load our input video in grayscale mode

Step 3: Find the faces in the image. If faces are found, it returns the positions of detected faces as Rect (x,y,w,h)

Step 4: Get the locations and create an ROI (Region of interest) for the face and apply eye detection on this ROI.

Step 5: Calculate the score based on the eyes open or closed.

Figure 2: Close Eye Image from the dataset

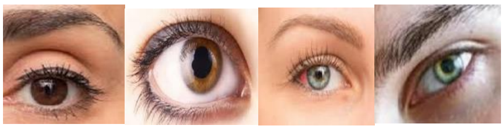


Figure 3: Open Eye Images from the dataset

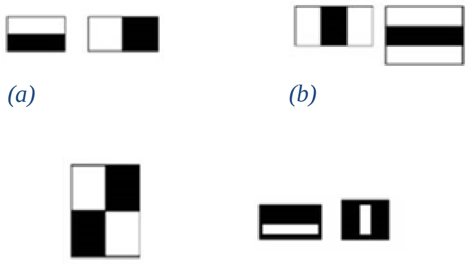


Figure 4: HAAR features (a) Edge features (b) Line features (c) Four rectangle features (d) Face detection features

Cascaded classifiers comprise stages each containing a strong classifier and the job of each stage.

C. Drowsiness Detection

Drowsiness detection is the classification process in the system. Many machine learning approaches are already developed for classification. But these approaches will not satisfy the goal. As the dataset was collected through Webcam in the vehicle environment, the Image has only a low light illumination. Deep learning approaches will give good results. Many Pre-trained deep networks are available for drowsiness detection. Two types of classification methods can be used. i) Analyze the eye region of interest to determine whether the eyes are opened or closed. ii) Analyse the entire region of interest of the face. Drowsiness can be classified according to the expression of the person.

Convolution Neural Network

Three types of layers make up the CNN which are the convolutional layers, pooling layers, and fully connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below. Convolution is a mathematical operation that allows the merging of two sets of information. In the case of CNN, convolution is applied to the input data to filter the information and produce a map. This filter is also called a kernel, or feature detector.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

1. Convolutional Layer

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM.

By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image concerning the size of the filter (MxM).
The output is termed the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

2. Pooling Layer

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce computational costs. This is performed by decreasing the connections between layers and independently operating on each feature map. Depending upon the method used, there are several types of pooling operations.
In Max Pooling, the largest element is taken from the feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.

3. Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.
In this, the input image from the previous layers is flattened and fed to the FC layer. The flattened vector then undergoes a few more FC layers where the mathematical function operations usually take place. In this stage, the classification process begins to take place.

4. Dropout
Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data hurting the model's performance when used on unknown data.
To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during the training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

5. Activation Functions

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network. It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH, and the Sigmoid functions. Each of these functions has a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred and for multi-class classification, generally, softmax is used.

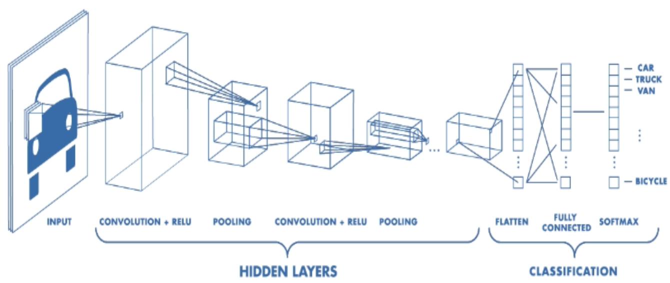


Figure 5: CNN Architecture

III. SYSTEM DESIGN

System Architecture:
When the driver is driving, the driver's face is captured by a camera, and it is converted into a video stream. The application then analyzes the video to detect drowsiness and fatigue and checks the level of drowsiness. In this stage, the main parts which should be considered for analysis are the driver's face tracking, the driver's fatigue state, and recognition of key regions of the face based on eye closure. Finally, if the drowsiness is detected, an alarm is given.

Fig depicts High-Level System Architecture consisting of the input to the model, preprocessing and evaluation in stages. Stage 1 involves pre-processing of the video stream for human face tracking. Stage 2 involves the extraction of facial key regions such as the eyes and mouth. Stage 3 involves the detection of drowsiness symptoms like eye closure, blinking, and yawning.

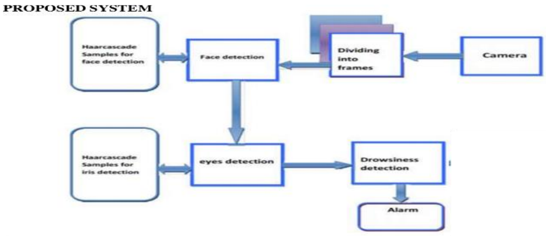


Figure 6: Proposed System

IV. Real-Time use Cases of Drowsiness Detection System:

- In the Security guard cabin.
- In the Military, where high-intensity monitoring of soldiers is needed.
- In Operators at nuclear power plants where continuous monitoring is necessary.
- In a classroom where students feel drowsy and inattentive during the online classes.

V. Result and Analysis

Apart from other research works, this work needs multiple deep learning approaches. Face detection from dim light images is the first objective of this work. These data were trained with our CNN algorithm and created a model. By wearing spectacles, it can cause the system to not recognize the eyes correctly and have an error in detecting the eyes of the driver. To overcome this problem, the system is required to run more samples of images with spectacles so that system can be trained and learn, thus making the system more efficient in detecting sleepy eyes.

Dim light may happen when a driver drives through a tunnel or under a shaded object or at night. Once the system detects the driver is in a dim light environment, the system still worked and act base on the closure of the eyes. After the calculation, the system can give an alert when a prolonged eye closure has occurred. Although the image taken is not clear and of low quality as the picture is taken by a web camera which is more economical and contains only a small number of pixels.

Epoch	Accuracy	Loss
1	0.8958	0.2807
2	0.9479	0.1737
3	0.9531	0.1269
4	0.9427	0.1575
5	0.9635	0.1164
6	0.9531	0.1131
7	0.9792	0.1011
8	0.9688	0.0724
9	0.9740	0.0839
10	0.9896	0.0540
11	0.9844	0.0418
12	0.9792	0.1083
13	0.9740	0.0929
14	0.9722	0.1027
15	0.9792	0.1435
Average	**0.96378**	**0.11793**

Table: Model Training Accuracy and Loss.

The Accuracy of the model comes out nearly 97%.

VI. Conclusion

According to the experimental results, it successfully detects a person during a drowsy condition using the eye behavior. However, there is still space for performance improvement. Further work will focus on detecting the distraction and yawning of the driver using mouth behavior. Other than that, using sensors, for example, liquor sensor and pulse sensor to distinguish liquor and heartbeat pace of the driver can be included for improvement in physiological-measure analysis.

VII. LIMITATIONS

The accuracy of the model degrades if the eye frames are not captured clearly due to any kind of obstacles such as goggles or spectacles having reflection). Camera operations such as auto adjustments to zoom and rotation are not considered in conducting experiments. Once the eyes are localized, zooming in automatically will help increase the accuracy. The accuracy of detection of eyes and mouth reduces when the driver is not facing the camera.

VIII. References

[1] World Health Organization (WHO), "GLOBAL STATUS REPORT ON ROAD SAFETY 2020 – Summary", 2021 Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.
[2] Deng and R. Wu, "Real-Time Driver-Drowsiness Detection System Using Facial Features," in IEEE Access, vol. 7, pp. 118727- 118738, 2019, doi: 10.1109/ACCESS.2019.2936663.
[3] Davis E. King, "Dlib-ml: A Machine Learning Toolkit" Journal of Machine Learning Research 10 (2009) 1755-1758, July 2009
[4] M. Danelljan, G. Häger, F. S. Khan and M. Felsberg, "Discriminative Scale Space Tracking," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 8, pp. 1561-1575, 1 Aug. 2017, doi: 10.1109/TPAMI.2016.2609928.
[5] A. Rosebrock , Eye blink detection with OpenCV, Python, and dlib, (2017) [Electronic] https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/