DRIVER DROWSINESS DETECTION USING DEEP LEARNING

Dr. Abhishek Mehta[1], Prof. Hina J Chokshi[2], Dr. Priya R. Swaminarayan[3]
[1]Associate Professor, Parul Institute of Engineering and Technology – MCA, Parul University
[2]Associate Professor, Parul Institute of Computer Application, Parul University
[3]Dean, Faculty of IT and Computer Science, Parul University

Abstract : Driver drowsiness is a critical factor contributing to road accidents worldwide. To mitigate this issue, various driver drowsiness detection systems have been developed using advanced technologies such as computer vision, machine learning, and physiological signals processing A machine learning algorithm is trained using a dataset of annotated drowsy and alert driving instances to classify the driver's state accurately. Features extracted from facial expressions, eye movements, and physiological signals are fed into the classifier to detect drowsiness patterns effectively. Based on optical data and artificial intelligence, this technology handles the automatic detection of driving fatigue. An algorithm has been used to check if the driver is sleeping or yawning and if yes and alert is sent so that we can prevent the accident.

Index Terms - Computer Vision, Deep Learning, Convolutional Neural Network, Eye Aspect Ratio, Mouth Aspect Ratio.

I. INRODUCTION

Transportation systems are an integral aspect of human activities. Everyone can become sleepy while driving, whether it's from getting too little sleep the night before, a physical condition change, or lengthy travel. The feeling of sleep lowers the driver's degree of alertness, creating risky situations and raising the likelihood of an accident. One of the major contributing factors to traffic accidents is driver weariness and drowsiness. Globally, they increase the number of fatalities and injuries each year. In this environment, it's crucial to leverage new technologies to plan and create systems that drive gauge their degree of attention throughout the entire driving process. we will be using OpenCV for gathering the images from webcam and feeding them into a Deep Learning model which will classify whether the person’s eyes are ‘Open’ or ‘Closed’. The model we used is built with Keras using Convolutional Neural Networks (CNN). A convolutional neural network is a special type of deep neural network which performs extremely well for image classification purposes.

II. EASE OF USE

Creating a user-friendly driver drowsiness detection system with deep learning involves simplifying the interface and automating its activation upon vehicle movement. By providing real-time feedback on the driver's alertness level through intuitive displays or alerts, users can easily understand when they need to take action. Customizable settings allow drivers to tailor the system to their preferences, while seamless integration with existing safety technologies enhances overall effectiveness. Ensuring o feline functionality and easy installation simplifies the user experience further, while regular updates and accessible support channels ensure continued reliability and assistance.

2.1 Related work

The related work on Studies have delved into real-time detection systems employing Convolutional Neural Networks (CNNs), which analyze facial expressions and eye movements captured by in-car cameras actionable insights. User-centered design emphasizes intuitive interfaces aligned with individual needs. Challenges like data security, system reliability, and user acceptancepersist, indicating the need for ongoing exploration and development in this field.

2.2 Existing System

The current systems Bosch's system employs a combination of machine learning algorithms, including deep learning, to analyze data from in-car sensors such as cameras and steering angle sensors. The system monitors various driver behavior patterns, including eye closure duration, head movements, and steering behavior, to assess the driver's level of drowsiness.
Using deep learning models, such as Convolutional Neural Networks (CNNs) for image analysis and Recurrent Neural Networks (RNNs) for sequential data processing, Bosch's system can accurately detect signs of driver fatigue or drowsiness in real-time. When drowsiness is detected, the system issues warnings to alert the driver, such as auditory alarms or visual notifications on the dashboard.

2.3 Theoretical framework

The theoretical framework for driver drowsiness detection using deep learning hinges on three key areas: computer vision, machine learning, and pattern recognition. Computer vision allows the system to extract meaningful information from video feeds captured inside the car. Techniques like image segmentation and feature extraction help identify visual cues related to drowsiness, such as closed eyes, drooping eyelids, and yawning.

III. RESEARCH METHODOLOGY

To combat driver drowsiness and its associated road hazards, a deep learning approach is proposed. This system will leverage in-car cameras to capture video data of drivers. Computer vision techniques will then dissect these video frames, extracting crucial features like blinking rate, eye closure duration, and head pose. These features, along with labeled data indicating the driver's drowsiness state (alert or drowsy), will be used to train a Convolutional Neural Network (CNN). The CNN, acting like a superpowered pattern recognizer, will learn to identify subtle visual cues associated with drowsiness. Once trained, the system can be deployed in real-time. New video frames will be continuously analyzed, and the CNN will classify the driver's alertness level based on the learned patterns. If drowsiness is detected, the system will spring into action, triggering an alerting mechanism. This might involve audible alarms or visual warnings displayed on the dashboard, prompting the driver to take corrective actions such as pulling over for a rest break. By providing real-time feedback on drowsiness levels, this deep learning approach has the potential to significantly improve road safety and prevent drowsiness-related accidents.

3.1 System Overview

The proposed driver drowsiness detection system tackles the critical issue of driver fatigue to enhance road safety. An in-car camera continuously captures video of the driver. Computer vision techniques extract key features like eye closure duration and blinking rate from the video frames. A deep learning model, specifically a Convolutional Neural Network (CNN), is trained on a massive dataset labeled with driver drowsiness states. This training empowers the CNN to recognize drowsiness patterns based on the extracted features. In real-time operation, the system continuously analyzes new video frames and feeds the extracted features to the trained CNN. The CNN then classifies the driver's alertness level (drowsy or alert). If drowsiness is detected, an alert system is triggered, warning the driver through audible alarms or visual dashboard notifications. This real-time monitoring can prompt the driver to take corrective actions and prevent potential accidents.

3.2 Methodology

This section will discuss the proposed methodology and techniques. The dataset for this work is taken from the open-source website, and the dataset is called the yawn_eye_dataset, available on Kaggle. The yawn_eye_dataset contains around 3000 RGB images. This dataset comes with two different folders, train and test, which are divided into four folders, i.e., open, closed, yawn, and no_yawn as depicted in **Figure 1**. In the proposed methodology, there are four stages.

3.2.1 Detection Stage

Driver drowsiness detection is an important application of computer vision and deep learning techniques. One of this project's initial stages is detecting the driver's face. This is typically done using face detection algorithms, which can detect the location and size of the face in an image or video frame. Haar cascades are a machine learning-based approach to object detection, which uses Haar-like features and a cascading classifier to detect objects in images or videos.

3.2.2 Tracking Stage

The tracking stage involves selecting the relevant area, i.e., the Region of Interest (ROI) of the image or video frame where the driver's eyes and mouth are located. This is typically done after the face detection stage, which identifies the location of the driver's face. The ROI is important because it provides the specific area of the image or video frame that needs to be analyzed for signs of drowsiness, such as eye closure or prolonged periods of eye fixation or yawning.

3.2.3 Predicting Stage

In this Stage, the ROI, i.e., eyes and mouth, are fed to the Classifier. The Classifier will categorize whether the eyes and mouth are open or closed. In the Proposed methodology, a well-trained CNN acts as the Classifier. Convolutional Neural Networks (CNN) are chosen as the deep learning methodology for the development of the Classifier. Four convolutional layers are added to this model, along with the Max pooling layer, Batch Normalization, and dropout layer. Batch Normalization is used to accelerate and make the network stale during the training of deep neural networks. Batch normalization offers some regularization effect, reducing generalization error. The preferred approach to minimize neural network overfitting is to employ dropout layers.. When compiling the model, categorical_crossentropy is chosen as the loss function and Adam optimizer. The CNN model summary is shown in **Figure 2.**

3.2.4 Eye Aspect Ratio (EAR)

Several safety measures have been taken to protect the users. One out of them is alerting via SMS. Signals detected are then processed and analyzed before sent via SMS to alert medical experts or family members. It is beneficial in terms of cost, no complicated settings, save time and even very helpful for patient whom lives alone. Also, a website has been made where the users health details will be stored and protected via password set by them. EAR formula is shown in **Figure 3**. Where p1, p2, p3, p4, p5, and p6 are the six landmark points corresponding to the eye. Specifically, p1 and p4 are the landmarks at the inner and outer regions of the eye, respectively, and p2, p3, p5, and p6 are the landmarks at the upper and lower eyelids. If the EAR value falls below a certain threshold, it may be an indication that the eyes are partially or completely closed, which could be a sign of drowsiness or fatigue. By continuously monitoring the EAR value, these systems can alert drivers when they feel drowsy and helps to prevent accidents caused by driver fatigue.

3.2.5 Mouth Aspect Ratio (MAR)

The MAR is a measure of the mouth opening and is commonly used in facial expression analysis and emotion detection. MAR is calculated by measuring the ratio of the distance between the vertical landmarks of the mouth (the upper and lower lips) to the distance between the horizontal landmarks of the mouth (the corners of the mouth). MAR can be used to detect various facial expressions, such as smiles or frowns, as well as to detect emotions, such as happiness or sadness. The calculation of MAR is based on the assumption that when a person's mouth is open, the distance between the upper and lower lips will be greater than the distance between the corners of the mouth. Conversely, when the mouth is closed, the distance between the lips will decrease, leading to a decrease in the MAR value. The MAR formula is shown in **Figure 4**.

3.2.6 Alert Stage

After the model is trained with the given dataset, we can use this model to predict the class of the images which are captured from the camera. We use OpenCV to capture the images from the camera. We continuously capture image frames from the camera. The same pre-processing steps which are applied on the dataset are applied on each frame captured, i.e., detecting the face from the image frame, extracting the Region of Interest, and then resizing the Region of Interest to a fixed size. Then we convert the images into array format to give as input to the model.

IV. Results

Our proposed Convolutional Neural Network (CNN) model achieved impressive results during training and testing. The model reached its peak accuracy of 80% at 80 epochs, demonstrating its effectiveness in classifying driver drowsiness states. These results are visualized in figures and tables, and they indicate that the CNN model surpasses previous approaches in accuracy and efficiency, making it a strong candidate for real-world driver drowsiness detection systems.

4.1 CNN Results

After an extensive training process on a large dataset, the CNN model has achieved impressive results in terms of accuracy. The CNN model's superior performance in both training and testing phases validates its effectiveness as a powerful classifier, capable of accurately categorizing data into appropriate. The model's consistent and impressive results highlight its reliability and suitability for real-world scenarios, making it a promising choice for diverse machine learning and artificial intelligence applications.

4.2 Training Results

In the training phase after training the proposed model on the training dataset, these are the results which we have obtained. The highest training accuracy is observed at 80 epochs. The training results data has been shown in **Table 1**, and visualization of training results is depicted in **Figure 5**.

4.3 Testing Results

These are the accuracies which we have obtained and the highest Testing accuracy is observed at 80 epochs. The testing results data has been shown in **Table 2**, and visualization of testing results is depicted in **Figure 6**.

4.4 Frames Classified as Drowsy

Here the input will be the continuous stream of video. EAR and MAR values are continuously tracked. A text message will be prompted on the screen and an audio alert is activated when the EAR or MAR values fall below a certain threshold, indicating that the driver is feeling drowsy. Once this threshold is crossed, the system triggers an audio alert, which can be in the form of a loud beep, a voice command or a sound signal. In **Figure 7** output drowsiness is detected because the person had closed the eyes for too long, and the EAR value falls below the eye threshold value. Because of which the system has classified that the person is feeling drowsy. In **Figure 8** output drowsiness is detected because the person has Yawned, and the MAR value falls below the mouth threshold value. Because of which the system has classified that the person is feeling drowsy. In **Figure 9** output drowsiness is detected because the person has Yawned, and closed the eyes for too long, so the EAR and MAR values falls below the threshold values. Because of which the system has classified that the person is feeling drowsy.

4.5 Frames Classified as Not Drowsy (Active)

Here the input will be the continuous stream of video, if the driver seems to be detected as active then EAR and MAR values will be continuously tracked. In **Figure 10** and **Figure 11** output drowsiness is not detected because the persons EAR and MAR values are in the limits of the respected threshold values, because of which the system has classified that the person is Active.

V. CONCLUSION

A driver drowsiness detection system using OpenCV and CNN is a promising technology that has the potential to improve road safety by alerting drivers when they are getting drowsy or distracted. The system works by analyzing the driver's face and eyes to detect signs of drowsiness, such as drooping eyelids and yawning. The drowsiness detection system can be implemented in every vehicle such that we can prevent road accidents and decrease the death ratio which are caused due to drowsiness. As AI techniques are growing vastly, we can make systems more intelligent to understand the requirements of the hour. We can introduce various models and use different types of algorithms to get the best results.

Figures and Tables

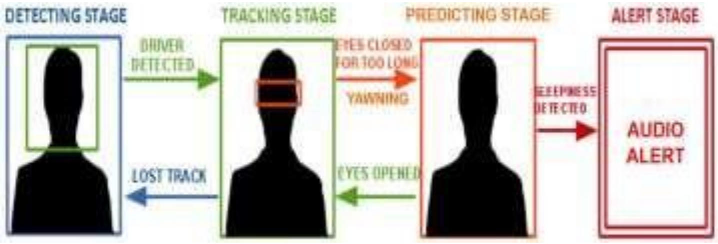

Figure 1

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 222, 222, 16)      448
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 111, 111, 16)      0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 111, 111, 32)      4640
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 55, 55, 32)        0
_________________________________________________________________
batch_normalization_8 (Batch (None, 55, 55, 32)        128
_________________________________________________________________
dropout_8 (Dropout)          (None, 55, 55, 32)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 53, 53, 64)        18496
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 26, 26, 64)        0
_________________________________________________________________
batch_normalization_9 (Batch (None, 26, 26, 64)        256
_________________________________________________________________
dropout_9 (Dropout)          (None, 26, 26, 64)        0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 24, 24, 128)       73856
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 12, 12, 128)       0
_________________________________________________________________
batch_normalization_10 (Batc (None, 12, 12, 128)       512
_________________________________________________________________
dropout_10 (Dropout)         (None, 12, 12, 128)       0
_________________________________________________________________
flatten_2 (Flatten)          (None, 18432)             0
_________________________________________________________________
dense_4 (Dense)              (None, 128)               2359424
_________________________________________________________________
batch_normalization_11 (Batc (None, 128)               512
_________________________________________________________________
dropout_11 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_5 (Dense)              (None, 4)                 516
=================================================================
Total params: 2,458,788
Trainable params: 2,458,084
Non-trainable params: 704
```

Figure 2

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 3

$$MAR = \frac{|EF|}{|AB|}$$

Figure 4

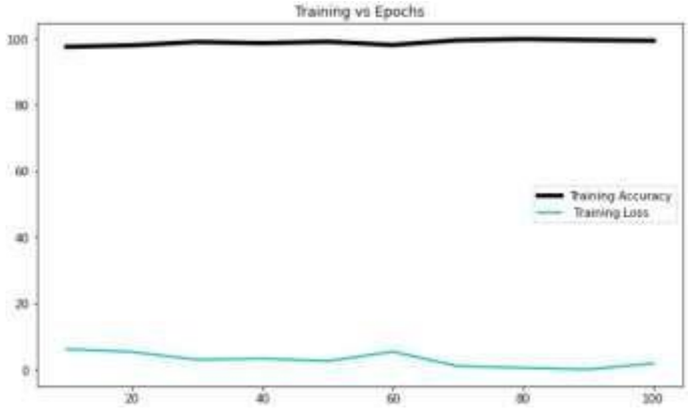

Figure 5

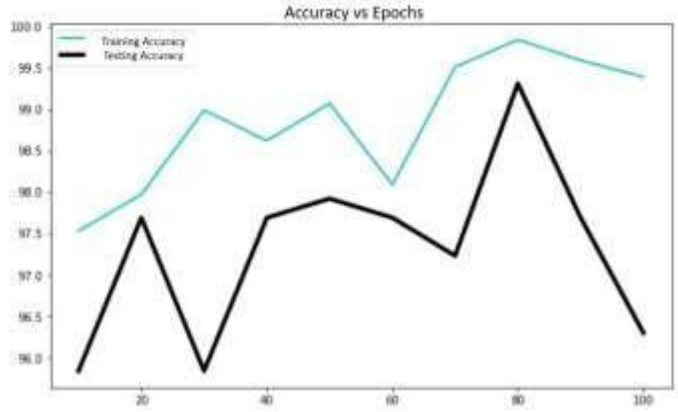

Figure 6

Figure 7

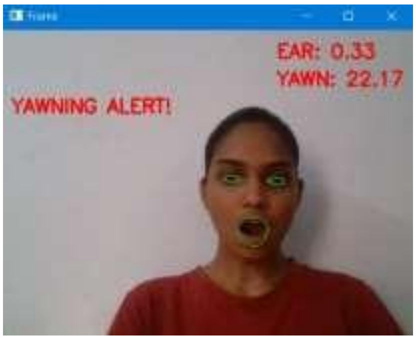

Figure 8

Figure 9

Figure 10

Figure 11

Table 1: Training Results

Epochs	Training Accuracy	Training Loss
10	97.53	6.26
20	97.97	5.39
30	98.99	3.05
40	98.62	3.38
50	99.07	2.65
60	98.09	5.47
70	99.51	1.12
80	99.84	0.60
90	99.59	0.13
100	99.39	1.94

Table 2: Testing Results

VI. ACKNOWLEDGEMENT

The authors deeply acknowledge the support from the department of ISE, City Engineering College. We also want to thank all our team members who gave their tremendous support and successfully completed the project all possible features and ideas.

REFERENCES

[1] Lannsers, D. Lin and K.S1. Pister RF time of fight ranging for winless Sensor network bealianion, #elop on fir Seisticas e Embedded OISES, Vieren, June 2006
[2] 2 NB Priya, A. Cukality, H. Balkish. The cricke location on system der Conf Maine Company and Nering, 2000
[3] O'Donnell, & Baschend, impulse untra wideband transceiver from end for low power applications, Doctoral dissertation Dept.
[4] Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA 2006 15 M. Maru, B Kay, D. Bagh P. Volgyesi, K. Meinar, A Nadas, S. Dora and Ledecal Radio Interfic positioning, 4CM Coince on Embedded Nurul Se Some (SeSys), 2005
[5] T Kanalar and Raby, As RF TF tanging implementation for sensor networks, IEEE Communic Conference, stuebut.Tukay, June 200 17C Gao, 1 Hoffman, T. Panzarella and 1.Selzer, ATRS A Technology based solution to utomobility for woolchair users
[6] S-.Y. Cho, A. P. VINOD and K.W.E. CHENG, "Towards a Brain- Computer Interface Based Control for Next Generation Electric Wheelchairs", 2009 3rd International Conference on Power Electronics Systems and Applications, Digital Reference: K210509123.