Human Driver's Drowsiness Detection System

Arun Prakash[1], B Poojitha Reddy[2], Vishnu Dinesh[3], Amal Dasan P[4], Prof. Mohammed Zabeeulla[5]

[1,2,3,4] *Student, Department of Computer Science and Engineering, Jain University, Bengaluru, India*
[5]*Guide, Department of Computer Science and Engineering, Jain University, Bengaluru, India*

---***---

Abstract - *An Advancements in technology and artificial intelligence in the past few years have led to improvements in driver monitoring systems. Many studies have collected real driver drowsiness data and applied machine learning algorithms to enhance the performance of these systems. This paper presents a review report on the project to develop a system for driver drowsiness detection to prevent accidents caused by driver fatigue. This paper contains reviews of recent systems using different methods to detect drowsiness. In this paper, the proposed system captures video of the driver's face to detect drowsiness and alert an alarm if needed. A machine-learning algorithm was applied to the model to evaluate the accuracy of this approach. Real-world implementation of the project gives an idea of how the system works and what can be done to improve the accuracy of the overall system. Furthermore, the paper highlights the observation, accuracy, and challenges of the system.*

***Keywords*: Accidents, Drowsiness Detection, Eyes Detection, Fatigue, Machine Learning, Yawn**

1. INTRODUCTION

We often hear of drunken driving, no seat belt, speeding, harsh weather, and mechanical failures. But one of the biggest and yet often unrecognized human errors is drowsy driving. A major problem not only in India but across the globe. The risk, danger, and often tragic results of drowsy driving are alarming indeed. Lack of sleep is the main culprit. And add to it the catalyst agents like medications, alcohol and sleep disorders, and sleepiness gets aggravated. Many drivers in the country sacrifice sleep, an often overlooked and dangerous behaviour that results in the majority of them being sleep-deprived while behind the wheel every day. There is no official count of lives lost in drowsy driving-related crashes in our country. Falling asleep at the wheel is suicidal. It is not only dangerous to the driver but all other road users. In this project we are going to build a detection system that uses OpenCV to capture drivers' faces using Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR) and a machine learning model to detect drowsiness

2. PROPOSED SYSTEM

According to current technology, monitoring drivers while driving is quite complex computation and expensive equipment, and it is also not comfortable to wear while driving. For example, EEG, and ECG to check the frequency and rhythm of the heart. As a new solution, a drowsiness detection system which uses a camera placed in front of the driver is more suitable to be used but the physical signs that will indicate drowsiness need to be located first to come up with a drowsiness detection algorithm that is dependable and accurate. Lighting intensity and the orientation of the driver are the problems that occur during the detection of eyes and mouth regions.

So, in this project, we propose a method to capture drivers' faces using a webcam or a small camera and analyze each frame of the video we are getting to detect drowsiness.

3. OBJECTIVE

- To suggest a way to detect fatigue and drowsiness while driving
- To investigates the physical changes of fatigue and drowsiness
- To develop a system that uses the closing of eyes and yawning to detect fatigue and drowsiness.
- To provide an alert (sound) when drowsiness occurs

4. SYSTEM IMPLEMENTATION

In our project, implementation is done in two parts. The first part is building a model to detect the status of the eyes. The next part is detecting how many yawns are captured. Since the closure of eyes and yawn count are important aspects of drowsiness.

We will be using Python, TensorFlow and OpenCV to build this detection system. OpenCV will be used to monitor the driver using a webcam and the feed will be transferred into our machine learning model to detect the status of the eyes.

Face landmarks:

DLib is a cross-platform software library originally written in C++ that contains machine learning algorithms. Even though it is a C++ library, many of its tools can be used in python. It is majorly used for face detection and facial landmark detection. The DLib library can be used to detect a face in an image and then find sixty-eight facial landmarks on the detected face.

Figure 1: *Facial landmarks*

Sometimes we do not need to use all the landmark points but for our project, we are going to use eyes and mouth landmark points.

Detection of Eyes:

At first, the video is captured using a webcam or a small camera. From that video, the face of the driver is detected using the Haar cascade algorithm and then the eyes are detected. Then we used our CNN model to detect the status of the eye with an accuracy of 98%. Haar Cascade Classifier is used for detection of the face and EAR is used on both eyes. EAR is defined as the proportion between the height and the width of the eye based on its landmarks. If the EAR goes below a threshold value i.e., closing of eyes for a particular period an alert in the form of an alarm will be heard by the driver.

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

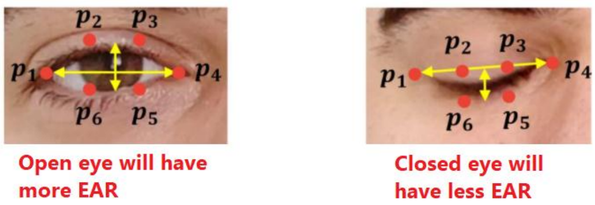

Figure 2*: EAR*

Yawn Count:

The mouth is represented by eight landmark points. At any point of time when a person opens his/her mouth to yawn, the distance between the upper and lower landmark point increases. The proportion between upper and lower lip distance to the level distance between the corner of the lips is used to determine MAR. We will assign a threshold yawn count. If the person's yawn count is more than the threshold count an alert in the form of an alarm will be heard by the driver.

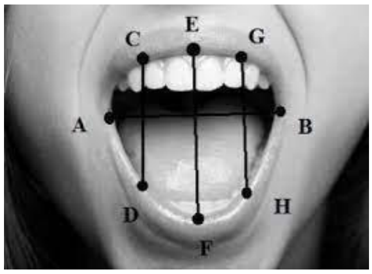

Figure 3: *MAR*

5. ARCHITECTURE

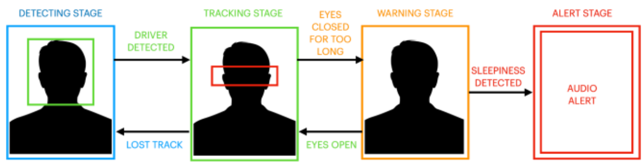

Figure 4: *Architecture*

Detecting stage:

In the initial detecting stage, the face of the person is detected and creates a square boundary as shown in the figure by enclosing the core part of drowsiness like eyes and mouth inside the boundary and for this we will be using Haar cascade by google to detect the face.

Tracking stage:

After detecting stage there will be tracking stage where the core region of drowsiness (eyes, mouth) will be tracked by using two parameters - MAR (Mouth Aspect Ratio), EAR (Ear Aspect Ratio) for tracking whether the person is closing eyes than a particular ratio as well as the person is opening mouth than a particular ratio in order to detect the drowsiness.

Warning stage:

After detecting the eyes and mouth if the person is tracked as yawning or sleeping the model will provide a warning alarm that will be just to alert the drowsiness to make the driver engage in driving.

Alert stage:

In this final stage, If the yawning and sleeping count goes more than a particular count there will be an alertness to stop driving. In this stage instead of giving a small alert the alarm will be continuously beeping until the driver stops the car.

6. SEQUENCE DIAGRAM

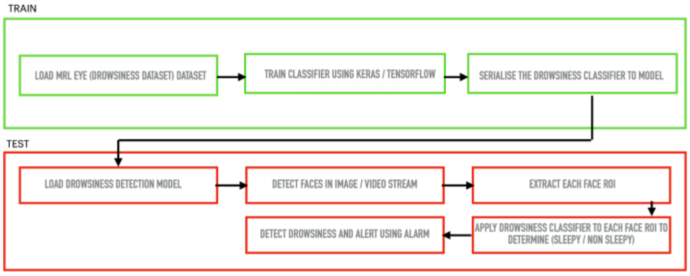

Figure 5: *Sequence Diagram*

Training stage:

In the training stage, the dataset (MRL eye) containing the images of classes (open, close) eyes will be pre-processed by resizing, converting to grayscale etc. As the better prediction is occurred when we have a greater number of samples so to occupy more number of samples, we will be doing data augmentation, by using image data generator we will be generating more samples that contains (rotated, flipped, inverted etc.) samples of the original image so while training the machine learns better. Moving further we are using a pre trained model (sequential) that contains pre trained parameters that can be directly used in our model in order to use that we will be importing the model and after adding layers that is needed for our classification, we will be fitting the model. After training the model will be saved to the system in .h5 format.

Testing stage:

In the testing stage, we will be using OpenCV for real time face detection using video streaming. For classification of drowsiness we will be importing the trained .h5 model as well as we will be using Haar cascade classifier for the face detection and to create boundary, moving further the ROI like eyes and mouth will be extracted for capturing drowsiness feature and if the drowsiness is captured through video the warning alarm will be given and if the count of detection will be increased the stopping alarm will be given to the driver in order to stop driving.

7. HARDWARE DETAILS

- Laptop or PC:

Figure 6: *Laptop*

- Small camera or webcam:

Figure 7: *Webcam*

8. SOFTWARE DETAILS

- Python 2.7 and above is required as we are using python as our language to implement, and we recommend the latest version 3.7 for the TensorFlow environment.
- OpenCV is a huge open-source library for computer vision, machine learning, and image processing. By using it, one can process images and videos to identify objects, faces, or even the handwriting of a human.
- Anaconda navigator: we need an "anaconda navigator" for the implementation of the project and it is used for launching a variety of python interpreters and python IDE. And we recommend the latest version.
- Jupyter notebook - we need to install the latest version of "Jupyter notebook" as a python interpreter, or you can use "visual studio code," or Google Colab which is a cloud-based online python interpreter by google.

9. OUTPUT

If the EAR goes below 0.25 and the yawn count goes above 4, the person is said to be

in drowsiness state. The detection was done based on whether the person was wearing spectacles or not.

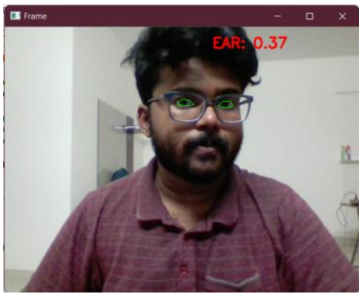

Figure 8*: With spectacles output 1*

The person has yawned four times as of right now.

***Figure 9**: With spectacles output 2*

The person has yawned more than four times. An alert will display along with an alarm ringing stating that the "Subject is yawning a lot".

***Figure 10**: With spectacles output 3*

The EAR has gone below 0.25 stating that the person is sleeping. A drowsiness alert will be displayed along with an alarm ringing.

***Figure 11**: With spectacles output 4*

10. OUTPUT ANALYSIS

To get a better understanding of the behaviour of the system, we investigated the results of the most effective model in the previous round of testing.

When we started this project, we thought of getting a better or better accuracy compared to the other studies. We then implemented the CNN model on the dataset, and we were provided with the best accuracy among all which is 98% for drowsiness detection.

First implementation: CNN (Inception V3)

Final implementation: CNN (Sequential)

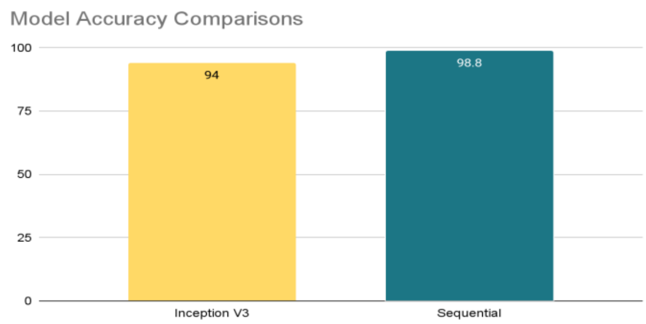

***Figure 12**: Model Accuracy Graph*

11. CONCLUSIONS

In the end, we were able to detect the drowsiness of the driver in real-time. This is done by installing a webcam or a small camera in front of the driver. A CNN model was applied to the real-time video to detect the status of eyes with 98% accuracy. If the eyes were closed for some frames in the video the driver will hear an alert sound.

Similarly, an algorithm was created using Haar Cascade Classifier to detect yawn counts based on MAR (Mouth Aspect Ratio). An alert sound was heard if the yawn count goes above the threshold count (i.e., four yawn counts in our method).

Theory 1: OpenCV, CNN.

Theory 2: CNN, LOP.

Theory 3: OpenCV, DLib, EAR, SVM.

Theory 4: Min Max Scalar, CNN.

Our Model: CNN (Inception V3, Sequential), OpenCV, EAR, MAR.

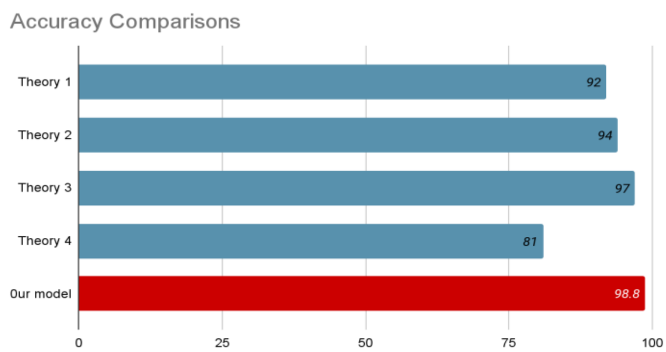

***Figure 13:** Comparison Accuracy Graph*

12. FUTURE SCOPE

The future work on this project can focus on the automatically zooming of the camera onto the eyes once they are localized. Algorithms can be improved to detect eyes and mouth. Lighting changes must be encountered to get better detection. Better video quality to get detection faster as the number of pixels increases, the detection will be accurate. And finally, this system can be connected to the speed control of the car. So, when the system detects the drowsiness state of the driver, it can alert the driver with sound while simultaneously reducing the speed of the car.

For the further proceeding of this project, A prototype model can be developed to stop the car when the driver completely goes under sleep. This can be developed by using IOT devices like Raspberry pi, Arduino and sensors like steering angle sensor, Other than getting drowsiness the person can go under panic attack, faint, dizzy where the driver won't be able to get the control of the car in that case it is necessary to develop a model that stops the car by detecting the face, hand angle position on steering wheel.

In the future, the population and technology grow along with that the usage of a variety of vehicles will be increasing, which will increase more and more traffic violations, So to stop this a system is very important. The model that we created as well as the proposed future model prototype will provide a milestone in the industry of artificial intelligence and automobiles in order to reduce the traffic accidents and violations.

13. REFERENCES

[1] V. Chaudhary, Z. Dalwai and V. Kulkarni, "Intelligent Distraction and Drowsiness Detection System for Automobiles," in IEEE, 04 August 2021.

[2] F. You, X. Li, Y. Gong, H. Wang and H. Li, "A Real-time Driving Drowsiness Detection Algorithm With Individual Differences Consideration," in IEEE Access, 2019.

[3] G. Roshini, Y. Kavya, R. Hareesh, M. Suma and N. Sunny, "Driver Distraction and Drowsiness Detection System," in IEEE, 04 August 2021.

[4] P. Bajaj, R. Ray, S. Shedge, S. Jaikar and P. More, "Synchronous System for Driver Drowsiness Detection Using Convolutional Neural Network, Computer Vision and Android Technology," in IEEE, Coimbatore, India, 03 June 2021.

[5] Flores, Marco Javier, José María Armingol and Arturo de la Escalera. "Driver drowsiness detection system under infrared illumination for an intelligent vehicle." Iet Intelligent Transport Systems 5 (2011): 241-251., "Driver drowsiness detection system under infrared illumination for an intelligent vehicle," vol. 5, 2011.

[6] Avigyan Sinha; R P Aneesh; Sarada K Gopal, "Drowsiness Detection System Using Deep Learning," in IEEE, 4 June 2021.

[7] Rateb Jabbara, Khalifa Al-Khalifaa, Mohamed Kharbechea, Wael Alhajyaseen, Mohsen , "Real-time Driver Drowsiness Detection for Android Application," in The 9th International Conference on Ambient Systems, Networks, and Technologies, 2018.

[8] Charlotte Jacobé de Nauroisa, Christophe Bourdina, Anca Stratulatb, Emmanuelle Diazb, Jean-Louis Vercher, "Detection and prediction of driver drowsiness using artificial neural network," in Elsevier, 2017.

[9] F. Friedrichs and B. Yang, "Camera-based drowsiness reference for driver state classification under real driving conditions," in 2010 IEEE Intelligent Vehicles Symposium, 2010.

[10] K. Satish, A. Lalitesh, K. Bhargavi, M. S. Prem and T. Anjali, "Driver Drowsiness Detection," in IEEE, 01 September 2020.

[11] Challa Yashwanth, "Driver Drowsiness Detection System Based on Visual Features," in IEEE, Greater Noida, Uttar Pradesh, India.

[12] Mohammad Amin Assari; Mohammad Rahmati, "Driver drowsiness detection using face expression recognition," in IEEE, 2 February 2012.