Drowsiness Detection for Drivers

Lavin Peeyus, Lisha Shaji, Rachel John

Department of Information Technology Engineering

Don Bosco Institute of Technology, Kurla, Mumbai.

rachelthomas2212@gmail.com

lishashaji28@gmail.com

vlavinpeeyus@gmail.com

***Abstract*— With the expansion in population, the occurrence of automobile accidents has also seen a rise. A detailed analysis shows that, around half million accidents occur during a year, in India alone. Further, around 60% of those accidents are caused because of driver fatigue. Driver's drowsiness or lack of concentration is considered as a dominant reason for such mishaps Driver fatigue affects the driving ability within the following 2 areas, a) It impairs coordination, b) The reaction time for judgment is longer. If drivers might be warned before they became too drowsy to drive safely, some drowsiness-related crashes might be prevented. Through this paper, we offer a true time monitoring system using image processing & face/eye detection techniques.**

Keywords— Drowsiness Detection, Facial landmarks, Face detection, Eye Detection, Real-time system, OpenCV, Flask.

I. Introduction

One of the most prevailing problems across the globe nowadays is the booming number of road accidents. Even in recent years, there has not been much improvement in the reduction of road accidents. One of the many reasons behind the rise of such road accidents is the driver's fatigue and drowsy state. Exhausted drivers who doze off at the wheel are responsible for about 40% of road accidents, says a study by the Central Road Research Institute (CRRI) on the 300-km Agra-Lucknow Expressway. Driver's drowsiness or lack of concentration is taken into account as a dominant reason for such mishaps. If drivers might be warned before they became too drowsy to drive safely, some drowsiness related crashes might be prevented. To prevent the driver from causing an accident, we propose to provide a system that detects whether a person is drowsy while driving, and if so, alerts him through a voice message in real-time. The system also captures the image of the driver in the drowsy state and saves it to use it as proof if necessary. The driver can register himself on the portal and the sign in when his journey begins, on clicking 'Start', the system applies facial landmark detection to detect and localize the key regions of the driver's face followed by continuously monitoring the driver. If their eyes have been closed for a certain amount of time, we will assume that they are starting to doze off and play an alarm to wake them up and grab their attention.

II. Literature Review

It is explained in [1], two major ways to detect blinking, more importantly detect drowsiness:

By biological approaches like electrooculogram electroencephalogram (EEG) or electrocardiogram (ECG) readings.
By image processing and computer vision methods. Following are a summary of a few scientific papers that are mainly related to the second category. [2] depicts a noteworthy review of the role of computer vision technology applied to the event of monitoring systems to detect distraction. Authors explain the foremost methods for face detection, face tracking and detection of facial landmarks, also because the main algorithms for biomechanical, visual, and cognitive distraction. Additionally, there are some algorithms detecting mixed sorts of distraction and therefore the relationship between facial expressions, key points for the event and implementation of sensors and test and training to driving monitoring systems are summarized. They proposed an algorithm that works in real time to detect the eye blinks.
In [3]. They show that recent landmark detectors, trained on datasets, show robustness against a head orientation with reference to a camera, and facial expressions, being sufficiently precise to estimate the extent of the eye opening. Also, they use a SVM classifier to detect eye blinks as a pattern of EAR values during a brief temporal window.
In [4], a blinking detection method is proposed, supported Gabor filters by measuring the space between the two arcs of the eye . First, the eye is detected by Viola-Jones' method [5]. Next, the Gabor filter extracts the pattern of the eye supported orientation angle. They apply a connected labeling method to detect the two arcs and measure the space between them compared to a threshold.

III. Methodology

This paper methodology is based on two main parts:

a. Real-time eye detection.
b. Drowsiness detection and warnings.

A. *Real-time eye detection.*

To build our real-time drowsiness system, it is necessary to spot the persons eye on the image and then based on it, we can begin to build the blink detector to acknowledge every blink and then distinguish it from a nap or drowse.

Typically, image processing methods for computing blinks involve some combination of: (1) eye localization, (2) thresholding to seek out the whites of the eyes, (3) determining if the “white” region of the eyes disappears for a certain period such that it indicates a blink, so we use a metric called eye ratio (EAR), introduced by [3] . This procedure is completed to capture one’s natural EAR and subtle changes when facial features vary. Our goal is to detect important facial structures like eyes and mouth on the face using the shape prediction method. We use the dlib for facial landmark detection which uses Histograms of Oriented Gradients and Linear Support Vector Machines within the procedure. The library provides landmarks for the whole face, displayed as light green dots in Fig. 1. The landmarks are adaptive to acknowledge the form of distinct human faces.

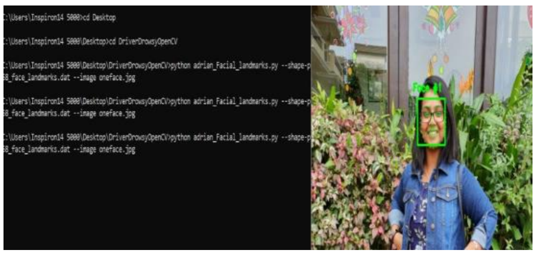


Fig 1: Localizing Facial landmarks for the entire face.

The EAR is calculated based on the following equation, proposed by ref. [3]:

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

Fig 2: Formula to calculate EAR value.

Using this the EAR is calculated for each frame of the video stream. Thus, a decrease in the EAR is expected when the person closes his eyes and then going back to a normal level when eyes are open again. This approach can be used to detect blinks. In order to detect drowsiness, two methods of classification were applied, based on the approach proposed by ref. [3].

The first approach is here called threshold model. A threshold was considered so that if the user’s EAR becomes lesser than this threshold for a number of frames then a blink is then detected.

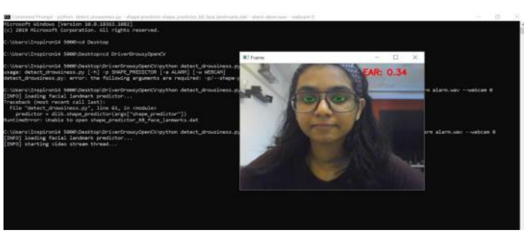


Fig 3: Localizing eye region and Determining EAR threshold value.

B. *Drowsiness Detection and Warnings*

Once predictions provided were considered satisfactory, real-time monitoring could be achieved.

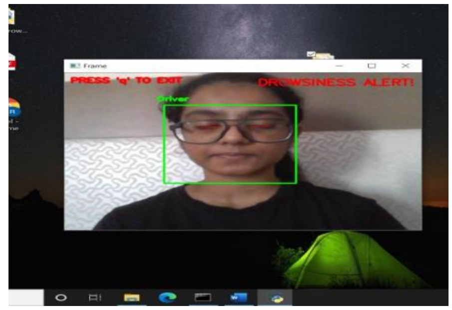


Fig 4: Detecting Drowsiness based on the threshold value and number of frames.

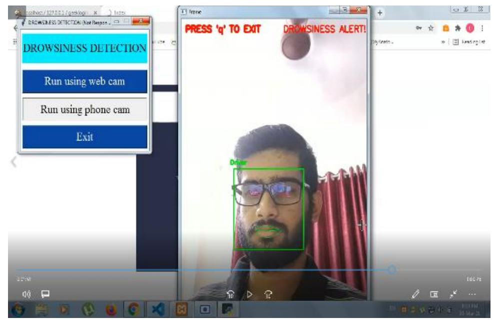


Fig 5: Using IP webcam application to detect drowsiness.

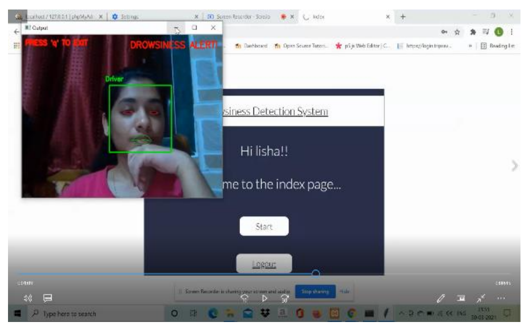


Fig 6: Using laptop webcam to detect drowsiness.

IV. PROPOSED WORK

We have proposed a real time system that begins with the driver registering himself on the web portal created with Html, Css, Jquery and Flask. The driver can then sign in with his credentials and click on ‘Start’.

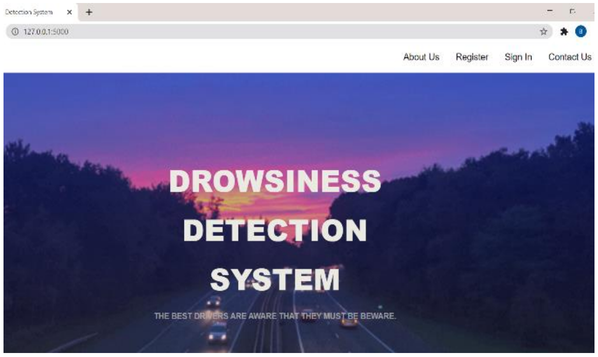


Fig 7: UI of the website.

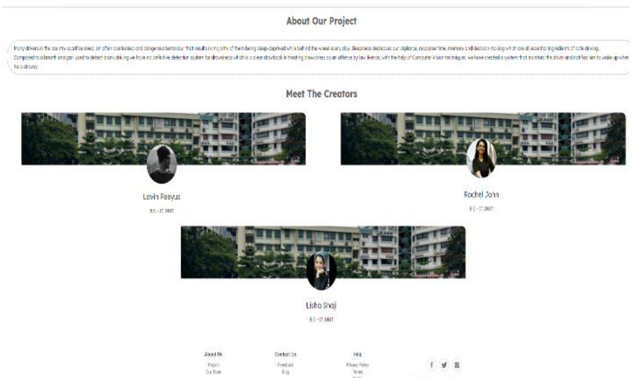


Fig 8: UI of the website.

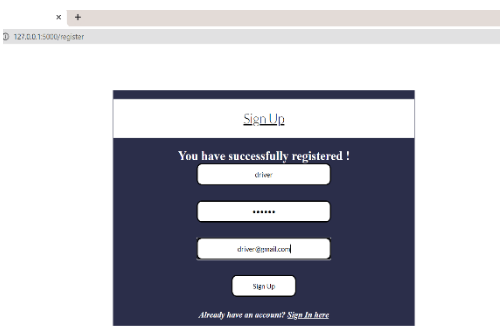


Fig 9: Registration page with success message.

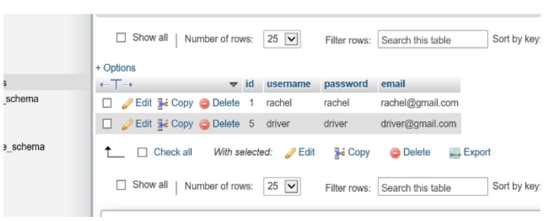


Fig 10: Credentials saved in database.

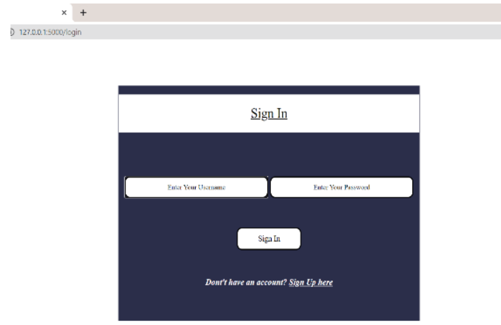


Fig 11: Login page.

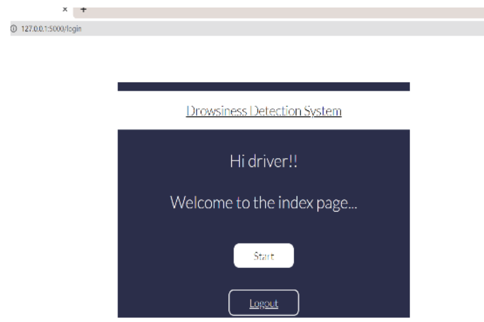


Fig 12: Index page after login.

When the start button is activated, the driver is presented with two options,

a. To monitor using Webcam
b. To monitor using phone cam.

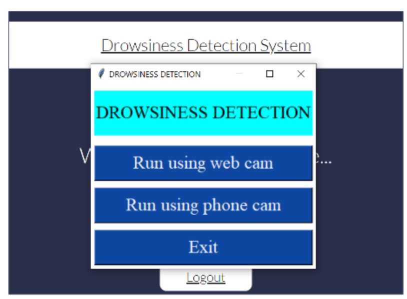


Fig 13: Dialog box for Webcam and Phone cam.

When the first option is selected, we instantiate our video stream using the webcam provided and then the facial landmarks produced by dlib is used to detect and localize the eye and mouth region of the driver specifically.

Dlib facial landmarks: The next step is to acquire the facial landmarks. The basic idea of this technique is to locate 68 specific points on face such as corners of the mouth, along the eyebrows, on the eyes, and so forth. It is a pre-trained detector available in the dlib 1library that is able to find these 68 co-ordinates on any face.

Fig 14: Pre-defined 68 co-ordinates of face defined in dlib library

EAR and MAR calculation: Eye aspect ratio can be calculated using equation (1)

$$\boxed{\text{EAR} = \frac{||p2 - p6|| + ||p3 - p5||}{2 \times ||p1 - p4||}}$$

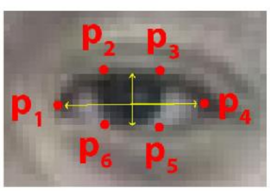
Equation (1)

where, p1, p2, ..., p6are 6 eye landmark shown in Figure 15a. Similarly mouth aspect ratio (MAR) can be calculate using equation (2)

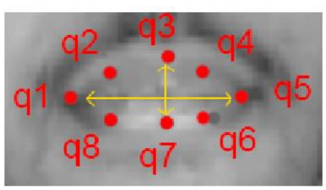
$$\boxed{\text{MAR} = \frac{||q2 - q8|| + ||q4 - p6||}{2 \times ||q1 - q5||}}$$

Equation (2)

where, q1, q2, ..., q8are 8 mouth landmark shown in Figure 15b.



(a) Each eye is represented by $6(x, y)$-coordinates facial landmark of eye region starting from left corner in clock wise direction.



(b) Mouth is represented by $8(x, y)$-coordinates facial landmark of mouth region starting from inner lip left corner.in clock wise direction

Fig 15: Representation of points for MAR and EAR calculation

Drowsiness Detection: The system continuously monitors the driver's states such as sleepiness, yawning and if the driver falls asleep or yawns more than 4 seconds, the systems alert the driver to switch back to normal state. After the driver's journey is completed, he can exit from the system. Upon exit, the images of the driver are captured if the system had caught him in a drowsy state and is saved in the 'dataset' folder. Along with this, a MAR and EAR graph Vs. Time is also plotted to make get a clear idea of the working of the system.

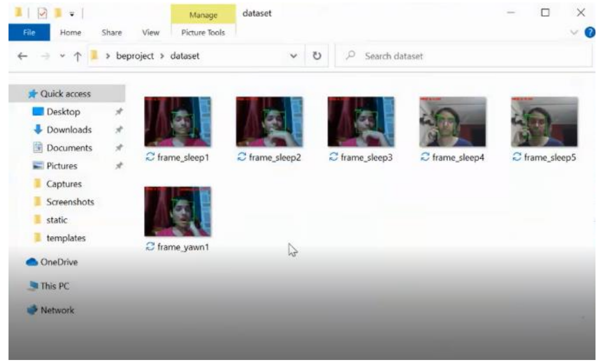


Fig 16: Captured Drowsy Image of Driver in Dataset folder.

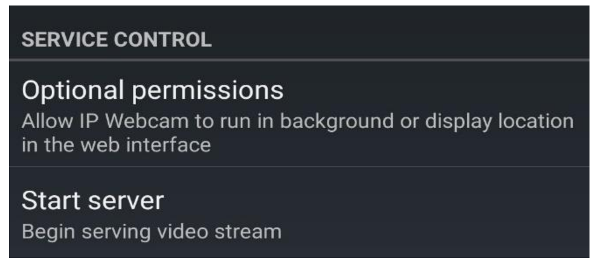


Fig 17: Start Server option on IP webcam interface.

When the second option is selected, we have used an Android Application named IP Webcam. Once we click on 'Start Server' on the application, the system monitors the driver with the same procedure as mentioned above through this application.

V. Future Work

As a future scope, this methodology can be further implemented as a service provided by ride-hailing applications under Driver Security. A thorough through survey and multiple testing can also be done to determine the average threshold to be used for better and accurate results according to different categories such as age and hours of sleep undergone.

VI. Conclusion

Hence, we implemented a complete Drowsiness Detection and alert system for safety of drivers and as a prevention to reduce the rising number of road accidents. We have successfully implemented the concept by adding face recognition which makes the system accessible for multiple drivers.

Acknowledgement

It gives us immense pleasure to present our report on the topic 'Drowsiness Detection for Drivers'. We would like to take this opportunity to thank our guide Prof. Janhavi Baikerikar for her constant support and for helping us and giving us proper insights and guidance.

References

[1] Lefkovits S., Lefkovits L., Emerich S. "Detecting the Eye and its Openness with Gabor Filters". *2017 5th International Symposium on Digital Forensic and Security (ISDFS). (2017)*

[2] Fernandez A., Usamentiaga R., Carus J.L., Casado R. "Driver distraction using visual- based sensors and algorithms". *Sensors (Switzerland). (2016)*

[3] Soukupová T., Cech J. "Real-Time Eye Blink Detection using Facial Landmarks". *21st Computer Vision Winter Workshop. (2016)*

[4] Arai K., Mardiyanto R. "Real Time Blinking Detection Based on Gabor Filter". *International Journal of Recent Trends in Human Computer Interaction (IJHCI), vol. 1, n. 3, 33–45. (2010)*

[5] Viola P., Jones M. "Robust real-time object detection". *International Journal of Computer Vision, vol. 57, n. 2, 137–154. (2001)*

[6] Lenskiy A. a., Lee J.-S. "Driver's eye blinking detection using novel color and texture segmentation algorithms". *International Journal of Control, Automation and Systems, vol. 10, 317–327. (2012)*

[7] Gupta, Isha & Garg, Novesh & Aggarwal, Apoorva & Nepalia, Nitin & Verma, Bindu. (2018). *Real-Time Driver's Drowsiness Monitoring Based on Dynamically Varying Threshold. 1-6.*