Preventing Accidents Through Fatigue Detection An Intelligent Driver Alertness System

Ms. Roopa, Ashwini R

[1]*Assistant Professor, Computer Science Department, JSS College of Arts Commerce and Science Ooty-Road, Mysuru.*

[2]*U.G. Student, Computer Science Department, JSS College of Arts Commerce and Science Ooty-Road, Mysuru*

---***---

Abstract - In this research paper, Driver drowsiness is a major cause of road accidents worldwide, resulting in severe injuries and loss of life. A real-time drowsiness detection system aims to enhance road safety by continuously monitoring the driver's alertness and providing timely warnings to prevent accidents. This system typically uses computer vision and machine learning techniques to analyze facial landmarks such as eye closure, blink rate, and head position. By integrating a camera with a microcontroller or computing device, the system captures live video feeds, processes them in real-time, and detects signs of fatigue. When drowsiness is detected, an alert is triggered through an audio or visual signal. This proactive approach provides an effective solution for mitigating fatigue- related accidents, especially in long-distance or night driving scenarios.

***Key Words*:** Drowsiness detection, Real-Time Monitoring, Driver Safety, Computer Vision, Eye Aspect Ratio (EAR), Fatigue Detection, Machine Learning, Facial Landmark Detection, Blink Detection, Road Safety

1. INTRODUCTION

Real Time Drowsiness behaviour which are related to fatigue are in the form of eye closing, head nodding or the brain activity. Hence, we can either signals, such as brain waves, heart rate and eye blinking to monitor drowsiness or consider physical changes such as sagging posture, leaning of driver's head and open/closed state of eyes.

The former technique, while more accurate, is not realistic since highly sensitive electrodes would have to be attached directly on the driver' body and hence which can be annoying and distracting to the driver. In addition, long time working would result in perspiration on the sensors, diminishing their ability to monitor accurately. The second technique is to measure physical changes (i.e. open/closed eyes to detect fatigue) is well suited for real world conditions since it is non-intrusive by using a video camera to detect changes. In addition, micro sleeps that are short period of sleeps lasting 2 to 3 minutes are good indicators of fatigue. Thus, by continuously monitoring the eyes.

2. LITERATURE SURVEY

literature survey for a **Real-Time Drowsiness Detection System** project. It summarizes key research papers, techniques, and findings related to drowsiness detection, especially using image processing and machine learning:

1.1 Drowsiness Detection using Eye Blink Patterns– PERCLOS Method *Authors:* W. W. Wierwille and L. A. Ellsworth

Summary: This research introduced the concept of **PERCLOS (Percentage of Eyelid Closure)** as a reliable metric for drowsiness detection. It measures the proportion of time the eyes are 80–100% closed over a period. PERCLOS has become a standard for fatigue detection due to its accuracy and low computational requirements.

1.2 Real-Time Eye Blink Detection using Haar Cascades – Viola-Jones Algorithm *Authors:* Paul Viola and Michael Jones *Summary:* The Viola-Jones object detection framework is widely used for face and eye detection. It uses Haar-like features and a cascade of classifiers, enabling efficient real-time performance on embedded systems. It forms the basis for many vision-based drowsiness systems.

1.3 Drowsiness Detection Based on Eye Closure and Yawning Detection using CNN

Authors: Ahmed Saeed, et al. *Summary:* This paper applied **Convolutional Neural Networks (CNNs)** to classify drowsiness by detecting both eye closure and yawning. The system showed higher accuracy compared to traditional methods, with the ability to adapt to different facial features and lighting conditions.

1.4 Driver Drowsiness Detection System Based on Video and EEG Signals *Authors:* X. Zhang et al. *Summary:* This hybrid system uses both **EEG signals** and **video-based facial analysis** for higher accuracy. EEG sensors measure brain activity to detect fatigue, while facial cues like blinking and yawning provide additional validation. Though accurate, the need for EEG hardware makes it less practical for real-world deployment.

1.5 A Real-Time Driver Fatigue Monitoring System Using Eye Tracking and Facial Landmarks
Authors: Soujanya, R., and B. P. Patil *Summary:* The study leverages **Dlib** and **OpenCV** to detect facial landmarks and calculate the **Eye Aspect Ratio (EAR)**, which drops significantly when the eyes are closing. EAR is a lightweight and effective method for continuous monitoring.

3. PROPOSED METHODOLOGY

The proposed real-time drowsiness detection system aims to monitor the driver's facial features using a camera and alert them if signs of fatigue are detected. The system uses computer vision and machine learning techniques for accurate, real-time performance. The implementation is divided into the following modules:

3.1Data Acquisition

- A **webcam or camera module** is installed in the vehicle, positioned to capture the driver's face continuously.
- Real-time video feed is passed to the processing unit (e.g., laptop, Raspberry Pi).

3.2Face and Eye Detection

- Facial features (face, eyes, mouth) are detected using the **Haar Cascade Classifier** or **Dlib's 68-point facial landmark detector**.
- The **Viola-Jones algorithm** (via OpenCV) is commonly used for robust face and eye detection in real-time.

3.3Feature Extraction

- The system extracts **eye aspect ratio (EAR)**:
  - EAR = (vertical eye landmarks) / (horizontal eye landmarks)
  - If EAR remains below a certain threshold (e.g., 0.25) for a specific number of frames, it indicates drowsiness.
- Optionally, **mouth aspect ratio (MAR)** can be used to detect yawning.

Drowsiness Detection Logic

- The system implements a threshold- based algorithm:
  - If EAR < Threshold for N consecutive frames (e.g., 20 frames), the driver is considered drowsy.
  - Yawning and head tilting can also be used as secondary indicators.
- A simple rule-based decision system is used to classify drowsy or alert states.

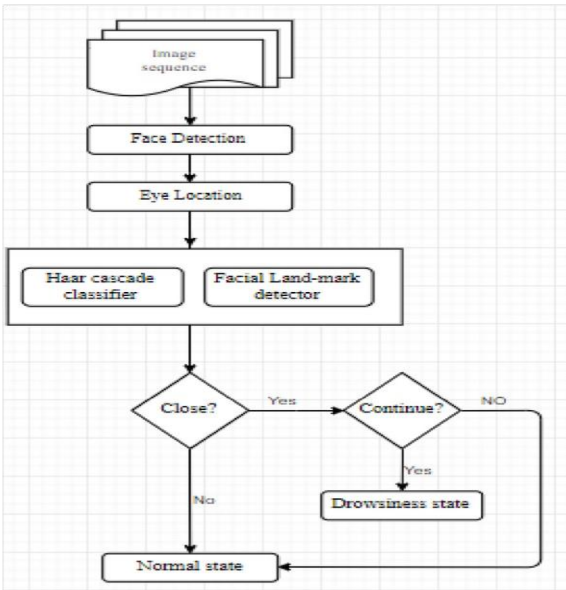
Alert System

- When drowsiness is detected:
  - An **audio alarm** (buzzer/speaker) is triggered to alert the driver.
  - Optionally, **vibration modules** or **visual alerts** (LED or on- screen warning) can be integrated.

Optional Enhancements

- Integration with **cloud storage** for logging drowsiness events
- Real-time data display on a **dashboard GUI.**

4. SYSTEM ARCHITECTURE



Face Detection: For the face Detection it uses
Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper,

"Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle. Fig. 3.4 represents five haar like features & example is shown in Fig.3.5

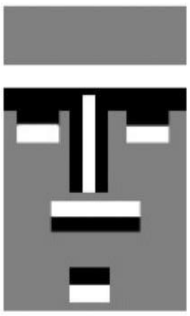


Fig 3.4

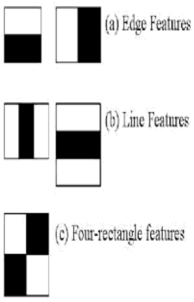


Fig 3.5

1. Eye detection: In the system we have used facial landmark prediction for eye detection Facial landmarks are used to localize and represent salient regions of the face, such as:

- Eyes
- Eyebrows
- Nose
- Mouth
- Jawline

Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more. In the context of facial landmarks, our goal is detecting important facial structures on the face using shape prediction methods. Detecting facial landmarks is therefore a twostep process:

- Localize the face in the image.
- Detect the key facial structures on the face ROI.

Localize the face in the image: The face image is localized by Haar feature-based cascade classifiers which was discussed in the first step of our algorithm i.e. face detection.

Localize the face in the image: The face image is localized by Haar feature-based cascade classifiers which was discussed in the first step of our algorithm i.e. face detection.

Detect the key facial structures on the face ROI: There are a variety of facial landmark detectors, but all methods essentially try to localize and label the following facial regions:

- Mouth
- Right eyebrow
- Left eyebrow
- Right eye
- Left eye
- Nose

The facial landmark detector included in the dlib library is an implementation of the One Millisecond Face Alignment with an Ensemble of Regression Trees paper by Kazemi and Sullivan (2014).

This method starts by using:

1. A training set of labelled facial landmarks on an image. These images are manually labelled, specifying specific (x, y)-coordinates of regions surrounding each facial structure.
2. Priors, of more specifically, the probability on distance between pairs of input pixels.

The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face.

The indexes of the 68 coordinates can be visualized on the image below:

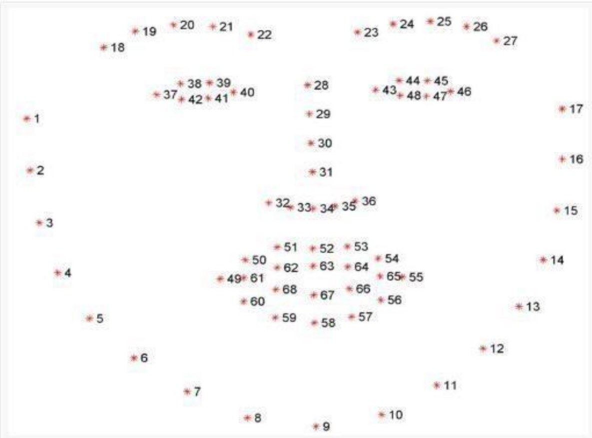


Fig.4.1 Visualizing the 68 facial landmark coordinates

We can detect and access both the eye region by the following facial landmark index show below

- The right eye using [36, 42].
- The left eye with [42, 48].

These annotations are part of the 68 point iBUG 300-W dataset which the dlib facial landmark predictor was trained on. It's important to note that other flavors of facial landmark detectors exist, including the 194 point model that can be trained on the HELEN dataset. Regardless of which dataset is used, the same dlib framework can be leveraged to train a shape predictor on the input training data.

2. Recognition of Eye's State:

The eye area can be estimated from optical flow, by sparse tracking or by frame-to-frame intensity differencing and adaptive thresholding. And Finally, a decision is made whether the eyes are or are not covered by eyelids. A different approach is to infer the state of the eye opening from a single image, as

e.g. by correlation matching with open and closed eye templates, a heuristic horizontal or vertical image intensity projection over the eye region, a parametric model fitting to find the eyelids, or active shape models. A major drawback of the previous approaches is that they usually implicitly impose too strong requirements on the setup, in the sense of a relative face-camera pose (head orientation), image resolution, illumination, motion dynamics, etc. Especially the heuristic methods that use raw image intensity are likely to be very sensitive despite their real-time performance.

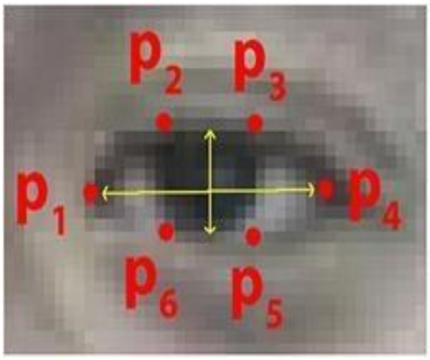
Therefore, we propose a simple but efficient algorithm to detect eye blinks by using a recent facial landmark detector. A single scalar quantity that reflects a level of the eye opening is derived from the landmarks. Finally, having a per-frame sequence of the eye- opening estimates, the eye blinks are found by an SVM classifier that is trained on examples of blinking and non-blinking patterns.

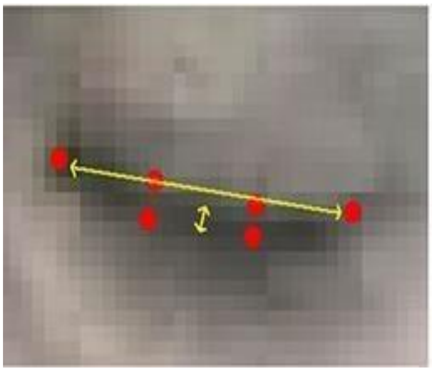
Eye Aspect Ratio Calculation:

For every video frame, the eye landmarks are detected. The eye aspect ratio (EAR) between height and width of the eye is computed. EAR = ||p2 − p6|| + ||p3 − p5|| (1) 2||p1 − p4||

where p1, . . ., p6 are the 2D landmark locations, depicted in Fig. 1. The EAR is mostly constant when an eye is open and is getting close to zero while closing an eye. It is partially person and head pose insensitive. Aspect ratio of the open eye has a small variance among individuals, and it is fully invariant to a uniform scaling of the image and in-plane rotation of the face. Since eye blinking is performed by both eyes synchronously, the EAR of both eyes is averaged.

Open and closed eyes with landmarks p(i) automatically detected. The eye aspect ratio EAR in Eq. (1) plotted for several frames of a video sequence.





3.**Eye State Determination:**

Finally, the decision for the eye state is made based on EAR calculated in the previous step.

If the distance is zero or is close to zero, the eye state is classified as “closed” otherwise the eye state is identified as “open”.

5.RESULT

Implementation of drowsiness detection with Python and OpenCV was done which includes the following steps: Successful runtime capturing of video with camera. Captured video was divided into frames and each frame were analyzed. Successful detection of face followed by detection of eye. If closure of eye for successive frames were detected, then it is classified as drowsy condition else it is regarded as normal blink and the loop of capturing image and analysing the state of driver is carried out again and again. In this implementation during the drowsy state the eye is not surrounded by circle or it is not detected, and corresponding message is shown.

Screenshots of the system shown below:

Figure 5.1: When a driver is awake

Figure 5.2: When a driver closes the eye to sleep.

6.CONCLUSION

Now a day, the accident ratio has been increased in large amount. Especially in a country like India, the rate has become more, and the main reason for these accidents to cause is drivers fatigue or drowsiness. We should always remember life gives us only one chance. Therefore, this is a device which would help a huge amount of percent from accidents. This system is used to detect drowsiness while driving by giving a buzzer and also alerting people by alert messages.

A non-invasive system to localize the eyes and monitor fatigue was developed. Information about the eyes position is obtained through various self-developed image processing algorithms. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for too long, a warning signal is issued. In addition, during monitoring, the system is able to automatically detect any eye localizing error that might have occurred. In case of this type of error, the system is able to recover and properly localize the eyes.

REFERENCES

[1] Facial Features Monitoring for Real Time Drowsiness Detection by Manu B.N, 2016 12th International Conference on Innovations in Information Technology (IIT) [Pg. 78-81] https://ieeexplore.ieee.org/ document/7880030

[2] Real Time Drowsiness Detection using Eye Blink Monitoring by Amna Rahman Department of Software EngineeringFatimaJinnah Women University 2015 National Softwarehttps://ieeexplore.ieee. org/document/7396336

[3] D. T. C. Cox and K. J. Gaston, ‘‘Likeability of garden birds: Importance of species knowledge & richness in connecting people to nature,’’ *PloS one*, vol. 10, no. 11, Nov. 2015, Art. no. e0141505.