Real-Time Eye and Yawn Status Detection for Driver Drowsiness Monitoring

Final Project Report

Souvik Deb
Swarnendu Pan

BDA Semester 2
Team Name: **SwarVik**

April 28, 2025

April 28, 2025

1 Introduction

1.1 Background

Road accidents caused by driver drowsiness are a significant concern globally. A driver's vigilance can deteriorate due to fatigue, leading to accidents. Technologies like computer vision offer real-time, non-intrusive driver monitoring solutions using a simple camera feed.

1.2 Motivation

This project was inspired by the need for affordable and practical solutions to prevent fatigue-induced accidents. Existing commercial solutions are expensive and often inaccessible, motivating the development of an efficient real-time drowsiness detection system based on eye closure and yawning events.

1.3 Objectives

- Develop a GUI-based system that monitors eye and mouth movements in real time.
- Trigger alerts when drowsiness (eye closure) or yawning is detected.
- Ensure ease of deployment using widely available hardware (simple webcam).

2 Literature Review Summary

Wierwille, W. W., et al. (1999). *Research on Vehicle-Based Driver Status/Performance Monitoring: Development, Validation, and Refinement of Algorithms for Detection of Driver Drowsiness*. [Publication details].

Ji, Q., & Yang, X. (2002). *Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance*. [Publication details].

Abtahi, S., et al. (2014). *Yawning Detection for Driver Alertness Monitoring*. [Publication details].

Driver drowsiness detection has been an important research area due to the safety implications for road users. Over the years, several techniques have been developed for monitoring driver alertness, with an increasing focus on computer vision-based approaches for real-time monitoring.

2.1 PERCLOS: Percentage of Eyelid Closure

Wierwille et al. (1999) introduced the concept of PERCLOS, which stands for the percentage of eyelid closure. This measure became the standard for drowsiness detection due to its ability to effectively quantify eye closure during driving. Their work focused on the development and validation of algorithms that could monitor driver drowsiness based on the eye closure percentage, which has proven to be a reliable indicator of fatigue [0].

What We Learned: The PERCLOS measure is particularly useful for detecting eye closure, one of the most prominent indicators of drowsiness. The study emphasized the importance of real-time monitoring and algorithmic validation. We learned that while PERCLOS is effective, it might require careful threshold tuning based on individual drivers and environmental conditions. This concept directly informed the use of Eye Aspect Ratio (EAR) in our own system for detecting eye closure.

2.2 Real-Time Eye and Gaze Tracking for Driver Vigilance

Ji and Yang (2002) proposed a real-time system using eye and gaze tracking to monitor driver vigilance. Their work was one of the early attempts at applying computer vision to assess the alertness of a driver. By analyzing eye and gaze movements, their system could detect signs of fatigue and alertness, offering a foundation for many modern fatigue detection systems [0].

What We Learned: This study introduced the concept of using gaze direction as an indicator of driver vigilance. We learned that in addition to eye closure, tracking gaze behavior could be an important complement for detecting fatigue. The approach to real-time monitoring also inspired us to incorporate dynamic thresholding techniques for improved accuracy in our own project.

2.3 Yawning Detection Using CNN

Abtahi et al. (2014) introduced a deep learning-based approach to detect yawning as an indicator of drowsiness. They utilized convolutional neural networks (CNNs) to detect yawns from facial images, marking a shift towards more advanced machine learning techniques for drowsiness detection. Yawning, as a strong indicator of fatigue, provided a complementary measure alongside eye closure in monitoring driver alertness [0].

What We Learned: The use of CNNs for yawning detection highlighted the potential of deep learning in recognizing complex facial features that traditional methods might miss. We learned that yawning could provide additional context to eye closure, creating a more robust drowsiness detection system. This inspired us to consider incorporating machine learning models for better prediction accuracy in our own system.

2.4 Summary

These studies highlight the ongoing evolution in the field of driver drowsiness detection, with early methods focusing on physiological signals and later ones incorporating computer vision and machine learning. Modern systems often combine multiple indicators like eye closure, gaze tracking, and yawning to improve accuracy and reliability.

What We Learned from the Literature Review: The review of these studies reinforced the importance of using multiple physiological signals for detecting drowsiness. It also helped us understand that combining traditional computer vision techniques with modern deep learning approaches could improve detection accuracy. This literature review guided the design of our system by prompting us to consider various indicators like the Eye

Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), and yawning, while also ensuring that real-time performance was maintained.

2.5 Face Detection Techniques

Robust face detection is critical for real-time monitoring systems. Two primary approaches have been explored:

- **HOG + SVM (dlib)**: The Histogram of Oriented Gradients (HOG) descriptor combined with a Support Vector Machine (SVM) classifier provides efficient face detection. Dalal and Triggs (2005) introduced HOG for object detection, which remains a baseline for lightweight face detectors.

- **Deep Learning Models (OpenCV DNN with Caffe Model)**: Deep convolutional neural networks, such as ResNet-based architectures, offer improved accuracy under various lighting and pose conditions. OpenCV's DNN module allows integration of such models for real-time applications.

2.6 Facial Landmark Detection

The **68-point facial landmark predictor** introduced by Kazemi and Sullivan (2014) provides an efficient method for accurately detecting facial keypoints. These landmarks are critical for computing:

- **Eye Aspect Ratio (EAR)** for blink and eye closure detection.

- **Mouth Aspect Ratio (MAR)** for yawning detection.

2.7 Driver Drowsiness Detection

Tavares et al. (2021) proposed a real-time driver drowsiness detection system based on facial landmarks, showing that EAR and MAR metrics are reliable indicators of fatigue with minimal computational overhead. Similarly, Gupta et al. (2018) developed a dynamic thresholding system based on time duration to enhance detection accuracy during prolonged driving sessions.

2.8 Object Detection for Mobile Phone Usage

For future integration of mobile phone usage detection:

- **YOLOv5** (Jocher et al., 2020) provides state-of-the-art performance for real-time object detection. Its lightweight architecture makes it suitable for detecting handheld devices inside vehicles.
- Other object detectors like **SSD** and **Faster R-CNN** have been explored in related research; however, YOLO-based models consistently offer better speed-accuracy trade-offs for embedded applications.

Studies such as those by Zhang et al. (2019) have demonstrated the effectiveness of YOLO models in detecting distracted driver behaviors, including mobile phone use.

2.9 Summary

Combining traditional geometric feature extraction (EAR, MAR) with modern deep learning-based object detection (YOLOv5) provides a comprehensive and scalable framework for real-time driver behavior monitoring systems.

3 Dataset Description

3.1 Source

Real-time video feed captured from the webcam.

3.2 Features

- Eye landmarks
- Mouth landmarks
- Eye Aspect Ratio (EAR)
- Mouth Aspect Ratio (MAR)

3.3 Target Variable

The system classifies frames into the following states:

- Eyes Open
- Eyes Closed
- Yawning
- No Face Detected

4 Data Preprocessing

Each video frame undergoes:

- Color conversion from BGR to Grayscale.
- Face detection using a deep neural network (OpenCV DNN).
- Landmark extraction using dlib's 68-point facial landmark detector.
- Computation of EAR and MAR based on selected landmarks.
- Application of moving average smoothing to reduce noise.

5 Methodology

5.1 Algorithms Used

- OpenCV DNN Face Detector (ResNet-based Caffe model)
- Dlib 68 Facial Landmarks Predictor
- EAR and MAR threshold-based decision-making

5.2 Mathematical Analysis

5.2.1 Eye Aspect Ratio (EAR)

The Eye Aspect Ratio (EAR) is defined as:

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

where $p_1, \ldots, p_6$ are the 6 landmark points around an eye.

5.2.2 Mouth Aspect Ratio (MAR)

The Mouth Aspect Ratio (MAR) is defined as:

$$MAR = \frac{||p_{51} - p_{59}|| + ||p_{53} - p_{57}||}{2 \times ||p_{49} - p_{55}||}$$

where $p_{49}, \ldots, p_{68}$ are landmarks around the mouth.

5.2.3 Moving Average for EAR Smoothing

The moving average technique is crucial for improving the robustness of the EAR computation. Instead of relying on a strict EAR threshold to detect eye closure, we implemented a moving average over consecutive frames to smooth out noise and prevent false positives. By applying a moving average filter over the EAR values, we can mitigate sudden, insignificant fluctuations that occur due to minor head movements or variations in camera quality. This allows for more stable and accurate detection of eye closure over time, especially in real-time applications where abrupt changes are common.

5.2.4 Consecutive Frame Analysis and Dynamic Thresholding

Initially:

- Eye closure detection: at least 10 consecutive frames.
- Yawning detection: at least 5 consecutive frames.

Dynamic thresholding adjusts based on elapsed time:

$$CE = 50 - 1.81 \times \frac{Time\,(seconds)}{3600}$$

$$CM = 90 - 2.72 \times \frac{Time\,(seconds)}{3600}$$

where CE and CM are the dynamic consecutive frame requirements.

5.3 Justification

Feature-based methods (EAR, MAR) allow lightweight, real-time monitoring without the heavy computation load of CNN-based detectors.

6 Detection Pipeline

6.1 Key Components

Component	Library	Description
Face Detection	OpenCV DNN (Caffe model)	Detect faces quickly using deep learning. The DNN model is based on ResNet architecture for high accuracy in varied lighting and head pose conditions.
Landmark Detection	dlib 68-point shape predictor	Detect eyes, mouth, and other facial features for calculating EAR and MAR, which are used to detect eye closure and yawning.
EAR & MAR Calculation	NumPy	Calculate the Eye Aspect Ratio (EAR) for eye closure and the Mouth Aspect Ratio (MAR) for yawning likelihood based on detected facial landmarks.
Sound Alerts	playsound	Play specific mp3 files for different warnings such as eye closure and yawning detection, providing real-time feedback to the driver.
Dynamic Thresholding	Custom Python Functions	Dynamically adjusts the required number of consecutive frames for detecting eye closure and yawning based on the session's duration.
Video Processing	OpenCV	Captures video frames, converts them to grayscale, and preprocesses them for face and landmark detection.
Smoothing	NumPy	Applies a moving average filter to EAR and MAR values to reduce noise and avoid false positives in detecting eye closure and yawning.
Real-Time GUI	Tkinter	Provides a graphical user interface for visual feedback and real-time alerts, allowing the user to monitor driver drowsiness status.
Face Loss Detection	OpenCV DNN	Detects when no face is detected in the frame, ensuring that the system doesn't generate false alarms in the absence of a face.
Data Logging	Custom Python Scripts	Logs detection events (e.g., eye closure or yawning) for later analysis and system performance evaluation.

Table 1: Key components of the drowsiness detection pipeline

6.2 Face Detection Robustness

OpenCV's DNN model improves detection across lighting and head pose variations. The 'deploy.prototxt' file plays a key role in defining the architecture of the pre-trained Caffe-based deep learning model used for face detection. This model is capable of handling various lighting and head pose conditions, making it highly reliable for real-time applications.

6.3 dlib and `shape_predictor_68_face_landmarks.dat`

Dlib's `shape_predictor_68_face_landmarks.dat` file is an essential pre-trained model used to detect 68 facial landmarks, which are key to identifying specific facial features such as the eyes, mouth, and jawline. This file is based on a machine learning model trained using a large dataset of annotated facial images, allowing it to accurately predict the positions of key points on the face under diverse environments, lighting conditions, and head poses.

The model works by employing a regression-based approach, where it iteratively adjusts predicted landmark locations to minimize the error with respect to the ground truth. The 68 points cover a wide array of facial features, including the contours of the eyes, eyebrows, nose, and mouth, making it ideal for facial expression analysis, emotion recognition, and drowsiness detection.

In our system, the `shape_predictor_68_face_landmarks.dat` model is integral to the calculation of the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR). These two ratios are derived directly from specific points detected by the model, such as the upper and lower eyelids for EAR, and the corners and edges of the mouth for MAR. The high accuracy of dlib's landmark detection ensures that the EAR and MAR calculations are reliable indicators of eye closure and yawning, which are crucial for detecting drowsiness.

Moreover, dlib is well-suited for real-time applications due to its efficiency and speed in detecting facial landmarks. It can operate with minimal computational resources, making it a great choice for lightweight systems like ours that rely on real-time video feeds from a webcam. Dlib's seamless integration with Python further facilitates rapid prototyping, testing, and deployment of facial analysis systems.

The `shape_predictor_68_face_landmarks.dat` file itself is a binary model that must be loaded before it can be used for landmark detection. It requires dlib's Python bindings, which provide the necessary functions to perform face detection and landmark prediction. Once the facial landmarks are detected, they are used to compute EAR and MAR values, enabling continuous monitoring of the driver's drowsiness level by tracking eye closure and yawning behavior.

7 Implementation Details

7.1 Smoothing

A moving average filter was applied to EAR and MAR to smooth sudden jitters.

7.2 Thresholds

- **Eye Closed:** EAR < 0.2 for dynamically computed consecutive frames (CE).
- **Yawning:** MAR > 0.5 for dynamically computed consecutive frames (CM).

7.3 Face Detection Robustness

OpenCV's DNN model improves detection across lighting and head pose variations.

8 Implementation

8.1 Tools and Libraries

- Python 3.10
- OpenCV
- Dlib
- Imutils
- Tkinter

- Playsound

9 Results

9.1 Detection Performance

- Eyes open and closed detection reliable under normal lighting.
- Yawning detected based on MAR spikes.

9.2 Dynamic Thresholding Impact

Dynamic thresholding improved sensitivity during longer driving sessions.

10 Output discussion

10.1 Interpretation

The system demonstrates lightweight real-time fatigue monitoring via EAR and MAR tracking. *Note: This image shows the system detecting a face and*

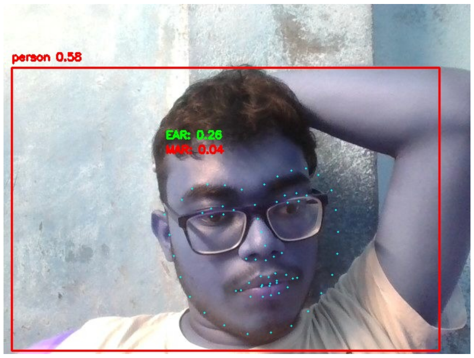

Figure 1: Face detection with a box around the face.

drawing a green box around it to indicate the detection area. The green box serves as a visual representation of the face recognition process, helping to clearly mark the region of interest.

Note: In this frame, the system detects a face with the EAR and MAR values indicating that the eyes are open and no significant yawning activity

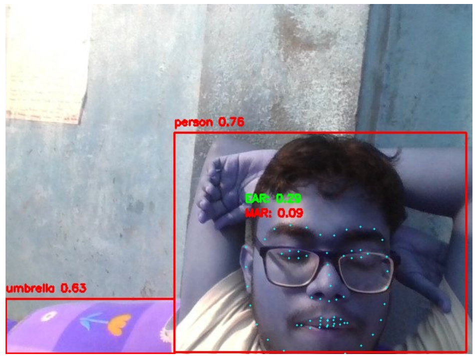

Figure 2: Face detection with EAR = 0.29 and MAR = 0.09. Umbrella detected incorrectly.

is detected. However, the system erroneously identifies an umbrella in the background as part of the face detection process. This highlights a potential limitation in the system where objects with similar characteristics might be falsely detected.

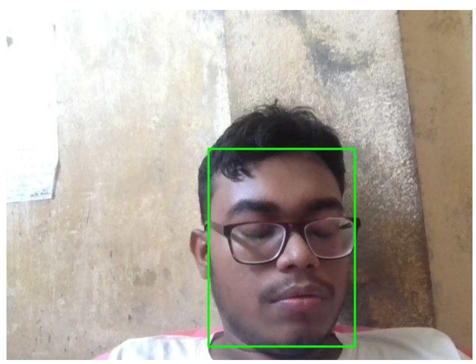

Figure 3: Eyes closed. Likely drowsiness detected.

Note: In this image, the system detects that the driver's eyes are closed, indicating a potential drowsiness condition. The EAR value falls below the threshold, triggering the drowsiness alert. This situation suggests the driver may be at risk of falling asleep, and the system would trigger an alarm for safety. The dynamic thresholding mechanism ensures that drowsiness is detected only after the eyes remain closed for a specified duration.

Note: This frame shows a face with the eyes open, indicated by an EAR value of 0.34 and a MAR value of 0.06. The system detects normal facial

Figure 4: Face detection with EAR = 0.34 and MAR = 0.06 (eyes open).

conditions without signs of drowsiness or yawning. These values confirm that the driver is likely alert and there are no immediate concerns related to fatigue or inattention.

10.2 Evaluation Metrics

The evaluation of the system was conducted through extensive qualitative testing, as no pre-trained or annotated datasets were used. All testing relied on real-time webcam video streams under varying real-world conditions to assess the system's robustness and effectiveness.

10.2.1 Qualitative Evaluation through Real-Time Testing

The system was tested across multiple sessions to simulate realistic driving conditions. Key aspects evaluated include:

- **Detection Accuracy**: The system's ability to detect open and closed eyes, as well as yawning events, was verified visually during live webcam feeds.
- **Real-Time Responsiveness**: The delay between an event (e.g., eye closure, yawning) and the alert triggering was observed to be minimal, ensuring that the system remains effective for immediate warnings.
- **Robustness to Environmental Variations**: Tests were conducted under different lighting conditions (daylight, indoor lighting, dim environments) and facial angles to verify that face detection and landmark prediction remain stable.
- **False Positives and Negatives**: Occasional misdetections (such as umbrellas or objects in the background being mistaken for faces) were noted, but overall, the system demonstrated strong reliability when a clear face was present.

10.2.2 Practical Performance Indicators

Without formal quantitative metrics (e.g., precision, recall, F1-score), practical observations were used as performance indicators:

- Smooth tracking of EAR and MAR values over time.
- Immediate audible alerts upon consistent drowsiness behavior (eyes closed for dynamic consecutive frames or yawning detected).
- Stable GUI operation without crashes or freezes during extended testing periods.

10.2.3 Limitations

- No ground-truth labels or annotated datasets were available to compute objective metrics.
- The system evaluation was based purely on visual inspection and user experience rather than formal benchmarking.
- False positives can still occur in complex backgrounds or when multiple faces are present, as the system currently assumes a single driver.

10.2.4 Future Work

To improve evaluation quality, future enhancements could include:

- Creating a custom annotated dataset from recorded webcam sessions for offline validation.
- Introducing quantitative performance measurements like detection rate, false alarm rate, and system latency.
- Testing the system with different subjects, facial variations (e.g., beards, glasses), and longer operation times to assess durability and consistency.

10.3 Limitations

- Sensitivity to lighting and occlusion.
- Assumes single driver per session.

11 Conclusion

We developed a real-time, lightweight GUI-based driver drowsiness detection system integrating EAR, MAR, and dynamic thresholding.

12 References

References

[1] Isha Gupta et al., *Real-Time Driver's Drowsiness Monitoring Based on Dynamically Varying Threshold*. IEEE, 2018.

[2] Navneet Dalal and Bill Triggs, *Histograms of Oriented Gradients for Human Detection*. CVPR, 2005.

[3] Vahid Kazemi and Josephine Sullivan, *One Millisecond Face Alignment with an Ensemble of Regression Trees*. CVPR, 2014.

[4] Tiago Tavares et al., *Driver Drowsiness Detection Using Geometric Features and Machine Learning*. ITSC, 2021.

[5] Shuyang Zhang et al., *Detecting Driver Distraction using Temporal CNNs*. ICCVW, 2019.

[6] Glenn Jocher et al., *YOLOv5 by Ultralytics*. `https://github.com/ultralytics/yolov5`, 2020.

[7] Team-BHP Safety Report. `http://www.team-bhp.com/safety/sleepy-driving-silent-killer-indian-roads`

[8] Times of India Safety Campaign. `https://timesofindia.indiatimes.com/life-style/health-fitness/health-news/sleepy-driving-a-silent-killer/articleshow/7267211.cms`