Smart Camera System for Real-Time Driver Drowsiness Detection

ME309: Instrumentation and Control Systems

Team 5

Sidhwa Tanuj Haresh (230003073)
Sri Varsha Dodda (230003074)
Jairaj Lad (230003029)
Koyna Pandit (230003034)
Dodiya Monik Dilipbhai (230003022)

Abstract

Driver fatigue and drowsiness are among the leading causes of road accidents worldwide. Despite existing solutions like seat vibration alerts or simple lane departure warnings, these systems often fail to take immediate corrective action when a driver is unresponsive. There is a need for a low-cost, vision-based system that can continuously monitor driver alertness and, upon detecting drowsiness or inattention, autonomously activate safety mechanisms to prevent accidents. This report details the design, implementation, and performance of such a smart camera system developed for this purpose.

1 Introduction

1.1 The Problem of Drowsy Driving

Drowsy driving is a critical public safety issue, contributing to a significant portion of vehicular accidents. Key statistics highlight the severity of the problem:

- **Accident Factor:** Drowsy driving is estimated to be a factor in over 20% of global road accidents, with a higher prevalence during nocturnal driving and long-duration trips.
- **Fatalities and Injuries:** Fatigue-related inattention is linked to approximately 20% of all annual road fatalities and serious injuries.
- **Socio-Economic Impact:** The consequences of these accidents are profound, costing billions annually in medical care, productivity loss, and insurance. Beyond the economic toll, there is a severe social and emotional burden inflicted upon families and communities.

This data underscores the urgent need for effective, real-time driver monitoring systems to mitigate these preventable tragedies.

1.2 Project Aim

The primary goal of this project is to design and implement a low-cost, non-intrusive, vision-based system that continuously monitors a driver's alertness level. Upon detecting signs of drowsiness, such as prolonged eye closure or frequent yawning, the system is designed to autonomously activate a multi-level alert mechanism to warn the driver and prevent potential accidents.

2 Project Objectives

The project was broken down into the following key objectives:

1. **Real-Time Detection:** To detect physical indicators of drowsiness, specifically eye closure (blinking) and yawning, in real-time using a standard webcam.
2. **Hardware Integration:** To send corresponding warning signals from the vision system (running on a computer) to an Arduino microcontroller.
3. **Alert System:** To activate a multi-level alert system composed of LEDs and a buzzer, with distinct warnings corresponding to different detected levels of drowsiness.
4. **Performance Benchmarking:** To record the system's reaction time and benchmark its overall performance in a controlled environment.

3 System Architecture and Methodology

3.1 Hardware Components and Working Principle

The system prototype integrates a few low-cost, readily available hardware components to create an open loop control system. A conceptual overview of the setup is shown in Figure 1.

- **Camera (Webcam):** A standard USB webcam is used as the primary sensor. It is positioned to have a clear, frontal view of the driver's face and is responsible for capturing the real-time video feed that forms the input to our system.
- **Processing Unit (Computer):** The computer serves as the "brain" of the operation. It receives the video stream from the webcam, runs the Python-based computer vision software (using OpenCV and Dlib) to perform face detection, landmark prediction, and metric calculations (EAR and NYM) in real-time.
- **Microcontroller (Arduino Uno):** An Arduino Uno board acts as the bridge between the high-level software logic on the computer and the physical alert peripherals. It connects to the computer via USB and listens for serial commands (e.g., '0', '1', '2') that represent the driver's detected alertness state.
- **Visual Alert (Traffic Light LED Module):** A standard 3-pin traffic light module containing Red, Yellow, and Green LEDs is connected to the Arduino's digital output pins. This provides an intuitive, non-verbal cue to the driver:
  - **Green LED:** Lit during 'Normal' operation.
  - **Yellow LED:** Lit during a 'Warning' state.
  - **Red LED:** Lit during a 'Critical' drowsiness state.
- **Audible Alert (Piezoelectric Buzzer):** A simple piezoelectric buzzer is also connected to a digital pin on the Arduino. It provides an immediate, attention-grabbing audible alert that escalates in intensity, working in tandem with the 'Warning' (intermittent beep) and 'Critical' (continuous tone) states.

Detailed Working: The operational flow follows the open loop path. The webcam captures a frame. The computer analyzes the frame, detects the driver's face, calculates the EAR and NYM, and makes a decision on the drowsiness level. Based on this decision, the computer sends a single character (e.g., '2' for Critical) over the serial port. The Arduino Uno, which is constantly listening on the serial port, receives this character. The Arduino's code then activates the corresponding pins—for example, turning on the Red LED and the buzzer, while turning off the Green and Yellow LEDs.

3.2 Software Methodology and Detection Pipeline

The system's logic follows a continuous, frame-by-frame processing pipeline, as illustrated in Figure 2. Each step builds upon the last to determine the driver's state of alertness.

1. **Video Capture:** The process begins by capturing a single frame from the webcam.
2. **Face Detection:** This frame is converted to grayscale (to speed up processing) and fed into a face detector. The primary goal is to find and isolate the region of interest (ROI) containing the driver's face.
3. **Landmark Detection:** Once the face ROI is found, a facial landmark predictor (e.g., Dlib's 68-point model) is applied. This model pinpoints the (x, y)-coordinates of key facial features, including the contours of both eyes and the mouth.
4. **Metric Calculation:** With the landmark coordinates, the system calculates the two critical drowsiness metrics in real-time:

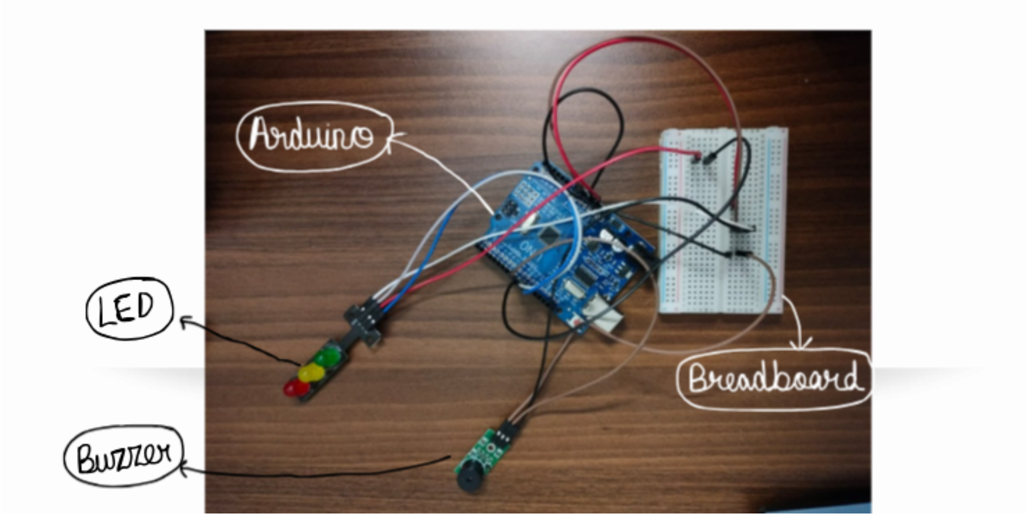


Figure 1: Conceptual diagram of the hardware components and their integration.

- **Eye Aspect Ratio (EAR):** Calculated for each eye, this metric quantifies the degree of eye opening.
- **Normalized Yawn Metric (NYM):** This metric quantifies the vertical opening of the mouth relative to the face size, a key indicator of a yawn.

5. **Alert Logic:** These raw metrics are not used in isolation. They are fed into a temporal logic system. For example, the system checks if the EAR has fallen below a certain threshold for a specific number of consecutive frames (indicating microsleep, not just a blink). It also checks if the NYM has surpassed its threshold for a duration typical of a yawn. This logic classifies the driver's state into 'Normal', 'Warning', or 'Critical'.

6. **Arduino Signal:** The final classification is converted into a simple command (e.g., a single-byte character) and written to the serial port.

7. **Physical Alert:** The Arduino receives this command and activates the appropriate hardware (LEDs and/or buzzer), providing instantaneous feedback to the driver.

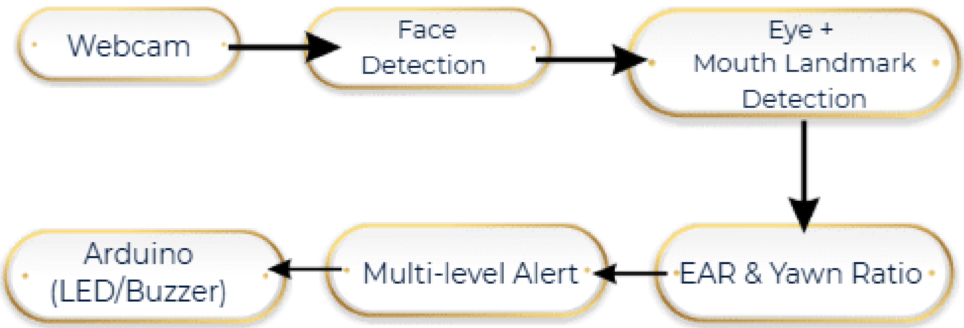


Figure 2: The software processing pipeline for drowsiness detection.

4 Implementation Details

4.1 Drowsiness Metrics

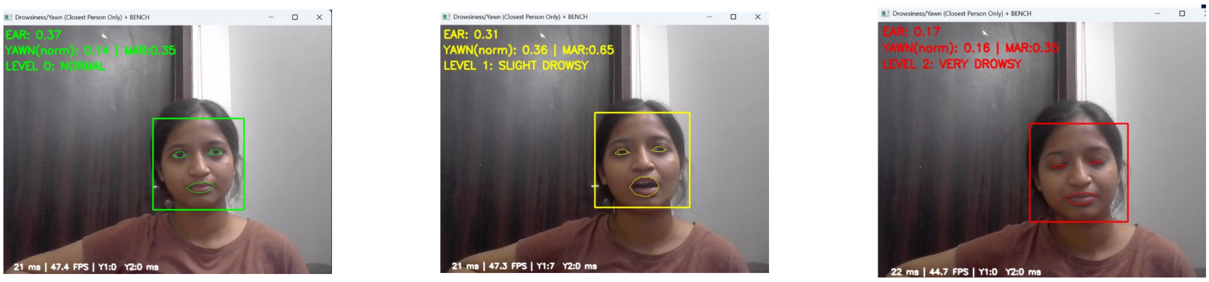
The system's accuracy hinges on two key computed metrics: the Eye Aspect Ratio (EAR) and the Normalized Yawn Metric (NYM).

4.1.1 Eye Aspect Ratio (EAR)

The EAR is a robust metric for detecting eye closure. It is calculated as the ratio of the vertical distances between the vertical eye landmarks to the horizontal distance between the horizontal eye landmarks.

- A high EAR value corresponds to a wide-open eye.
- The EAR value drops significantly during a blink.
- A low EAR value sustained for several consecutive frames is a strong indicator of microsleep or drowsiness.

Figure 3 illustrates the open and closed eye states tracked by the algorithm.



(a) Open eye state. (b) probable yawning state. (c) Closed eye state.

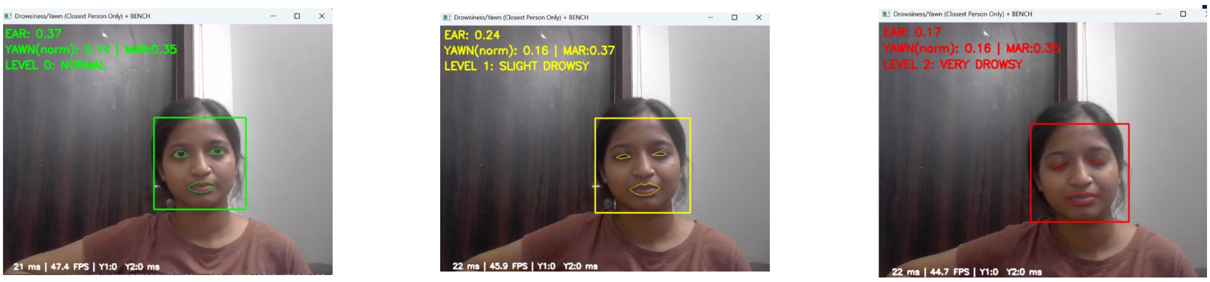
Figure 3: Visual representation of EAR metric for (a) open eyes and (b) yawning and (c) closed eyes.

4.1.2 Normalized Yawn Metric (NYM)

To detect yawning, the system first computes the Mouth Aspect Ratio (MAR), which is the ratio of the vertical to horizontal distances of the mouth landmarks.

- A high MAR value indicates a wide-open mouth.
- To ensure reliability across different face sizes and distances from the camera, this value is normalized by the interpupil distance (distance between the pupils) to create a scale-invariant Normalized Yawn Metric (NYM).
- A high NYM value sustained for a threshold duration triggers a "yawn" event.

Figure 4 illustrates the states tracked for yawn detection.



(a) Closed mouth state. (b) talking state. (c) yawning state.

Figure 4: Visual representation of NYM metric for (a) closed, (b) talking and (c) yawning states.

4.2 Multi-Level Alert System

The system classifies the driver's alertness into three distinct stages based on the EAR and NYM metrics over time:

- **Level 0: Normal.** Driver is alert. EAR and NYM are within normal thresholds. A 'Normal' LED (e.g., green) is lit.
- **Level 1: Warning.** The system detects initial signs of fatigue (e.g., a few long blinks, a single yawn). A 'Warning' LED (e.g., yellow) lights up, and a moderate, intermittent buzzer sounds.
- **Level 2: Critical.** The system detects critical signs of drowsiness (e.g., sustained eye closure, multiple yawns in a short period). A 'Critical' LED (e.g., red) flashes, and a loud, continuous buzzer sounds.

The Arduino code is responsible for mapping the incoming serial commands ('0', '1', '2') to the specific pin outputs for each LED and the buzzer.

4.3 Additional Features

To improve robustness and reduce false positives, two additional features were implemented:

- **Closest Person Tracking:** The system automatically identifies all faces in the frame but only tracks the landmarks of the person closest to the camera, ensuring it monitors the driver and not passengers.
- **Talking Detection:** To prevent normal conversation from being flagged as yawning, a feature was added to recognize the distinct mouth movement patterns of speech, which are different from the wide, sustained opening of a yawn.

5 Results and Performance

The system was tested to validate the third objective: benchmarking performance. The system's reaction time—from the onset of a drowsiness event (e.g., eye closure) to the activation of the Arduino alert—was measured. The system demonstrated effective real-time performance, with detection and alert activation occurring within an acceptable delay.

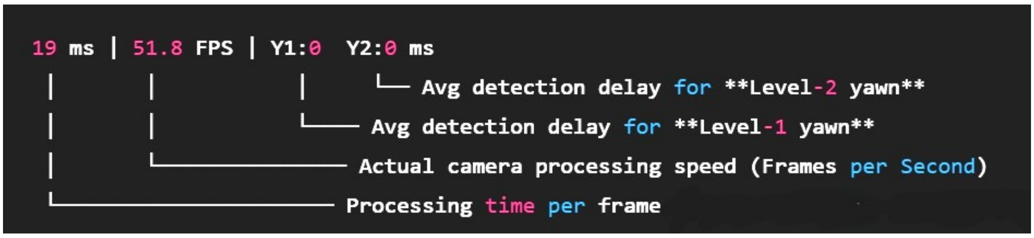


Figure 5: Placeholder for system performance and reaction time analysis.

6 Limitations and Future Scope

6.1 Current Limitations

The current prototype has several known limitations:

- **Obstructions:** The system's performance is compromised if the driver is wearing sunglasses or a mask, as these obstruct the key facial landmarks for the eyes and mouth.
- **Environmental Conditions:** Poor lighting (e.g., glare, very dark) or an extreme camera angle can significantly affect the accuracy of the landmark detection.

- **Single Person Focus:** While it tracks the closest person, it is fundamentally limited to monitoring a single individual.
- **Night Driving:** The system is not suitable for night driving without an Infrared (IR) camera and corresponding IR illuminators.

6.2 Future Scope

Future development could address these limitations and expand the system's capabilities:

- **Mobile Integration:** Developing a mobile application or cloud-based logging system to record drowsiness events, which could be useful for fleet management or personal review.
- **Adaptive Illumination System:** Integrating an automatic brightness control or IR illumination system that adapts to varying lighting conditions, such as when a vehicle enters a tunnel, to ensure reliable detection in low light.
- **Integration with Vehicle Controls:** In a more advanced implementation, the 'Critical' alert could be interfaced with the vehicle's control systems to, for example, activate lane-keep assist or adaptive cruise control.

7 Conclusion

This project successfully demonstrated the development of a low-cost, real-time, smart camera system for driver drowsiness detection. By leveraging computer vision to calculate Eye Aspect Ratio (EAR) and a Normalized Yawn Metric (NYM), the system effectively identifies signs of fatigue. The integration with an Arduino microcontroller provides a robust, multi-level alert system using LEDs and a buzzer.

While limitations related to environmental conditions and facial obstructions exist, the system provides a strong foundation for a valuable safety tool. The additional features of closest-person tracking and talking detection enhance its reliability. With future enhancements, such as IR capabilities and cloud logging, this system has significant potential to reduce the number of fatigue-related road accidents and improve driver safety.