PROJECT ABSTRACT

Driver Drowsiness Detection using Eye Tracking

A safety system that watches the driver's eyes through a webcam, computes the eye-aspect-ratio with facial landmarks and triggers a loud alert when drowsiness is detected. Built with OpenCV and dlib/MediaPipe, it logs drowsy events with timestamps — a socially impactful AI project with a compelling live demo.

01 Title

Driver Drowsiness Detection using Eye Tracking

02 Introduction / Background

Driver fatigue causes a large share of road accidents, yet most Indian vehicles carry no fatigue-monitoring system. Camera-based drowsiness detection closes that gap using hardware everyone already owns: an ordinary webcam. It is a strong final-year project — real-time vision, landmark geometry and threshold logic in a working safety app with a memorable live demo.

03 Problem Statement

Commercial driver monitors are costly and proprietary, while student demos false-alarm on every blink. WakeGuard computes the Eye Aspect Ratio (EAR) from facial landmarks each frame, filters normal blinking with a consecutive-frame counter, and fires a loud alarm only on sustained closure — with every event logged for analysis.

04 Objectives

- Track the driver's eyes in real time from a standard webcam.
- Compute the Eye Aspect Ratio (EAR) per frame from facial landmarks.
- Filter normal blinks with a consecutive-frame counter.
- Fire a loud, configurable alarm only on confirmed drowsiness.
- Log every event to CSV with timestamps and durations.
- Provide a calibration wizard and a live EAR dashboard.

05 Proposed System / Working

Input: a live webcam stream of the driver. Processing: each frame passes through face detection, then a dlib 68-point predictor (or MediaPipe Face Mesh with 468 landmarks) marks six points per eye and computes the Eye Aspect Ratio — open eyes read 0.25–0.30, closed eyes about 0.15. A consecutive-frame counter increments while EAR stays below the calibrated threshold and resets

the moment the eyes reopen, so normal 100–400 ms blinks never accumulate.

Decision and output: only sustained closure — 48+ consecutive frames, roughly 1.5–2 seconds — fires the loud pygame alarm. Every event is logged to CSV with timestamps, duration and minimum EAR, and the live dashboard plots the EAR graph with status and alert history. A calibration wizard measures the driver's personal open- and closed-eye EAR values before monitoring begins.

06 Technologies / Components Used

Python 3.9+ · OpenCV · dlib · MediaPipe · SciPy · NumPy · pygame · CSV logging

07 Key Features

- Real-time eye tracking from any standard webcam.
- EAR computed per frame from dlib 68-point landmarks.
- MediaPipe Face Mesh alternative (468 landmarks).
- Consecutive-frame counter filters normal blinks.
- Loud pygame alarm with configurable sound and volume.
- Live dashboard: EAR graph, status, alert history, CSV event log.
- Calibration wizard for personal EAR thresholds.

08 Applications / Use Cases

- Fleet and long-haul truck fatigue alerts.
- Public-transport driver supervision and mining-vehicle operator safety.
- Fatigue-behaviour research platforms for academic labs.

09 Future Scope

- Fusion with yawning detection (mouth aspect ratio) for multi-cue drowsiness.
- Head-pose tracking to catch distraction as well as drowsiness.
- IR night-vision camera support for low-light cabins.
- Mobile app with Bluetooth wearable buzzer for two-wheeler riders.
- Fleet cloud dashboard with driver-fatigue analytics.

10 Conclusion

WakeGuard turns an ordinary webcam into a real-time drowsiness guard. With blink filtering, per-user calibration and honest evaluation, it is a socially impactful, demo-strong project the student can defend end to end.