Driver Distraction Detection and Alert System

Team & Project Basics

- **Team Name:** FocusDriveX
- **Module Chosen:** Driver Distraction Alert
- **Problem Statement:**

  Drivers often experience or distractions, especially during long or night journeys, which can lead to delayed reaction times and severe road accidents. Our system aims to detect early signs of driver fatigue or distraction and provide passenger-friendly, multi-stage alerts to ensure safety without causing panic.
- **Proposed Solution:**

  Our project uses a camera-based monitoring system with AI-driven facial landmark and head movement analysis to detect signs of distraction in real time. Upon detection, it triggers a **three-level alert system** — gentle airflow with mist spray, seat vibration, and finally an audible alert — ensuring the driver regains focus without alarming passengers. This non-intrusive, multi-stage approach enhances road safety while maintaining comfort for both driver and passengers.

1. Project Overview

The Driver Distraction Detection and Alert System is designed to monitor a driver's attention in real-time using a laptop webcam. It detects eye closure, head movements, mobile usage, and side mirror checking to provide timely alerts and improve driving safety.

Objectives

- Detect driver distraction using Eye Aspect Ratio (EAR) and head orientation.
- Alert the driver in three stages based on distraction duration.
- Detect mobile phone usage and differentiate it from Bluetooth usage.
- Provide visual feedback on the driver's status.

2. System Components

Software Components

- **Python 3.11.9**
- **Libraries:** OpenCV, MediaPipe, NumPy, Pygame

Features

1. **Eye and Face Detection:**
2. Uses MediaPipe Face Mesh.
3. EAR calculation to monitor eye closure.
4. **Distraction Detection:**
5. Detects if the driver is looking away from the road.
6. Ignores side mirror checks up to 10 seconds.
7. **Mobile Usage Detection:**
8. Alerts if the driver uses a mobile device.
9. Ignores Bluetooth headset usage.

   Three Stage Alert:

10. Stage 1 (20 sec): Air flow + water spray simulation.
11. Stage 2 (25 sec): Vibrating seat simulation.
12. Stage 3 (30 sec): Sound alert simulation.

3. Implementation Details

3.1 Eye Aspect Ratio (EAR)

The EAR formula is calculated as:

```
EAR = (||p2 - p6|| + ||p3 - p5||) / (2 * ||p1
```

Where p1-p6 are the specific eye landmark coordinates.

3.2 Head Orientation

- Head direction is inferred from nose tip relative to eye centers.
- Side mirror checking is identified if head turns within a threshold range.

3.3 Alert Logic

- Continuous distraction triggers alerts in three stages.
- Stage timing:
- Stage 1: 20 seconds • Stage 2: 25 seconds
- Stage 3: 30 seconds
- Mobile usage triggers immediate alert.

3.4 Visual Feedback

- Green dots indicate detected eyes.
- Status messages: Focused , Distracted , Checking Side Mirror , Using Mobile .

4. How to Run

1. Ensure Python 3.11.9 is installed.
2. Install required libraries:

```
pip install opencv-python mediapipe numpy pygame
```

3. Place the alert sounds in the project folder.
4. Run the main script:

```
python driver_monitoring.py
```

5. Press `q` to exit.

5. Observations

- The system accurately detects eye closure and distractions.
- Alerts correspond with distraction duration.
- Mobile usage detection works with placeholder logic and can be integrated with YOLO for real implementation.
- Side mirror checks are ignored for 10 seconds to avoid false positives.

6. Future Improvements

- Integrate YOLO for mobile detection.
- Implement real-time Bluetooth headset detection.

- Optimize for hardware alerts (fan, vibration motor, sound) when moving to embedded platforms like Jetson Nano.

End of Document

Thank You