

## 1. Vision-Based Drowsiness Detection

- **Camera Input & Setup:** Real-time video input is captured using a Pi Camera or similar.
- **Facial Landmark Detection:** MediaPipe Face Mesh identifies key facial landmarks (eyes, mouth, nose).
- **Eye State & Blink Detection:** Using landmark points for eyes, PERCLOS (percentage eye closure) and blink detection algorithms determine if eyes are closed or open, signaling drowsiness.
- **Yawn Detection:** Mouth landmarks and aspect ratio are monitored to detect yawning frequency.

## 2. Telemetry & Physiological Data Fusion

- **Steering Telemetry:** steering angle information is read to correlate driving behaviour.
- **Smartwatch Integration:** Heart rate and SpO2 signals are fetched from a connected smartwatch via serial or Bluetooth.
- **Data Fusion:** Scores and detections from vision and behaviour are combined algorithmically in real-time to enhance accuracy and robustness.

## 3. Alert System & Interventions

- **Tiered Alerts:** Based on severity and persistence of detected drowsiness or distraction states, a tiered alert system triggers:
  - Subtle audio buzz or beep
  - Voice prompts or text-to-speech notifications
  - SMS alerts sent via API to emergency contacts.
- **Hardware Alerts:** ESP32 microcontroller controls LEDs and buzzers to provide visual and audible signals.
- **SMS Sending API:** Twilio API is integrated for sending external alerts.

## 4. Privacy & Deployment

- **On-Device Processing:** All critical inferences are done on Raspberry Pi device to retain privacy.
- **Real-Time Edge Computing:** The system is built to run efficiently in real-time on low-powered edge devices.

## 5. Software Architecture

- **Programming Language:** Python is used throughout, including for model inference, telemetry reading, and alert handling.
- **Key Libraries:** OpenCV (image capture/process), MediaPipe (face mesh), TensorFlow Lite (model inference), PySerial (communication), Tkinter & PIL (GUI for testing/development).
- **Modular Design:** Separate modules for vision processing, watch serial data, model prediction, and alerting are implemented for maintainability.

