

AI-Based Sleep-Aware Media & Smart Room Control System

on Raspberry Pi

Team Members: Gayeon Lee · Hyunsoo Lee

Goal: Detect user drowsiness or sleep and automatically control media playback and smart room devices. Users can choose via web interface: Wake-Up Mode (buzzer alert) or Sleep-Aid Mode (auto media & room control).

The Problem

Many people watch videos or listen to music before sleeping. When they fall asleep:



Media keeps playing



Lights may remain on



Power is wasted



Sleep quality can be affected



Our Solution — Two Modes via Web

Via web interface, users choose a mode. In Sleep-Aid Mode, the system automatically:



Pauses media playback



Dims or turns off lights



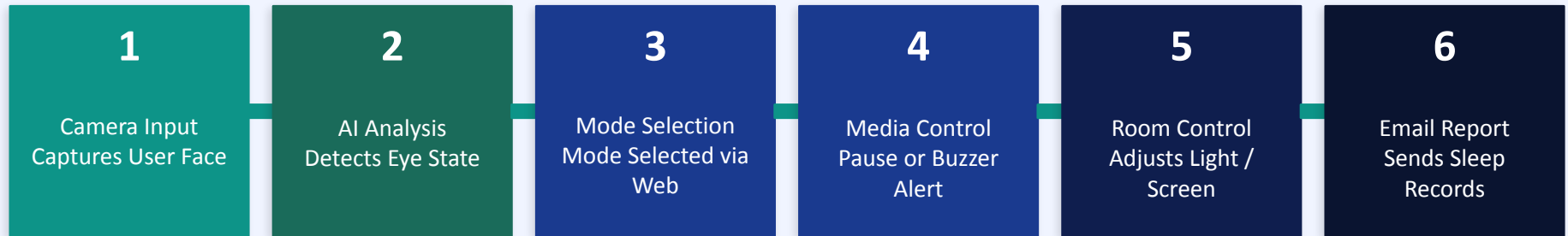
Adjusts screen brightness or turns off the display



Sends sleep status reports by email



Wake-Up Mode: Triggers buzzer alert to wake user



Current Implementation Progress

04

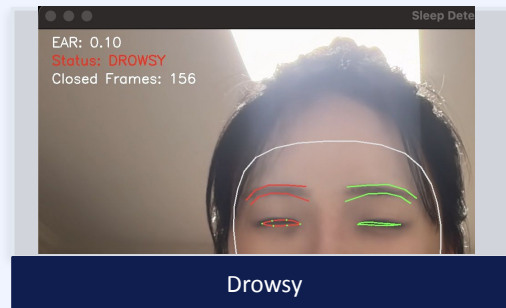
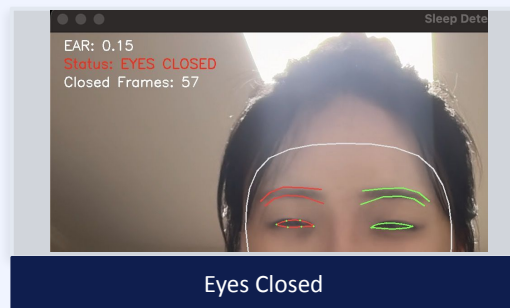
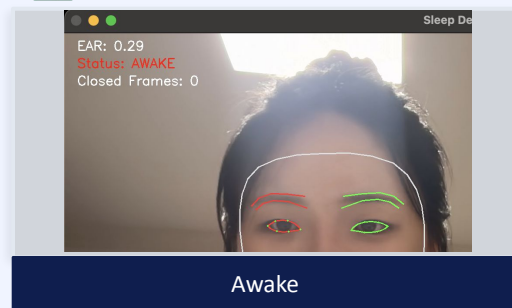
So far, we implemented and tested the core detection logic on a laptop webcam environment.

✓ Completed

- ✓ Real-time camera input was successfully tested
- ✓ MediaPipe Face Mesh was applied to detect facial landmarks
- ✓ Face, eyes, nose, and mouth landmarks recognized in real time
- ✓ Eye landmarks were extracted from the face mesh
- ✓ Eye Aspect Ratio (EAR) was calculated
- ✓ Eye-open and eye-closed states were successfully detected
- ✓ Frame-based drowsiness detection logic was added



Presentation Screenshots



Main Detection Method

- 1 Detect facial landmarks using MediaPipe Face Mesh
- 2 Extract left and right eye landmarks
- 3 Calculate Eye Aspect Ratio (EAR)
- 4 Count how long the eyes remain closed
- 5 Classify state as Awake, Eyes Closed, or Drowsy

$$EAR = (\|p2-p6\| + \|p3-p5\|) / (2 \times \|p1-p4\|)$$

```
sleep_detection.py

# Calculate Eye Aspect Ratio
avg_ear = (ear_left + ear_right) / 2

if avg_ear < EAR_THRESHOLD:
    closed_frames += 1
else:
    closed_frames = 0

if closed_frames >= CLOSED_FRAMES_THRESHOLD:
    status = "DROWSY"
elif closed_frames > 0:
    status = "EYES CLOSED"
else:
    status = "AWAKE"
```

Hardware Components

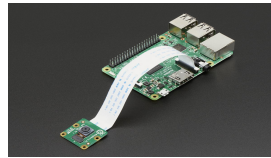
06

Raspberry Pi



Main processing unit

Camera Module



Face / eye capture

LED / RGB LED



Room light simulation

Mini Lamp



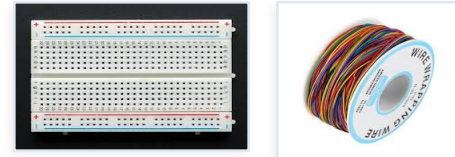
Device power control

Buzzer



Drowsy alert sound

Breadboard & Wires



Circuit assembly



Camera will be used for sleep detection | Mini Lamp will represent smart room control



Issues



Simple eye closure may cause false detection



Raspberry Pi may have limited real-time video processing performance



The mini mood light control method has not been finalized yet.



Solutions

→ Plan to move tested detection code to Raspberry Pi later

→ Will reduce camera resolution or optimize frame processing if speed is slow

→ Considering a safe 5V USB power control method using a relay module or MOSFET module.

Expected Results & Future Plan

08

Expected Results

- ◆ Detect drowsiness or sleep in real time
- ◆ Pause media playback automatically
- ◆ Control lights or room devices automatically
- ◆ Allow users to select Wake-Up or Sleep-Aid mode via web
- ◆ Reduce unnecessary power consumption
- ◆ Add email reporting for sleep logs and drowsiness events

Future Plan

- 1 Test the camera module on Raspberry Pi
- 2 Connect lamp and relay control
- 3 Add media pause function & buzzer wake-up alert
- 4 Integrate detection, web UI & smart room control
- 5 Add email reporting for sleep logs and drowsiness events

Thank you