**Sleep Event Detector**

1. **Overview**.

The sleep event detector is built using YAMNet backbone and a custom sneeze/sniff classifier on top of it. It is designed to detect:

| **Index** | **Event** | **Detector** |
| --- | --- | --- |
| 0 | snore | YAMNet subset |
| 1 | cough | YAMNet subset |
| 2 | fart | YAMNet subset |
| 3 | speech | YAMNet subset (sleep talking) |
| 4 | laughter | YAMNet subset |
| 5 | music | YAMNet subset |
| 6 | silence | YAMNet subset |
| 7 | sneeze | Sneeze/sniff classifier |
| 8 | sniff | Sneeze/sniff classifier |
| 9 | neither | Sneeze/sniff classifier |

1. **Model Details**

* File: sleep\_detector.tflite
* Framework: TensorFlow Lite
* Input: 1D waveform array
* Output: Array of 10 floats
* Expected input:
  + Length: 80,000 samples (5 seconds at 16 kHz)
  + Data type: Float32
  + Shape: [80000]
* Output format: [yamnet\_scores(7), sneeze\_sniff\_probs(3)]

1. **How it works**

* Recording audio: User presses Start Recording; audio is recorded at 16kHz.
* Preprocessing: Audio converted to float32 waveform; sliced into 5-second windows.
* TFLite inference: Each window passed into model; returns probabilities for 10 events.

1. **Recommended thresholds**:

{ snore: 0.3, cough: 0.3, fart: 0.3, speech: 0.5, laughter: 0.1, music: 0.3, silence: 0.5, sneeze: 0.3, sniff: 0.3 }

1. **Overnight Audio Processing (React Native)**

* Start recording on “Start Sleep”, keep audio in memory or temporary storage.
* Slice the audio into 5-second windows (or another configurable length).
* Run the TFLite model on each window.
* If a window exceeds thresholds for any event, save it as a separate WAV clip:
  + Filename: <event>\_<timestamp>.wav
  + Timestamp: relative to sleep start (or absolute datetime).
* After “Stop Sleep”, show a summary screen with all detected clips and allow playback.
* References