Audio Splitter - Technical Documentation

Audio Splitter - Technical Documentation

Overview

This document provides a detailed explanation of the Audio Splitter Python tool. The script captures live microphone input, detects keystrokes based on amplitude spikes, logs actual keypress events, and splits the audio into `.wav` segments accordingly.

1. Class: AudioSplitter

Constructor (__init__)

AudioSplitter(silence_threshold=500, min_keystroke_gap=0.1, output_dir="split_audio", log_file="keystroke_log.json")

Parameters:

- silence_threshold: Minimum amplitude to detect a keystroke.
- min_keystroke_gap: Time in seconds to ignore rapid sequential detections.
- output_dir: Directory where split audio files are stored.
- log_file: JSON file to log actual keyboard keypresses.

2. Method Details

callback()

Captures and buffers audio data from the microphone.

Monitors amplitude; if above the threshold and enough time has passed, logs it as a keystroke.

start_recording()

Opens PyAudio stream and begins recording.

Uses non-blocking callback for live processing.

stop_recording()

Audio Splitter - Technical Documentation

Stops and closes the audio stream.

Terminates PyAudio instance.

save_audio_segment()

Saves a portion of recorded frames to a `.wav` file.

split_by_keystrokes()

Uses keystroke_times to segment the audio.

Saves each segment to a file in output_dir.

log_actual_keystroke()

Captures keyboard events using keyboard.hook().

Appends event info (key, timestamp, frame index) to actual_keystrokes.

save_keystroke_log()

Writes logged keypress events to the JSON file defined in log_file.

monitor_keys()

Continuously hooks to keyboard events in a separate thread until ESC is pressed.

run()

Main entry point: starts audio + keyboard recording.

Waits for ESC key, then stops recording.

Splits audio and saves keystroke log.

3. Script Execution

Command-Line Interface

python main.py --silence-threshold 600 --keystroke-gap 0.2 --output-dir output --log-file log.json

Arguments:

Audio Splitter - Technical Documentation

--silence-threshold (int): Amplitude threshold for keystroke detection.

--keystroke-gap (float): Minimum gap between amplitude peaks.

--output-dir (str): Directory for saving split audio files.

--log-file (str): File to save keyboard keypress logs.

4. Output Files

- Audio Segments: .wav files (e.g., segment_1.wav, segment_2.wav, ...) saved in output_dir

- Keystroke Log: JSON log of actual keyboard presses with:

  - Key pressed

  - ISO timestamp

  - Frame index of when it occurred

5. Dependencies

pip install pyaudio keyboard numpy

- pyaudio: For recording microphone input

- keyboard: For key event listening

- numpy: For audio processing (amplitude calculations)

6. Notes

- The SILENCE_THRESHOLD and MIN_KEYSTROKE_GAP are critical for tuning accuracy.

- keyboard library may need elevated privileges on some systems.

- Ensure microphone is available and not blocked by other software.

7. License