Overview

This project is designed to process an audio file containing multiple people speaking, identify different speakers, and generate a text transcript labeled by speaker. The system processes the audio file step by step, applying various techniques to achieve accurate speaker diarization (identifying "who spoke when") and transcription.

Project Workflow

The project follows a structured approach, where each step processes the audio to achieve diarization and transcription:

1. **Audio Conversion:**
   - The raw audio file is processed using `convert_audio.py`.
   - It converts the file to a required format (`processed.wav`), ensuring it has the correct audio properties.
2. **Speaker Identification (Diarization):**
   - The `diarization_pyannote.py` script analyzes the audio and detects when each speaker talks.
   - The results are saved in a `diarization_results.json` file, marking timestamps for each speaker.
3. **Audio Visualization (Optional):**
   - The `visualize_audio.py` script generates a waveform of the audio to visualize sound patterns.
4. **Full Audio Transcription:**
   - The `whisper_asr.py` script uses an AI-based transcription model to convert the entire audio into text.
   - This results in a file `whisper_transcript.txt` that contains the full conversation.
5. **Speaker-Wise Transcription:**
   - The `speaker_wise_transcription.py` script combines the speaker identification data with the transcription.
   - It segments the audio into different dialogues and labels them with the correct speaker.
   - The final labeled transcript is saved as `final_transcript.txt`.

Technologies Used

- **Python** for scripting and automation.
- **Pydub** for audio processing.

- **Pyannote Audio** for speaker diarization (identifying who speaks when).
- **Whisper (OpenAI)** for transcribing speech to text.
- **Librosa & Matplotlib** for visualizing the audio waveform.
- **JSON** for storing diarization results.
- **FFmpeg** for handling audio file formats.

Output and Results

- `processed.wav` - The standardized version of the input audio.
- `diarization_results.json` - A file showing speaker time segments.
- `whisper_transcript.txt` - Full transcription without speaker labels.
- `final_transcript.txt` - Speaker-wise labeled transcription for easy understanding.
- A waveform visualization for analysis (optional).

Use Cases

- Transcribing and labeling speakers in **meetings and interviews**.
- Generating subtitles for **podcasts, panel discussions, and talk shows**.
- Creating **AI-powered meeting notes**.
- Enhancing **accessibility** for recorded conversations.