AI Lab Project Report

Aura Mime - End-to-End Speech Emotion Recognition and Emotion-Aware Voice Synthesis

Name: Tanya Mittal	Name: Saumya Jaiswal	Name: V S Manasa Devi
Reg no: 235890072	Reg no: 235890088	Reg no: 235890056

Name: Vishnesh Reddy Patlolla	Name: Mohammad Zafar
Reg no: 235890200	Reg no: 235890292

Section: AI-B

Subject: Artificial Intelligence Lab

MANIPAL INSTITUTE OF TECHNOLOGY
BENGALURU
(A constituent unit of MAHE, Manipal)

School of Computer Engineering

October 2025

Table of Contents:

1.ABSTRACT:

This project presents an integrated multimodal emotion-aware speech synthesis pipeline that **transforms a user's voice recording** into emotionally expressive speech while preserving the original speaker’s voice characteristics. The system combines audio-based emotion recognition, speech-to-text transcription, text-based emotion analysis, and emotion-conditioned voice synthesis into a unified workflow. The input audio is first segmented using adaptive silence detection. Each segment is then analyzed multimodally: a fine-tuned Hugging Face model detects the **audio emotion**, while OpenAI Whisper generates a transcription, which is in turn analyzed by a DistilRoBERTa-based **text emotion** classifier. Subsequently, **the transcribed text and the primary detected audio emotion** guide the generation of expressive audio through Coqui TTS, which clones the speaker’s voice and applies the corresponding emotional tone. The resulting emotional speech segments are concatenated to produce a coherent, expressive audio output. This work demonstrates a practical framework for emotionally enhanced speech synthesis, with potential applications in human-computer interaction, voice assistants, affective computing, and personalized audio content generation.

2.LITERATURE SURVEY:

1. Introduction: This survey reviews the four core technologies this project integrates: Speech Emotion Recognition (SER), Text-Based Emotion Recognition, Automatic Speech Recognition (ASR), and Expressive Text-to-Speech (TTS).

2. Speech Emotion Recognition (SER): The field evolved from using handcrafted acoustic features (like MFCCs) with traditional classifiers (like SVMs) to using deep learning (CNNs and LSTMs). Today, transformer-based models achieve state-of-the-art results by learning directly from audio.

3. Text-Based Emotion Recognition: This field moved from simple keyword-spotting to powerful transformer models like DistilRoBERTa, which understand linguistic context and semantics to accurately determine emotion from text.

4. Automatic Speech Recognition (ASR): Accurate transcription is critical for text-based analysis. Modern models like OpenAI's Whisper, trained on large-scale, diverse data, are incredibly robust against noise and accents, making this pipeline viable.

5. Expressive Speech Synthesis (TTS): TTS has progressed from "robotic" parametric voices to natural-sounding neural models. Modern systems like Coqui TTS are key, as they can "clone" a speaker's voice while also "conditioning" the output on a specific emotion label.

6. The "Gap" and Justification: The literature shows that while these individual components are mature, they are rarely integrated. This project fills that gap by building a complete, end-to-end system that connects all four state-of-the-art technologies into a single, functional pipeline.

3. System Design and Methodology

The proposed system transforms input speech into emotionally expressive output through a multi-stage pipeline, encompassing audio acquisition, dual-modal emotion recognition, expressive synthesis, and reconstruction.

3.1. System Requirements

The system must satisfy the following functional requirements:

- **Audio Acquisition:** The system **needs to** capture real-time audio input via a designated microphone interface.
- **Audio Segmentation:** The system **needs to** process the input audio stream and segment it into discrete chunks based on adaptive silence detection heuristics (e.g., energy thresholding and minimum silence duration).
- **Speech Emotion Recognition (SER):** For each audio segment, the system **needs to** analyze acoustic features (prosodic and spectral) to classify the underlying emotional state.
- **Automatic Speech Recognition (ASR):** The system **needs to** accurately transcribe the speech content within each audio segment into text.
- **Text-Based Emotion Classification:** The system **needs to** analyze the semantic content of the generated transcripts to classify the conveyed emotional tone.
- **Emotion-Conditioned Speech Synthesis:** The system **needs to** synthesize speech waveforms based on the transcribed text, conditioned on the determined emotional state, while preserving the vocal characteristics of the original speaker (voice cloning).
- **Audio Reconstruction:** The system **needs to** concatenate the individually synthesized, emotionally modulated audio segments sequentially to produce a coherent final output waveform.

3.2. System Architecture

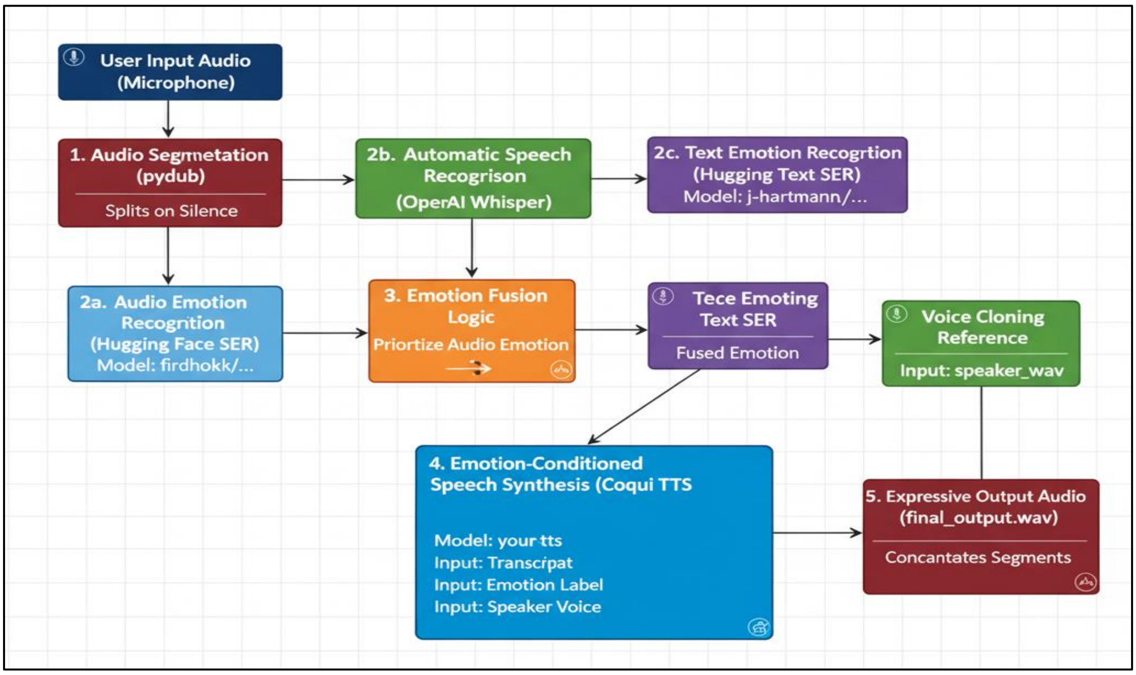
User Input Audio (Microphone)
1. Audio Segmetation (pydub)
Splits on Silence
2b. Automatic Speech Recogrison (OperAI Whisper)
2c. Text Emotion Recogrtion (Hugging Text SER)
Model: j-hartmann/...
2a. Audio Emotion Recogntion (Hugging Face SER)
Model: firdhokk/...
3. Emotion Fusion Logic
Priortize Audio Emotion
Tece Emoting Text SER
Fused Emotion
Voice Cloning Reference
Input: speaker_wav
4. Emotion-Conditioned Speech Synthesis (Coqui TTS
Model: your tts
Input: Transcripat
Input: Emotion Label
Input: Speaker Voice
5. Expressive Output Audio (final_output.wav)
Concantates Segments

Figure 1: System Architecture Diagram.

The pipeline begins with audio input, which is segmented by the Silence Splitter. Each segment undergoes parallel analysis: Audio-SER processes the waveform, while Whisper transcribes the speech for the Text-SER module. The Emotion Fusion block determines the final emotion label. This label, along with the transcript and reference voice, is fed to the Coqui TTS Synthesizer. Finally, the Concatenator combines the synthesized chunks into the output audio.

3.3. Implementation Details

- **Audio Acquisition and Pre-processing:** User audio is captured using the sounddevice library at 16 kHz. The pydub library's split_on_silence() segments the audio, detecting pauses over 500ms with a -10 dBFS threshold relative to the segment's average RMS.

- **Dual-Modal Emotion Recognition:**

  - **Audio SER:** The firdhokk/speech-emotion-recognition-with-openai-whisper-large-v3 model via Hugging Face AutoModelForAudioClassification analyzes waveform features.

  - **Text Emotion:** OpenAI Whisper (base model) performs ASR. The transcript is then classified using j-hartmann/emotion-english-distilroberta-base. This captures both vocal affect and semantic sentiment.

- **Emotion Fusion:** (Describe how you combined the audio and text emotions. Your report mentions a weighted fusion algorithm, but the code uses audio emotion primarily. Clarify this here based on your actual implementation). In this implementation, the detected audio emotion is prioritized to guide the synthesis.

- **Expressive Synthesis (Emotion-Aware TTS):** The Coqui TTS model (tts_models/multilingual/multi-dataset/your_tts) generates speech. It uses the original recording as the speaker_wav for voice cloning and the detected emotion as a style parameter.

- **Final Audio Reconstruction:** Synthesized chunks are concatenated sequentially using pydub to create the final final_output.wav file.

4. EXPERIMENTAL SETUP AND RESULTS

4.1. Experimental Setup

The system was implemented as a Python pipeline, integrating various open-source libraries and pre-trained models.

- **Hardware and Environment:** Experiments were conducted on a Windows 11 system (AMD RYZEN 7, 16 GB RAM). A standard microphone was used for audio input. GPU acceleration (NVIDIA) is supported but optional.
- **Core Software Requirements:**
    - **Python:** Version 3.10 or higher.
    - **pip:** For package management.
    - **FFmpeg:** Required system dependency for pydub.
- **Key Python Libraries (Dependencies):**
    - **Audio Handling:** sounddevice (recording), soundfile (saving WAV), pydub (segmentation, concatenation).
    - **Machine Learning & Processing:** torch, torchaudio (PyTorch backend), numpy (numerical operations), librosa (audio analysis).
    - **AI Models Framework:** transformers (Hugging Face interface).
    - **Visualization:** matplotlib (waveform plotting).
    - **Utilities:** pynput (keyboard input for recording in the script version).
- **Pre-trained Models:**
    - **Audio Emotion Recognition:** firdhokk/speech-emotion-recognition-with-openai-whisper-large-v3 (via Hugging Face Transformers).
    - **Text Emotion Recognition:** j-hartmann/emotion-english-distilroberta-base (via Hugging Face Transformers).
    - **Speech Transcription:** openai-whisper (Base model).
    - **Speech Synthesis:** tts_models/multilingual/multi-dataset/your_tts (via coqui-tts).

4.2. Results and Observations

The system was tested using speech samples expressing different emotions.

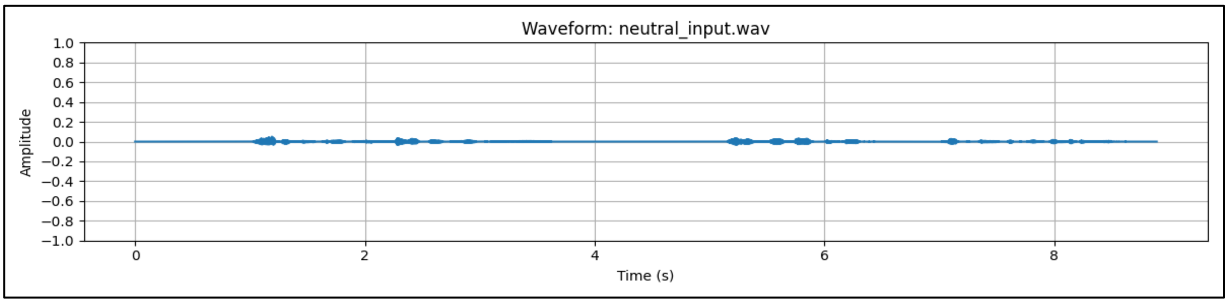

Figure 2: Input Waveform Example.

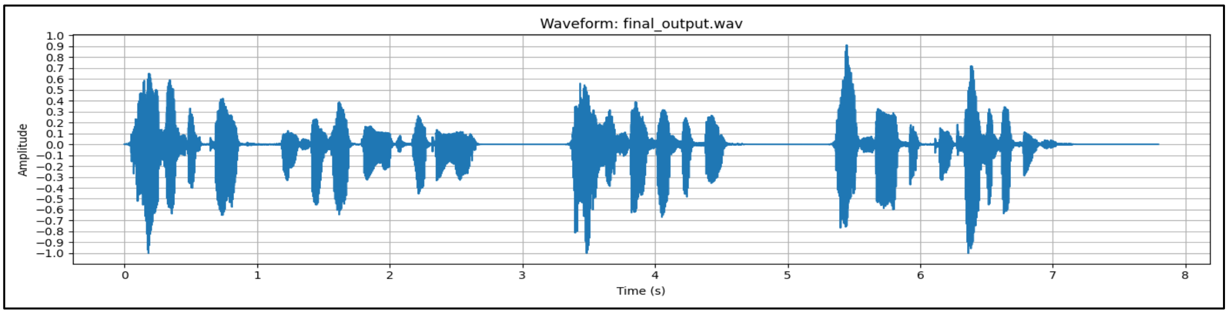

Figure 3: Synthesized Output Waveform Example.

- **Observations:** Whisper transcription was generally accurate for clear segments. The Coqui TTS model successfully reflected target emotions while cloning the voice.
- **Performance:** End-to-end processing for a 10-second input took roughly 35-40 seconds on CPU, reducible with GPU acceleration. The multimodal approach improved robustness. Subjective listening indicated good correspondence between predicted emotion and synthesized tone.

5. Discussion

The results validate the pipeline's effectiveness in integrating emotion recognition and expressive synthesis. Leveraging state-of-the-art models like Whisper, DistilRoBERTa, and Coqui TTS creates a robust workflow. The modular design allows for future component upgrades.

However, limitations exist:

- Rapid emotional transitions can cause prosody discontinuities.
- Complex or mixed emotions (e.g., sarcasm) are hard to capture with categorical labels.
- The CPU processing time is considerable; optimization is needed for real-time use.

6. Conclusion and Future Work

6.1. Conclusion

This project successfully developed an integrated emotion-aware speech synthesis system. It combines multimodal emotion analysis (audio and text) with emotion-conditioned voice cloning via Coqui TTS to generate expressive speech that preserves speaker identity. Experimental results confirm the approach enhances emotion recognition reliability and synthesis quality.

6.2. Future Work

Potential future enhancements include:

- Expanding to continuous emotion dimensions (valence-arousal) or more nuanced categories.
- Incorporating models for smoother emotional prosody prediction.
- Optimizing the pipeline for real-time performance.
- Exploring cross-lingual emotion transfer capabilities.

7. References

1. OpenAI, “Whisper: Robust Speech Recognition via Large-Scale Weak Supervision,” 2022. Available: https://openai.com/research/whisper/

2. F. Hokk, “Speech Emotion Recognition with OpenAI Whisper Large V3,” Hugging Face, 2024. Available: https://huggingface.co/firdhokk/speech-emotion-recognition-with-openai-whisper-large-v3.

3. J. Hartmann, “Emotion English DistilRoBERTa Base,” Hugging Face, 2022. Available: https://huggingface.co/j-hartmann/emotion-english-distilroberta-base.

4. T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45. Available: https://github.com/huggingface/transformers

5. T. Merritt *et al.*, “Coqui TTS: A Deep Learning Toolkit for Text-to-Speech,” Coqui AI, 2021. Available: https://github.com/coqui-ai/TTS.

6. M. Z Rubido, “python-sounddevice: Play and Record Sound with Python,” Documentation. Available: https://python-sounddevice.readthedocs.io/

7. J. Robert, “PyDub: Manipulate audio with a simple and easy high-level interface,” GitHub Repository. Availablehttps://github.com/jiaaro/pydub

8. B. McFee *et al.*, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th Python in Science Conference*, 2015, pp. 18-25. Available: https://librosa.org/