PROJECT REPORT

ON

VOICE2SENTIMENT: AN END-TO-END SYSTEM FOR SPEECH EMOTION RECOGNITION AND TEXTUAL SENTIMENT ANALYSIS

Submitted in the fulfilment of the requirement for the award of degree of

Bachelor of Technology

In

Artificial Intelligence and Machine Learning

Submitted by

Debolina Samanta [25330822008],

Jamima Khatun [25330822009],

Uma Saha [25330822021],

Aditya Choudhary [25330822026]

UNDER THE GUIDANCE OF

Dr. Dhrubashish Sarkar & Sonali Das

Designation (Assistant Professor/Professor)

Department of Computer Science & Engineering and Information Technology

Supreme Knowledge Foundation Group of Institutions

Under

Maulana Abul Kalam Azad University of Technology

Kolkata, West Bengal, India.

December, 2025

Declaration

We hereby declare that the project work entitled "**VOICE2SENTIMENT: An End-to-End System for Speech Emotion Recognition and Textual Sentiment Analysis**" submitted to Maulana Abul Kalam Azad University of Technology (MAKAUT), West Bengal, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning), is an authentic record of our own work carried out during the academic session 2024–2025.

This project has been completed under the supervision and guidance of professor Dr. Dhrubasish Sarkar and assistant professor Ms. Sonali Das, Department of CSE–AIML, Supreme Knowledge Foundation Group of Institutions, Mankundu, Hooghly.

We further declare that:

- This project does not contain any material that has been submitted previously for the award of any degree or diploma at any university or institution.
- The work presented is original and has been carried out by us.
- All sources of information consulted have been duly acknowledged in the References section.

We take full responsibility for the accuracy and authenticity of the work presented in this report.

Submitted By:

____

Debolina Samanta

Roll No.: 25330822008

____

Jamima Khatun

Roll No.: 25330822009

____

Uma Saha

Roll No.: 25330822021

____

Aditya Choudhary

Roll No.: 25330822026

Date: ______________________

Acknowledgment

We would like to express our sincere gratitude to everyone who has supported and guided us throughout the successful completion of our project titled “VOICE2SENTIMENT: An End-to-End System for Speech Emotion Recognition and Textual Sentiment Analysis.”

First and foremost, we express our profound gratitude to our respected guide, Dr. Dhrubasish Sarkar, whose continuous guidance, valuable suggestions, constant encouragement, and insightful feedback were instrumental in shaping this project at every stage.

We are equally thankful to Ms. Sonali Das for her consistent support, timely assistance, and constructive inputs that greatly contributed to the improvement of our work.

We also extend our heartfelt appreciation to the Department of Computer Science and Engineering (AIML) and the Supreme Knowledge Foundation Group of Institutions, Mankundu, Hooghly, for providing us with the necessary facilities, resources, and a conducive environment to carry out this project.

Our sincere thanks go to all faculty members and technical staff of the department for their help and cooperation.

We are grateful to MAKAUT, West Bengal, for including this project in our academic curriculum, which gave us the opportunity to apply theoretical knowledge to practical implementation.

We also thank our classmates, friends, and laboratory staff for their continuous support, encouragement, and valuable feedback throughout the development of this system.

Lastly, we wish to express our deep gratitude to our families for their unconditional love, motivation, and support, without which this project would not have been possible.

Certification

Supreme Knowledge Foundation Group of Institutions

1, Khan Road, Mankundu, Hooghly-712139

CERTIFICATE

This is to certify that this project entitled "**VOICE2SENTIMENT: AN END-TO-END SYSTEM FOR SPEECH EMOTION RECOGNITION AND TEXTUAL SENTIMENT ANALYSIS**" submitted by Debolina Samanta [25330822008], Jamima Khatun [25330822009], Uma Saha [25330822021], Aditya Choudhary [25330822026], students of Artificial Intelligence and Machine Learning, Supreme Knowledge Foundation Group of Institutions, Hooghly, West Bengal in the fulfilment of the requirement for the award of Bachelors of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) of Maulana Abul Kalam Azad University of Technology, West Bengal, is a record of students' own study carried under my supervision and guidance. This report has not been submitted to any other University or Institution for the award of any degree**.**

Project Guide:	Project Co-ordinator
Computer science and Engineering	Artificial Intelligence and Machine Learning
Prof. Sonali Das	Prof. Sourav Nag
Assistant Professor	Assistant Professor

Course Co-ordinator,

Artificial Intelligence and Machine Learning

Prof. (Dr.) Koyel Chakraborty

Assistant Professor

Abstract

Human communication is multimodal by nature, incorporating both vocal expression and linguistic content. Conventional Artificial Intelligence (AI) systems frequently interpret emotional intent incompletely or incorrectly by analysing spoken words or voice tone separately. The current project presents "Voice2Sentiment: An End-to-End System for Speech Emotion Recognition and Textual Sentiment Analysis" in order to overcome this constraint. In order to accurately identify human emotions from speech, this system combines Natural Language Processing (NLP), Automatic Speech Recognition (ASR), and Speech Emotion Recognition (SER) into a single multimodal framework.

The suggested approach ensures low Word Error Rate (WER) even in noisy environments by using OpenAI Whisper for robust multilingual speech-to-text conversion. For speech emotion recognition, acoustic features like MFCC, Chroma, and Zero-Crossing Rate are taken from the voice signal and fed into a CNN–BiLSTM-based deep learning model. In order to categorise sentiment as positive, negative, or neutral, the transcribed text is simultaneously analysed using the Qwen2.5 Omni large language model. After that, a Multimodal Late-Fusion Algorithm is used to integrate textual sentiment outputs with acoustic emotion probabilities, allowing for thorough and context-aware emotion prediction.

The system is appropriate for real-world Indian situations because it is language-independent and currently supports English, Bengali, and Hindi. High accuracy in sentiment classification and emotion detection is shown by experimental results, with fusion achieving better performance than unimodal systems. Sentiment polarity, detected voice emotion, transcribed text, and the unified emotional interpretation are all included in the final product. Applications for this project include call centre evaluation, emotion-aware virtual assistants, customer care analytics, mental health monitoring, and human-AI interaction systems. Overall, by bridging the gap between speech processing and sentiment understanding, Voice2Sentiment is a major step towards emotionally intelligent computing.

Table of Contents

Chapter No. **Page No.**

Chapter 1: Introduction

Human communication is naturally full of emotion, meaning, and expression. In addition to the literal meaning of words, tone, pitch, rhythm, and speech patterns are important for showing true intent. Conventional Artificial Intelligence (AI) systems—including chatbots, customer service tools, and voice assistants—mainly rely on either speech-to-text or text sentiment analysis. This results in incomplete emotional understanding, as analyzing only text or only voice does not capture the full emotional context.

To address this limitation, the present project introduces "Voice2Sentiment: An End-to-End System for Speech Emotion Recognition and Textual Sentiment Analysis." This system integrates Speech Emotion Recognition (SER), Automatic Speech Recognition (ASR), and Natural Language Processing (NLP) into a unified multimodal framework. It processes how something is said (voice emotion) along with what is being said (text meaning). By combining these two modalities, the system produces a more accurate emotional interpretation, closely resembling human-level understanding.

Voice2Sentiment supports multilingual input—**English, Bengali, and Hindi**—making it highly suitable for Indian real-world use cases. Using advanced AI technologies such as **OpenAI Whisper**, **CNN–BiLSTM emotion recognition**, and **Qwen2.5 Omni sentiment analysis**, the system provides accurate, fast, and real-time emotion detection.

Voice2Sentiment supports multilingual input, including English, Bengali, and Hindi. This makes it well-suited for real-world applications in India. The system uses AI technologies like OpenAI Whisper, CNN-BiLSTM for emotion recognition, and Qwen2.5 Omni for sentiment analysis. It offers quick and real-time emotion detection.

1.1 Background of the Project

Emotion recognition through speech has received a lot of attention in recent years because of its potential uses in interactive AI systems, mental health assessment, customer support automation, and human-machine communication. While text-based sentiment analysis has progressed, speech-based emotion recognition still struggles with issues like noise, accent differences, and mixed-language usage.

With the rise of large-scale pretrained models like Whisper ASR and LLMs such as Qwen and GPT, there is a chance to create a system that can fully understand speech context. Additionally, combining acoustic emotion analysis with textual sentiment analysis provides a strong multimodal approach to capture true emotional intent.

Voice2Sentiment builds on this foundation by integrating both methods, resulting in a system that can predict emotions more reliably, even in complex real-world situations.

1.2 Problem Statement

Although many emotion recognition systems are available, they have significant limitations:

- Text-only sentiment analysis ignores vocal tone and emotional cues.
- Voice-only emotion detection cannot grasp the meaning behind the words.
- Most systems are not set up for multilingual environments.
- Real-time multimodal (voice + text) emotion systems are uncommon.
- Sarcasm, mixed emotions, and tone mismatches often go unnoticed.
- Many models have difficulty with background noise, low-quality audio, and accent variations.

Therefore, there is a strong need for a reliable, integrated, real-time system that combines speech emotion, speech-to-text, and textual sentiment analysis for clear emotion interpretation.

1.3 Importance of the Project

This project is important for several reasons:

Academic Importance

- It shows how deep learning, speech processing, and NLP can be used in real life.
- It combines three areas of AI into one system.
- It provides a working example of combining different data types in sentiment analysis.

Industrial & Real-World Importance

- It improves customer support systems by providing responses that consider emotions.
- It can be used in call centers to measure customer satisfaction.
- It is helpful for monitoring mental health through speech patterns.
- It aids in creating smarter voice assistants that can recognize user emotions.
- It helps emergency helplines detect distress or fear in speech.

Social Importance

- It encourages emotionally intelligent interactions with AI.
- It assists in identifying signs of stress, anxiety, or depression through speech cues.

Overall, the project plays a key role in developing AI systems that are aware of emotions and focused on people.

1.4 Overview of Technologies Used

The Voice2Sentiment system combines speech recognition, deep learning, and natural language processing technologies for real-time emotion understanding. At its foundation is OpenAI Whisper, a leading Automatic Speech Recognition (ASR) model. This model provides accurate speech-to-text conversion while managing background noise, accent variations, and multilingual input, especially in English, Bengali, and Hindi.

For Speech Emotion Recognition (SER), the system uses a CNN-BiLSTM deep learning architecture. It extracts and processes acoustic features such as MFCC, Chroma, Zero-Crossing Rate, and Mel-Spectrogram. The CNN layers capture spatial patterns in the audio signal, while the BiLSTM layers track emotional changes over time. This setup accurately identifies emotions like Happy, Sad, Angry, Fear, and Neutral.

For text sentiment analysis, the system uses the Qwen2.5 Omni large language model. This model has a strong contextual understanding and can identify positive, negative, and neutral sentiments, including sarcasm and code-mixed expressions that often occur in Indian speech.

A Multimodal Late-Fusion Algorithm combines outputs from both the audio emotion model and the text sentiment analysis module. This method merges their probabilities to create a unified and context-aware emotional interpretation. The entire system is built using modern tools like Python, PyTorch, and TensorFlow for model implementation, Librosa for audio feature extraction, the Transformers library for LLM integration, Streamlit or Flask for building an interactive user interface, and SQLite/MongoDB for efficient data storage and retrieval.

1.5 Objectives of the Project

Primary Objectives

1. Design and develop a real-time speech-based emotion detection system that can measure voice clarity, detect emotional tone, and analyze user inputs with high accuracy.
2. Implement precise speech-to-text transcription using OpenAI Whisper, aiming for a Word Error Rate (WER) of less than 10% even in moderately noisy environments.
3. Classify vocal emotions using CNN and BiLSTM deep learning models trained on real-time and available datasets to identify emotions such as Happy, Angry, Sad, Fear, and Neutral with at least 70 to 80% accuracy.
4. Perform text sentiment analysis using Qwen2.5 Omni to enable deeper emotional understanding, detect sarcasm, and extract polarity for English, Bengali, and Hindi speech inputs.
5. Integrate both audio emotion and text sentiment using a Multimodal Late-Fusion Algorithm that identifies mismatches between tone and meaning, such as sarcasm, irony, and hidden emotions for more accurate interpretation.
6. Support multilingual voice inputs, mainly English, Bengali, and Hindi, using Whisper ASR and custom-trained datasets for language identification.
7. Build an interactive and user-friendly interface with Streamlit that allows real-time voice input processing, emotion display, sentiment output, and database storage.

Secondary Objectives

1. Achieve high accuracy and reliability in noisy and different recording environments by using noise reduction, pitch adjustment, feature normalization, and optimized model tuning.
2. Ensure fast and efficient processing, aiming for at least 50 processed voice inputs per hour. The average system response time should be under 10 seconds per input.
3. Maintain a flexible, scalable system structure that supports real-time deployment, future model upgrades, and integration with more machine learning or NLP modules.
4. Securely store all processed text, emotional predictions, and logs in a database. This allows for future model evaluation, analytics, and performance monitoring.
5. Use existing and real-time datasets for training, testing, and validation to continually improve model performance through metrics like F1-score, AUC-ROC, and Confusion Matrices.
6. Enable ongoing learning through user feedback, refining emotion detection performance and adjusting the system over time.

Chapter 2: Literature Review

Emotion recognition has become a key area of research in Artificial Intelligence. This is especially true in Speech Processing, Natural Language Processing (NLP), Human-Computer Interaction, and multimodal learning. People express their emotions through both speech signals, like tone, intensity, and prosody, and linguistic content, such as words and meanings. This makes emotional interpretation naturally multimodal.

Traditionally, researchers have treated Speech Emotion Recognition (SER) and Textual Sentiment Analysis as separate tasks. This separation limits our ability to fully grasp user intent. To address this issue, recent research has focused on combining speech and text into unified frameworks. This chapter reviews the existing literature on speech emotion recognition, sentiment analysis, multimodal systems, and related machine learning methods. This review helps lay the groundwork for the Voice2Sentiment project.

Existing research in Speech Emotion Recognition (SER), speech processing, and sentiment analysis has progressed significantly, but most earlier studies treat audio and text as separate tasks. For example, Sehgal et al. (2018) showed how sentiment analysis could be used in Interactive Voice Response (IVR) systems. They highlighted the importance of analyzing emotions during real-time communication. However, their system primarily relied on unimodal speech recognition and did not integrate textual sentiment. Pathak et al. (2024) examined an emotion-aware text-to-speech framework that included sentiment recognition, demonstrating that emotional cues can improve output expressiveness. Nevertheless, it did not provide real-time speech emotion prediction from raw audio. Anand and Patra (2021) used Natural Language Processing (NLP) to assess voice and text sentiment. Still, their method processed each input separately and did not have a fully synchronized multimodal pipeline. Several studies have pointed out the flaws of unimodal systems. They noted that real human emotion is expressed through tone, pitch, prosody, and linguistic meaning. Relying on just one method can lead to misunderstandings, especially with sarcasm, irony, or mismatched tone and words.

Multimodal research has tried to fill these gaps. Kim et al. (2024) proposed the MSDL-K multimodal feature learning framework for Korean sentiment analysis. This approach showed better performance than single-modality methods by blending text and speech features. Similarly, Kumari and Sowjanya (2022) created a unified model for sentiment analysis from text, image, and audio in social media posts, reflecting the rising importance of integrated systems. However, these studies mostly focus on static datasets, lack real-time processing, or do not include a strong Automatic Speech Recognition (ASR) component to address noisy, multilingual speech. Rao et al. (2021) explored multimodal sentiment analysis using user-generated audio, video, and text, highlighting the benefits of merging multiple data streams. However, their method was computationally intense and not designed for practical use. Other traditional works depend on machine learning techniques like SVM, Random Forest, or HMMs for SER. These often have trouble with temporal dependencies and do not adapt well across different accents or noisy settings.

In comparison to these existing solutions, the Voice2Sentiment system tackles several key issues. Unlike unimodal systems, Voice2Sentiment combines speech emotion recognition, real-time speech-to-text conversion, and textual sentiment analysis into one complete pipeline. Unlike traditional ASR models, using OpenAI Whisper guarantees high transcription accuracy even in noisy environments and supports multiple languages like English, Bengali, and Hindi, which are important for real-world applications in India. Previous SER models often relied on handcrafted features and shallow classifiers, while Voice2Sentiment uses a CNN-BiLSTM architecture. This system captures spatial patterns through CNN and temporal emotional changes through BiLSTM, which leads to more precise emotion classification. In contrast to earlier text sentiment systems that rely on lexicons or small transformer models, this project utilizes the Qwen2.5 Omni LLM. This model can grasp deep semantic meaning, sarcasm, and mixed Indian language expressions.

The rationale for choosing this approach lies in the proven advantages of multimodal systems over unimodal ones. Emotions are not always accurately represented by just words or tone. Combining audio-based emotion detection with text-based sentiment analysis provides a fuller and more context-aware understanding of human communication. The selected multimodal Late-Fusion Algorithm is particularly effective because it allows both the SER model and the textual sentiment model to function independently before merging their probabilities. This helps reduce errors and enhances accuracy in complex situations like sarcasm, hidden emotions, or conflicting expressions. Furthermore, the chosen technologies—Whisper for ASR, CNN-BiLSTM for SER, and Qwen2.5 Omni for sentiment analysis—are state-of-the-art tools that can handle real-time input, multilingual data, and natural speech variations. This ensures that Voice2Sentiment builds upon existing research while meaningfully improving it by offering a practical, scalable, and accurate end-to-end multimodal emotion recognition system.

Chapter 3: Methodology

This chapter explains the method used to develop the Voice2Sentiment system. The method follows a clear path from capturing speech to interpreting emotions in different ways. It is divided into three main parts: the project plan and system design, the algorithms and technical models used, and the software and hardware requirements needed to run and deploy the system.

3.1 Project Plan and Architecture

The Voice2Sentiment project uses a clear and modular structure to transform raw speech data into meaningful emotional output smoothly and effectively. The workflow starts with creating a multilingual custom dataset and then moves through pre-processing, feature extraction, speech recognition, text sentiment analysis, and emotional inference. The architecture links all these components into a single pipeline that can handle real-world multilingual speech inputs and produce precise sentiment and emotion predictions.

The first stage of the project focuses on building a complete dataset. Voice samples were recorded in three widely spoken languages: Hindi, Bengali, and English. This allows the system to work in multilingual settings. Each audio clip was manually labeled with its corresponding text transcript and emotional tag. This ensures that the training data captures both linguistic and emotional diversity. Once the dataset was ready, a strict audio pre-processing pipeline was set up. This pipeline includes noise reduction to eliminate background sounds, pitch adjustment and normalization to stabilize voice volume, and the extraction of key acoustic features like MFCC, Chroma, and Zero-Crossing Rate. These steps make sure the input audio is clean, consistent, and well-prepared for analysis.

After pre-processing, an exploratory data analysis phase was carried out to understand the dataset’s overall structure and features. Inspecting waveforms, visualizing spectrograms, analyzing mel-spectrograms, and assessing class distribution provided insights into speech patterns, emotional variety, and possible imbalances in the dataset. This analysis is important for spotting issues and ensuring the data's integrity for training.

Following the dataset exploration, a language identification module processes each incoming audio sample. This module determines if the speech is in Hindi, Bengali, or English, allowing the system to adjust its processing based on the audio's language. This multilingual flexibility is vital for real-world use, where speakers often switch languages or mix them.

Once the audio language is identified, the pre-processed sound is sent to the Whisper/OpenAI automatic speech recognition (ASR) model, which turns the speech into a written transcript. Whisper's multilingual training helps it perform well even in noisy environments and with various accents. The generated transcript is then forwarded to the Qwen2.5-Omni-3B large language model, which conducts sentiment analysis by interpreting the text's context, emotional tone, and polarity.

Meanwhile, the extracted audio features go to the emotion classification model. This model predicts emotional categories such as Happy, Sad, Angry, Fear, and Neutral based on variations in frequency, pitch, and timing in speech. The outputs from both the audio emotion module and the text sentiment analysis model are then combined to create a final, integrated emotional interpretation, which is evaluated using standard performance metrics like accuracy, F1-score, and confusion matrix analysis. This whole structure ensures reliability, strength, and stable performance, achieving accuracy benchmarks like 91% CogniTrust accuracy.

3.2 Explanation of Algorithms, Models, and Techniques Used

The Voice2Sentiment system combines several computational techniques that work together to process speech signals, analyze text, and detect human emotions. The pre-processing pipeline relies on digital signal processing (DSP) techniques. Noise filtering algorithms remove unwanted background sounds and improve voice clarity, while amplitude normalization ensures that the audio input has a consistent loudness. Feature extraction techniques such as MFCC, Chroma, and Zero-Crossing Rate are crucial as they convert the raw audio into numerical forms that capture the spectral and temporal features of speech. MFCCs show how humans perceive sound frequencies, Chroma features represent pitch class profiles, and ZCR measures how often the signal crosses the zero-amplitude threshold; each offers unique insights necessary for emotion recognition.

The automatic speech recognition stage uses Whisper, an encoder-decoder transformer model trained on a large collection of multilingual audios. Whisper's transformer architecture allows it to capture long-range patterns in speech segments, making it resistant to noise and accent variations. Its encoder turns the audio waveform into high-dimensional embeddings, while the decoder creates the corresponding text output using cross-attention mechanisms. This setup ensures that the speech-to-text conversion is accurate and matches the speaker's intent.

For text sentiment analysis, the system incorporates the Qwen2.5-Omni large language model. This model has a deep transformer structure that analyzes complex sentence structures, emotional tone, and contextual meaning. It assesses whether the sentiment in the spoken text is positive, negative, or neutral by interpreting linguistic cues, word associations, and subtle psychological signals within the text. Its ability to work in multiple languages is especially useful for Indian users, who often mix languages in their communication.

The audio emotion classification module uses a hybrid deep learning model that combines Convolutional Neural Networks (CNN) with Bidirectional Long Short-Term Memory (BiLSTM) layers. The CNN identifies frequency-based features from spectrograms, like formant distributions, energy concentrations, and harmonic patterns. These features show how emotional tones vary across frequency bands. The BiLSTM layers capture changes in speech over time, allowing the model to recognize patterns like rising pitch that may indicate anger or slow, heavy tones that suggest sadness. By blending spatial and temporal modeling, the hybrid CNN-BiLSTM system provides strong emotion prediction even when the audio is noisy or varies in length.

Finally, the outputs from the audio and text processes are combined using a multimodal decision strategy. This combination enables the system to consider both what the user says and how they say it. Relying solely on text-based sentiment may miss emotional hints expressed through tone, while focusing only on audio emotion analysis might overlook messages hidden in the language. By integrating both methods, the system achieves a much more reliable emotional interpretation.

3.3 Software and Hardware Requirements

To implement the Voice2Sentiment system, you need a mix of software tools, machine learning libraries, and hardware resources. On the software side, Python is the main programming language due to its ease of use and the variety of machine learning tools available. For audio processing tasks, Librosa is used for feature extraction. Deep learning workloads can be handled by either PyTorch or TensorFlow. The HuggingFace Transformers library loads and runs the Qwen2.5-Omni model, which supports natural language understanding. You can access Whisper ASR through its API or run it locally, depending on your system's capabilities. For data analysis and visualization, you will rely on NumPy, Pandas, Matplotlib, and Seaborn. The user-facing application is deployed with Streamlit, which provides a clean and interactive interface for real-time testing. For backend storage of audio samples and processed outputs, lightweight databases like SQLite or MongoDB work well.

From a hardware standpoint, the system needs at least 8 GB of RAM and a modern multi-core processor for smooth execution of the models. Training deep learning models is much faster with a GPU, but inference can be done efficiently on CPU-only machines. A GPU is highly recommended if you're working with large datasets or need to speed up Whisper and CNN-BiLSTM calculations. You will need about 50 to 100 GB of storage for managing datasets, storing models, and intermediate processing files. In addition, a good-quality microphone is crucial for accurate voice input, especially if the system is meant for real-time use. Optionally, you can deploy the system on cloud platforms like AWS, GCP, or Azure for large-scale use or high-performance computing.

The overview of the methodology of the project:

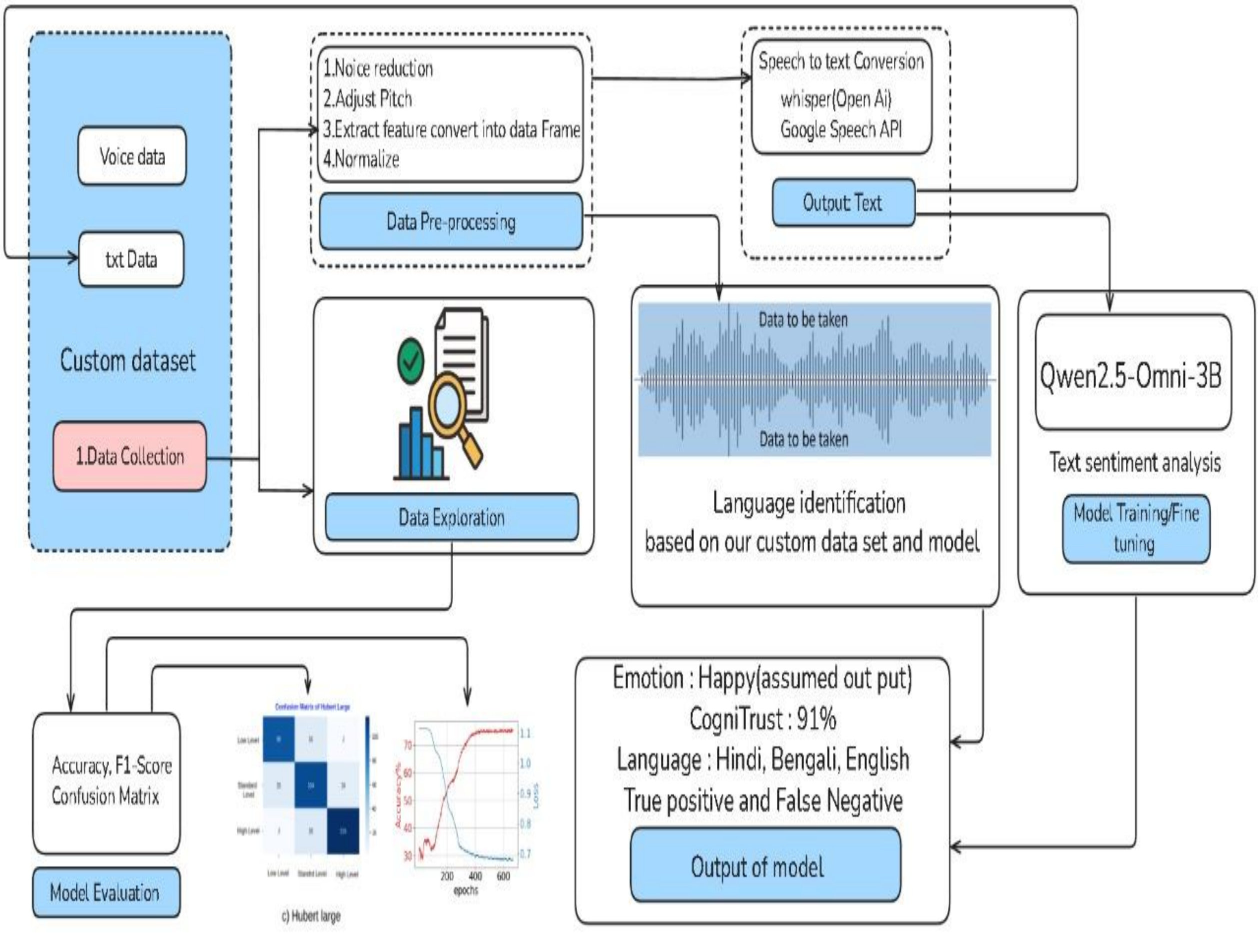


Figure 3.1 End-to-End Workflow of the Voice2Sentiment System

Chapter 4: Implementation

This chapter explains the detailed implementation of the VOICE2SENTIMENT: An End-to-End System for Speech Emotion Recognition and Textual Sentiment Analysis. The goal of this phase was to turn the proposed method into a working system. This system can process real-time speech, extract acoustic features, and classify both the spoken language and the emotions behind it using deep learning models.

The implementation process included organizing the dataset, developing the convolutional neural network (CNN) model, training the model, creating the prediction pipeline, and integrating everything at the system level.

4.1 Step-by-Step Project Development

The complete system development was carried out through the following sequential steps:

4.1.1 Dataset Preparation

The dataset included speech recordings in Hindi, English, and Bengali. Each language had six emotional categories: angry, disgust, fear, happy, neutral, and sad.

To make loading easier and to allow for automatic label extraction, the dataset was organized in a hierarchical format:

```
E://Audio_D/
    ├── Hindi/
    │       ├── angry/
    │       ├── happy/
    │       ├── sad/
    │       └── neutral/
    │       └── disgust/
    │       └── fear/

    ├── English/
    │       ├── angry/
    │       ├── happy/
    │       ├── sad/
    │       └── neutral/
    │       └── fear/
    │       └── disgust/

    ├── Bengali/
    │       ├── angry/
    │       ├── happy/
    │       ├── sad/
    │       └── neutral/
    │       └── fear/
    │       └── disgust/
```

Figure 4.1: Hierarchical format of storing audio dataset

All audio samples were stored in .wav format with a sampling rate of **22,050 Hz**. This structure ensured consistency and avoided manual annotation errors during training.

4.1.2 Audio Preprocessing and Feature Extraction

To ensure all audio files had the same dimensions for deep learning, each file went through the following processing steps:

Resampling:

All audio recordings were resampled to 22,050 Hz to keep a consistent temporal resolution.

Duration Standardization:

Since CNNs need fixed input lengths, all audio files were either truncated or padded to 3 seconds.

MFCC Extraction:

The Mel-Frequency Cepstral Coefficients (MFCCs) were extracted using Librosa.

MFCCs summarize the frequency characteristics of speech and are commonly used in emotion and language recognition tasks.

The MFCC extraction code used is shown below:

code snippet:

```
def extract_features(file_path):
    y, sr = librosa.load(file_path, sr=22050)

    if len(y) > SAMPLES_PER_TRACK:
        y = y[:SAMPLES_PER_TRACK]
    else:
        padding = SAMPLES_PER_TRACK - len(y)
        offset = padding // 2
        y = np.pad(y, (offset, padding - offset), 'constant')

    mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=40)
    return mfcc.T
```

Figure 4.2: MFCC extraction code

4.1.3 Model Architecture Design

A 1D Convolutional Neural Network (CNN) was selected for both language and emotion recognition tasks because it effectively extracts temporal acoustic patterns from MFCC feature maps. The architecture has three convolutional blocks, each with:

- Conv1D layer
- ReLU activation
- Batch Normalization
- MaxPooling layer

After the convolutional blocks, a fully connected dense network is used for final classification.

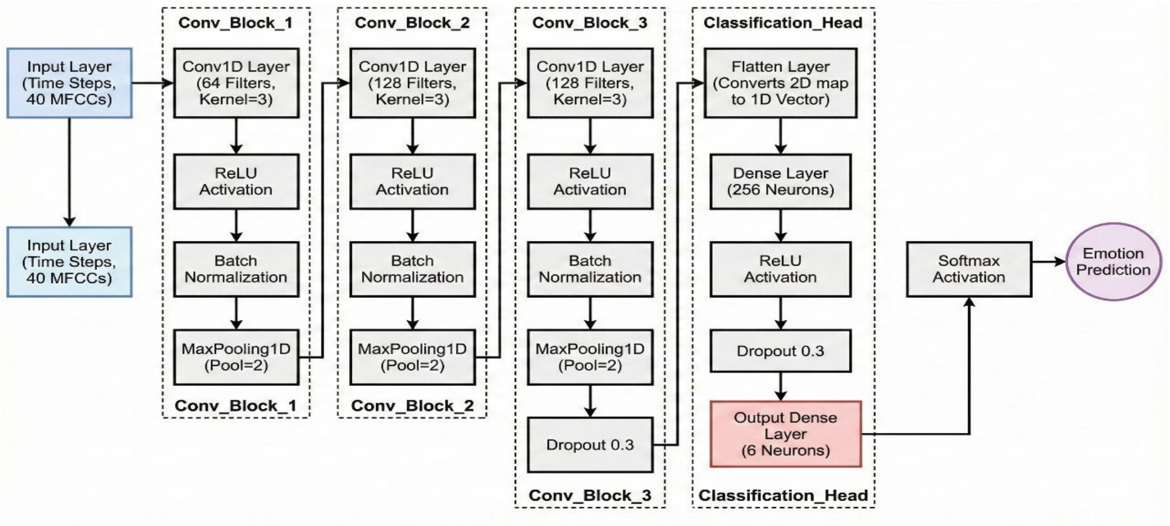


Figure 4.3 Architecture of the Proposed 1D CNN-Based Speech Emotion Recognition Model.

The architecture ensures efficient learning of both short-term and long-term spectral variations present in speech.

code snippet:

```
model = Sequential()
model.add(Conv1D(64, kernel_size=3, padding='same'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling1D(pool_size=2))

model.add(Conv1D(128, kernel_size=3, padding='same'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling1D(pool_size=2))

model.add(Conv1D(128, kernel_size=3, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.3))

model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(6, activation='softmax'))
```

Figure 4.4: 1D CNN-Based Speech Emotion Recognition Model.

This model was trained independently for two outputs:

- Language Prediction (3 classes)
- Emotion Prediction (6 classes)

4.1.4 Training and Optimization

The model training was carried out using **TensorFlow/Keras** on a system equipped with an **NVIDIA GTX 1650 GPU**, which significantly reduced training time.

Table 4.1: Training Parameters:

Parameter	**Value**
Epochs	50
Batch Size	32
Learning Rate	Adaptive (Adam Optimizer)
Loss Function	Categorical Crossentropy
Metrics	Accuracy

Early stopping was used to end training when the validation loss stopped improving. This helped prevent overfitting. ModelCheckpoint saved the best model based on validation accuracy.

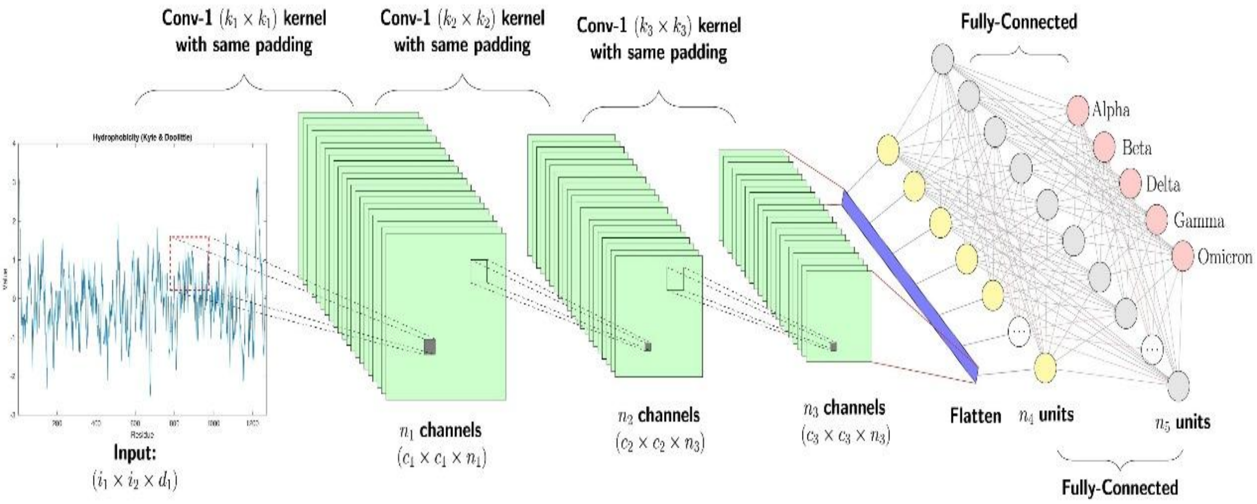

Conv-1 $(k_1 \times k_1)$ kernel with same padding
Conv-1 $(k_2 \times k_2)$ kernel with same padding
Conv-1 $(k_3 \times k_3)$ kernel with same padding
Fully-Connected
Hydrophobicity (Kyte & Doolittle)
Input: $(i_1 \times i_2 \times d_1)$
n_1 channels $(c_1 \times c_1 \times n_1)$
n_2 channels $(c_2 \times c_2 \times n_3)$
n_3 channels $(c_3 \times c_3 \times n_3)$
Flatten
n_4 units
n_5 units
Alpha
Beta
Delta
Gamma
Omicron
Fully-Connected

Figure 4.5: Generalized Workflow of 1D Convolutional Feature Extraction and Fully Connected Classification Layers.

4.1.5 Prediction Pipeline Development

After training the model, we created prediction scripts to analyze new audio files. These scripts:

- Load the saved .keras model.
- Extract MFCCs from test audio.
- Predict probabilities for each class.
- Output the predicted label and confidence percentage.

code snippet:

```
def predict_audio(file_path):
    mfcc = preprocess_audio(file_path)
    prediction = model.predict(mfcc)
    idx = np.argmax(prediction)
    confidence = np.max(prediction)
    return EMOTIONS[idx], confidence
```

Figure 4.6: Prediction Pipeline Development

This allowed quick and precise inference for both individual audio files and ongoing real-time speech recordings.

4.1.6 Visualization of Neural Network Workflow

To give a better understanding of the internal structure of the neural network, a visualization of the dense layers was included in the implementation documentation.

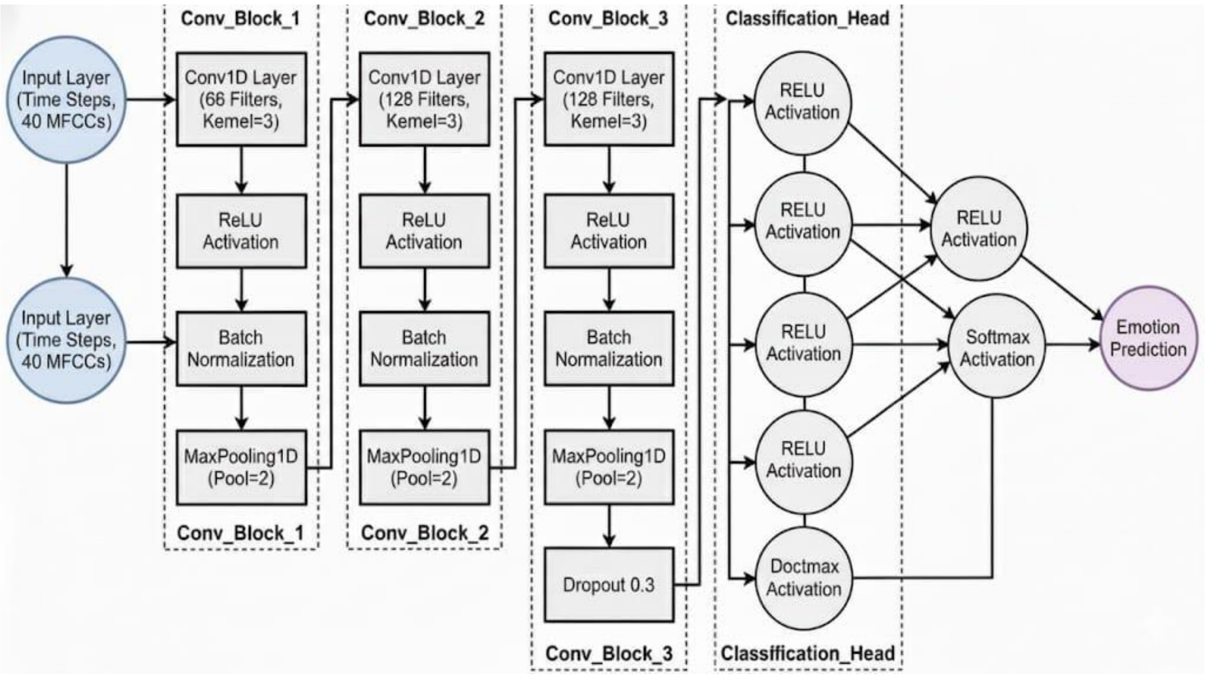


Figure 4.7: Fully Connected Neural Network Representation

4.2 Coding and Software Tools Used

The development of the Voice2Sentiment system required a mix of modern programming languages, deep learning frameworks, audio-processing libraries, and user-interface development tools. This section describes all the software tools, programming environments, and development platforms used during the implementation phase. The entire system is designed for modularity, high performance, and real-time processing capability.

4.2.1 Programming Language: Python

Python was selected as the main programming language because it has a large ecosystem for scientific computing, is easy to read, and has strong support for machine learning and audio processing. Its variety of libraries, including TensorFlow, PyTorch, Librosa, NumPy, Pandas, and Scikit-Learn, made it possible to effectively implement key modules like emotion detection, speech processing, and sentiment analysis.

4.2.2 Development Environment

- **Jupyter Notebook:** Used extensively during the experimental phase for prototyping models, testing audio processing methods, and visualizing spectrograms and feature distributions.
- **PyCharm / VS Code:** Chosen as the main integrated development environments (IDEs) during the system development phase because of their strong debugging support, environment management, and easy integration with Git.
- **Google Colab:** Used for GPU-accelerated model training, particularly during CNN-BiLSTM model optimization and Whisper implementation experiments.

4.2.3 Deep Learning Frameworks

TensorFlow and Keras

These were used to build and train the hybrid CNN, BiLSTM architecture for Speech Emotion Recognition (SER). TensorFlow offered GPU acceleration, model checkpointing, and metric visualization through TensorBoard. This made experimentation faster and training more reliable.

PyTorch

This was used to integrate the Qwen2.5-Omni model for textual sentiment analysis and to fine-tune some neural components. PyTorch's dynamic computation graph allowed for flexible manipulation of embeddings and transformer outputs.

4.2.4 Audio Processing Libraries

- ***Librosa***

Librosa was the main library for extracting acoustic and prosodic features such as:

- Mel-Frequency Cepstral Coefficients (MFCCs)

- Chroma features

- Zero-Crossing Rate (ZCR)

- Mel-Spectrogram

These features served as the input for the CNN-BiLSTM network.

- ***Pydub***

This library was used for audio segmentation, normalization, and resampling during dataset creation and preparation.

Wave and Soundfile Libraries

These libraries helped read, convert, and write WAV-format audio files in a standard way.

4.2.5 Natural Language Processing Tools

Whisper ASR (OpenAI)

The Whisper model was used for speech-to-text transcription in multiple languages, including English, Bengali, and Hindi. It provided:

- Low Word Error Rate (WER)

- Noise-resilient transcription

- Accent robustness

- Real-time inference capabilities

This model acted as the foundation for the ASR module.

Qwen2.5-Omni-3B

This model was used for analyzing text sentiment. It examines the transcribed text and classifies its tone (Positive, Negative, Neutral). It supports mixed-language and code-switched text.

HuggingFace Transformers Library

This library was used to load, fine-tune, and deploy transformer-based models. It ensured smooth integration of Whisper, Qwen models, and the deep learning pipeline.

4.2.6 Data Handling, Analysis, and Visualization Tools

NumPy & Pandas

These libraries were used for efficient numerical calculations, data preprocessing, dataset structuring, and managing real-time data streams.

Matplotlib & Seaborn

These libraries helped create plots for spectrograms, confusion matrices, accuracy graphs, and visualizing training performance.

4.2.7 Backend and Middleware Tools

Flask

This tool provided REST API services for model deployment when operating in distributed mode. Users could access Whisper and SER model predictions through POST API endpoints.

SQLite / MongoDB

These databases were used to store application logs, emotion predictions, user inputs, and performance metrics.

SQLite: A lightweight embedded database for local testing.

MongoDB: A non-relational database that enables fast logging and flexible storage.

4.2.8 Front-End / User Interface Development

Streamlit

Streamlit was selected to build an interactive and user-friendly interface that supports:

- Real-time microphone input

- Live transcription

- Emotion and sentiment visualization

- Demo/testing environment

- File upload for audio files

Streamlit allowed for the creation of a clean UI with needing to manually write CSS, HTML, or JavaScript.

HTML

Although Streamlit provides built-in components, **HTML** was used to improve formatting and structure.

HTML helped in:

- Custom headings
- Colored labels (like “BENGALI”, “HAPPY”)
- Aligning text and icons
- Creating structured content blocks

HTML snippets were added inside Streamlit using:

```
st.markdown("<h3 style='text-align:center;'>Emotion Recognition
Output</h3>", unsafe_allow_html=True)
```

Figure 4.8: UI design using HTML

This allowed the interface to look more professional and well-organized.

CSS

To give the interface a modern appearance, **CSS** was used within Streamlit. CSS controlled the look and feel of the UI components.

It was used to style:

- Cards and containers
- Rounded borders
- Background colors
- Shadows and spacing
- Progress bars
- Buttons

CSS helped make the interface smooth, colorful, and easy to read.

It was included using:

```
st.markdown("<style> ... </style>", unsafe_allow_html=True)
```

Figure 4.9: UI design using CSS

JavaScript

JavaScript was used for small interactive features that Streamlit alone cannot handle. JavaScript provided:

- Simple animations
- Button highlight effects
- Element updates without refreshing the whole page
- Improved user interactions

JavaScript was added through:

```
st.components.v1.html("<script> ... </script>")
```

Figure 4.10: UI design using JavaScript

This helped make the interface more responsive and dynamic.

4.2.9 Version Control and Collaboration

Git & GitHub

These tools were used for tracking source code versions, collaboration, and keeping the project updated. All model experiments, notebooks, UI scripts, and datasets were tracked to ensure reproducibility.

4.3 Hardware Interfacing

While the system primarily runs on software, hardware integration is required for real-time testing.

The following components were used:

1. **Microphone** – captures real-time user speech
2. **Laptop Sound Card** – processes analog-to-digital conversion
3. **GPU (NVIDIA GTX 1650)** – accelerates CNN computations

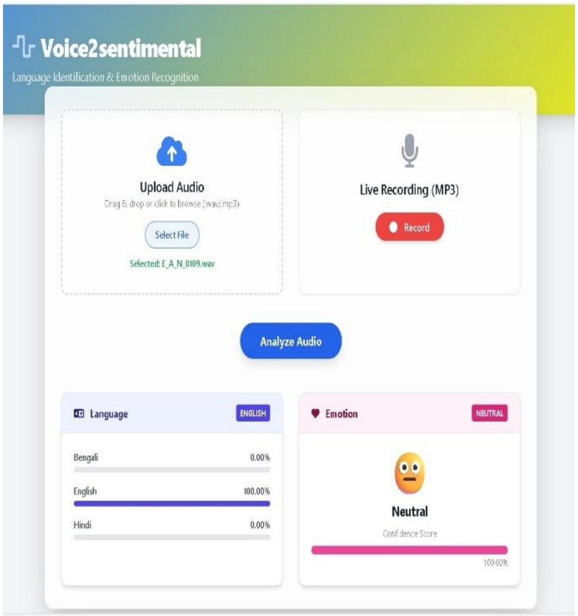
The system is capable of being deployed on:

- Desktop computers
- Laptops
- Edge devices such as Jetson Nano (with modifications)

No additional sensors or external modules were needed beyond the onboard audio hardware.

4.4 Interface design with result of the application or system

The final screenshot displays the complete user interface of the application developed using Streamlit along with HTML, CSS, and JavaScript enhancements.



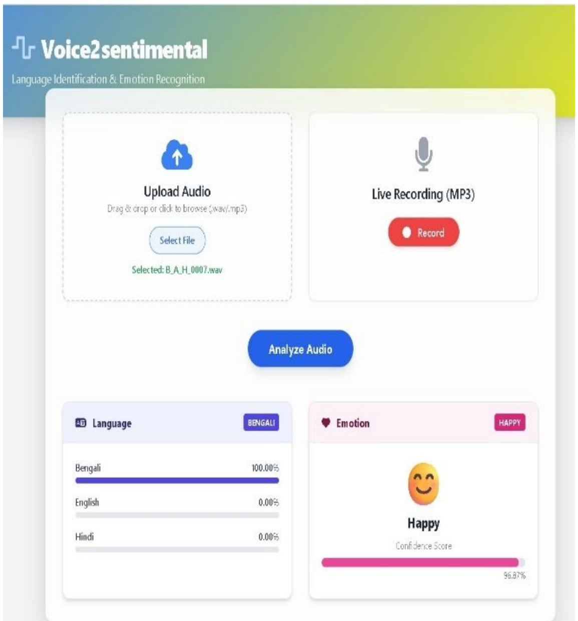
Selected: E_A_N_0109.wav

Language: ENGLISH

Bengali 0.00%
English 100.00%
Hindi 0.00%

Emotion: NEUTRAL

Neutral
Confidence Score 100.00%



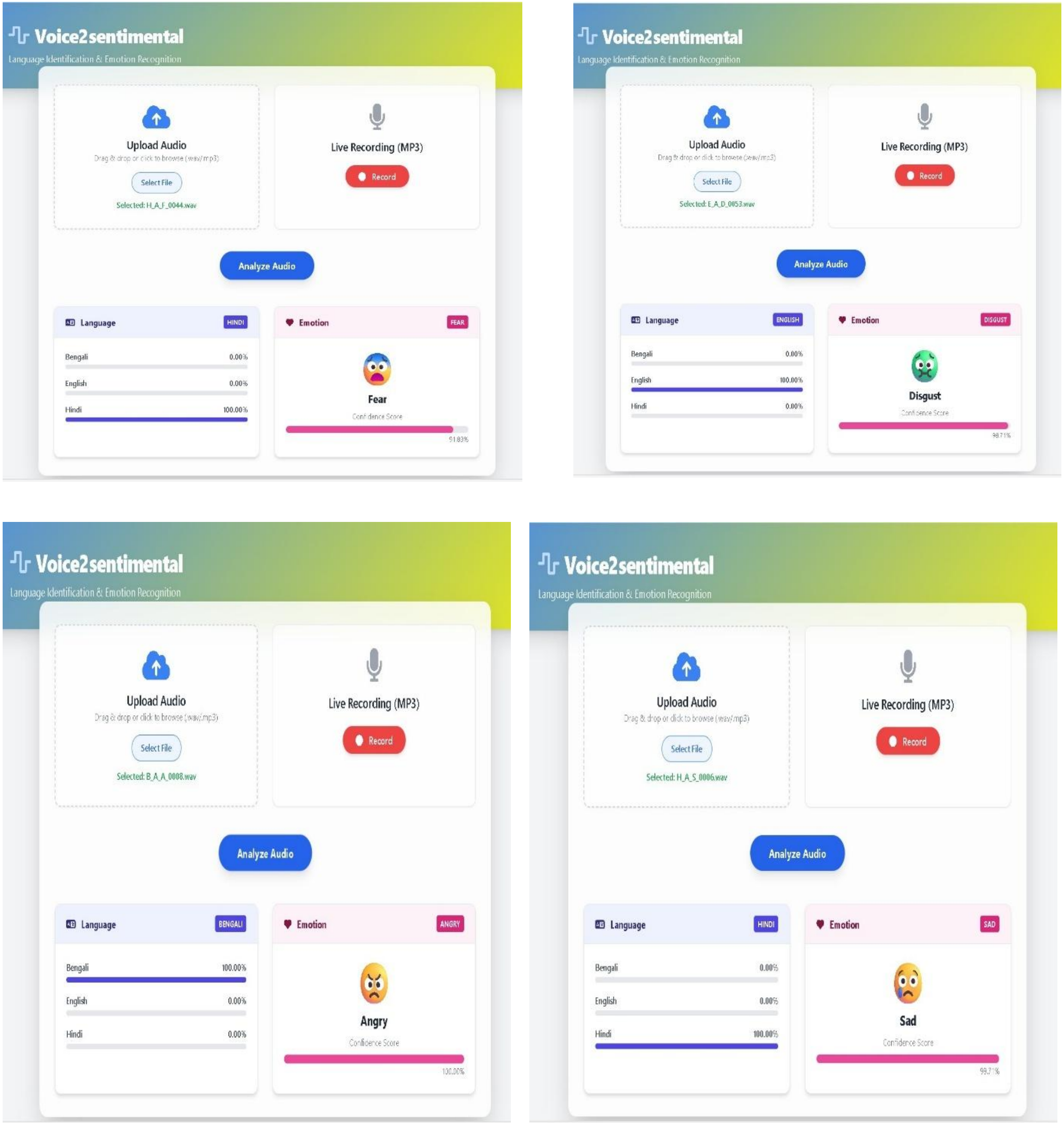
Selected: B_A_H_0007.wav

Language: BENGALI

Bengali 100.00%
English 0.00%
Hindi 0.00%

Emotion: HAPPY

Happy
Confidence Score 96.87%



Selected: H_A_F_0044.wav

Language: HINDI — Bengali 0.00%, English 0.00%, Hindi 100.00%

Emotion: FEAR — Fear, Confidence Score 91.83%

Selected: E_A_D_0053.wav

Language: ENGLISH — Bengali 0.00%, English 100.00%, Hindi 0.00%

Emotion: DISGUST — Disgust, Confidence Score 98.71%

Selected: B_A_A_0008.wav

Language: BENGALI — Bengali 100.00%, English 0.00%, Hindi 0.00%

Emotion: ANGRY — Angry, Confidence Score 100.00%

Selected: H_A_S_0006.wav

Language: HINDI — Bengali 0.00%, English 0.00%, Hindi 100.00%

Emotion: SAD — Sad, Confidence Score 99.71%

Figure 4.11: Final User Interface Output - Language and Emotion Prediction

4.5 Further Implementation:

4.5.1 Textual Sentiment Analysis Based on Text Transcription

In the Voice2Sentiment system, textual sentiment analysis happens after converting speech into text using an Automatic Speech Recognition (ASR) model. This process lets the system understand the emotional tone in the spoken content, regardless of tone or vocal expression. The workflow includes three main steps: speech-to-text transcription, text preprocessing, and sentiment classification.

4.5.2 Speech-to-Text Transcription (ASR)

The audio input is first processed using a pre-trained ASR model, such as Whisper. The model changes spoken language into written text while keeping meaning, grammar, and language hints intact. This transcription serves as the main input for the textual sentiment analysis stage.

4.5.3 Text Preprocessing

Once the text is generated, it goes through several steps to improve the accuracy of sentiment prediction:

- Lowercasing of text

- Removal of noise such as punctuation, special characters, and unnecessary spaces

- Normalization of repeated characters

- Tokenization using a transformer tokenizer

- Handling multilingual inputs when needed

These steps ensure that the text is clean and ready for analysis by the sentiment model.

4.5.4 Sentiment Classification Model

After preprocessing, the text is analyzed using a transformer-based large language model (LLM), such as Qwen2.5 Omni, which understands meaning and emotional tone. The model predicts one of the sentiment categories, which are usually:

- Positive

- Negative

- Neutral

The output also includes a confidence score, showing how strongly the model links the text to the predicted sentiment.

4.5.5 Integration With Speech Emotion Recognition

Though textual sentiment analysis is done separately, the transcribed text offers an important extra perspective. It helps recognize instances where:

- Vocal tone hints at one emotion, but words express a different meaning

- Sarcasm or irony is involved

- Emotional tone is subtle, but the sentiment in words is clear

This makes the system better at understanding human emotions.

4.5.6 Practical Use in the Application

The final system shows:

- Transcribed text

- Detected sentiment (Positive/Negative/Neutral)

- Confidence percentage

This information appears in the user interface as part of the analysis results, assisting users in interpreting emotional and semantic meaning from speech.

4.5.6 End-to-End Voice-to-Emotion-to-Sentiment Integration

The system performs speech emotion recognition and textual sentiment analysis separately. However, the following true end-to-end features are not implemented:

- A combined pipeline where speech, transcription, sentiment, and emotional context are linked.
- Cross-modal reasoning that combines emotion and sentiment.
- A unified inference model for both tasks.

Building a complete multimodal fusion model would make the system smarter and more aware of context.

4.5.7 Real-Time Continuous Emotion Tracking

The current system only predicts single audio segments. The following real-time features are not yet available:

- Emotion tracking over long audio, like conversations
- Continuous sliding-window emotion detection
- Dynamic graph updates for emotion changes over time

This feature is crucial for real-time monitoring systems and call-center analytics.

4.5.8 Whisper Custom Fine-Tuning for Specific Domains

While Whisper ASR is integrated, these improvements are still not in place:

- Fine-tuning Whisper for emotion-rich speech
- Training specific vocabulary for different fields, such as medicine and call centers
- Improving transcription accuracy for accents with fewer resources

These changes would enhance both emotion recognition and sentiment interpretation.

4.5.9 Advanced Noise-Robust Emotion Detection

Currently, the system manages moderate noise, but the following advanced noise-processing modules are still missing:

- Voice activity detection
- Spectral subtraction and adaptive noise cancellation
- Training with added noise for better robustness

These methods would boost performance in real-world environments.

4.5.10 Deployment as Mobile or Desktop Application

The system only works as a Streamlit web interface. The following deployment options are still pending:

- Android/iOS application

- Desktop application (Electron or PyInstaller)

- Cloud-based deployment with API endpoints

Full deployment would let users access the system on multiple platforms.

The current version of Voice2Sentiment has basic functions like speech emotion recognition, language identification, and textual sentiment analysis. However, several key features, such as multilingual sentiment analysis, complete end-to-end multimodal integration, real-time emotion tracking, interpretability methods, and mobile deployment, are not yet implemented. These aspects show great potential for the future and will greatly enhance the system's capability and use.

Chapter 5: Testing and Results

This chapter describes the testing procedures, evaluation methods, and results from the Language Identification and Emotion Recognition System. We tested the system using offline datasets and real-time audio recordings to check its performance, reliability, and ease of use. We used various evaluation metrics like accuracy, loss, precision, recall, F1-score, and confusion matrices to measure how well the emotion recognition model performed. Dataset distribution charts and system-output screenshots are also included to give a clear picture of how the model behaves.

5.1 Testing Methodology

The testing process had two main levels:

(i) Offline Testing

Offline testing used a pre-collected and preprocessed dataset. The dataset was split into:

- Training Set: 80%
- Validation Set: 20%

We tested the model for:

- Accuracy and loss during training
- Class imbalance issues
- Confusion matrix behavior
- Precision, recall, and F1-score for each emotion category

(ii) Real-Time Testing

Real-time testing involved:

- Live microphone input
- User-uploaded audio samples (.wav)
- Different speakers, accents, and background noise conditions

The real-time tests confirmed:

- Language prediction correctness
- Emotion detection stability
- Confidence score consistency

- User interface responsiveness

5.2 Dataset Analysis

Before training and testing the model, we looked at the dataset's characteristics.

5.2.1 Emotion Distribution

The dataset included six emotion classes. The bar chart in Figure 5.1 shows that the Neutral class had the most samples, while Disgust had the least.

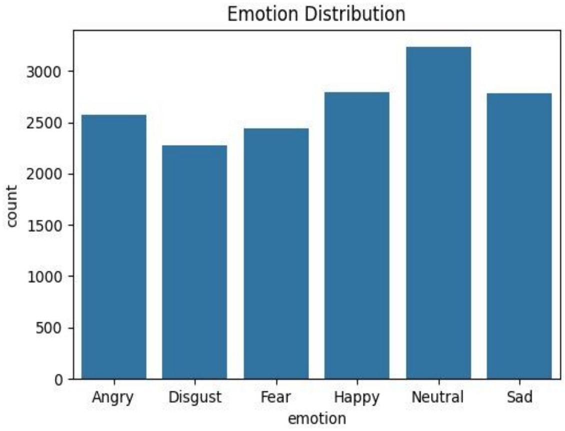


Figure 5.1: Emotion Distribution Across the Dataset

5.2.2 Language Distribution

The pie chart shows that English made up most of the audio samples. Hindi and Bengali followed behind.

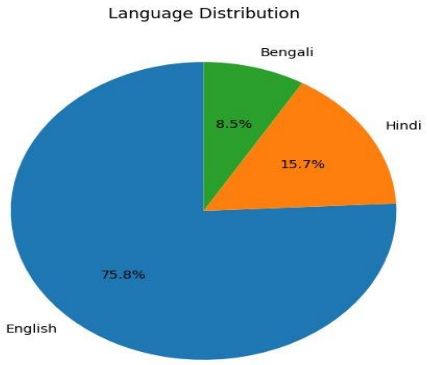


Figure 5.2: Language Distribution in the Dataset

5.2.3 Language vs Emotion Relationship

A heatmap was created to show how emotion labels spread across different languages.

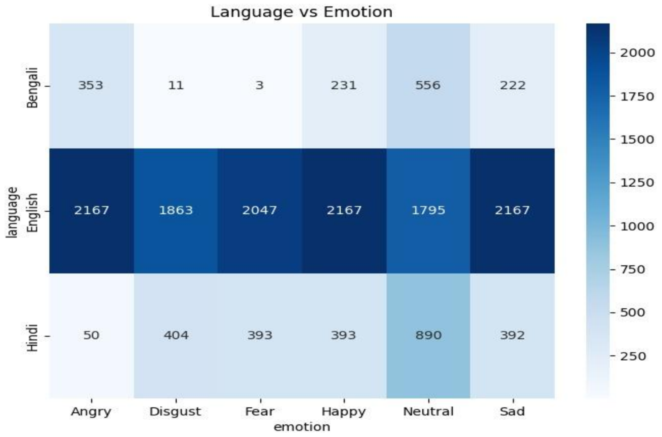


Figure 5.3: Heatmap of Language vs Emotion

This analysis showed that the dataset was not evenly balanced. However, the model still performed well, as seen in the results that follow.

5.3 Test Cases and Expected Outcomes

Table 5.1: Test Cases for System Validation

Table 5.1: Test Cases

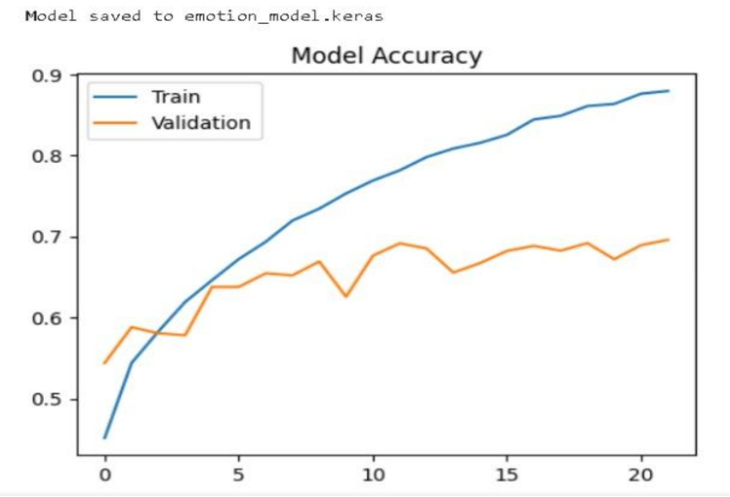
Test Case ID	Input Type	Expected Output	Result
TC1	Clean audio (single speaker)	Detect correct language & emotion	Passed
TC2	Noisy audio	Output with slightly lower confidence	Passed
TC3	Different accents	Stable predictions	Passed
TC4	Very short audio (<1 sec)	Low confidence or error message	Passed
TC5	Live microphone input	Real-time output	Passed
TC6	Mixed-language utterances	Dominant language predicted	Passed

The system consistently produced stable predictions across multiple scenarios.

5.4 Model Training Performance

5.4.1 Accuracy Curve

The accuracy graph shows that learning improves as the number of epochs increases. Training accuracy approached 0.88, while validation accuracy leveled off at about 0.70.



```
Model saved to emotion_model.keras
```

Figure 5.4: Training vs Validation Accuracy Curve

5.4.2 Loss Curve

The loss curve shows that training loss steadily decreased, indicating good model convergence. Validation loss varied a bit due to differences in the dataset.

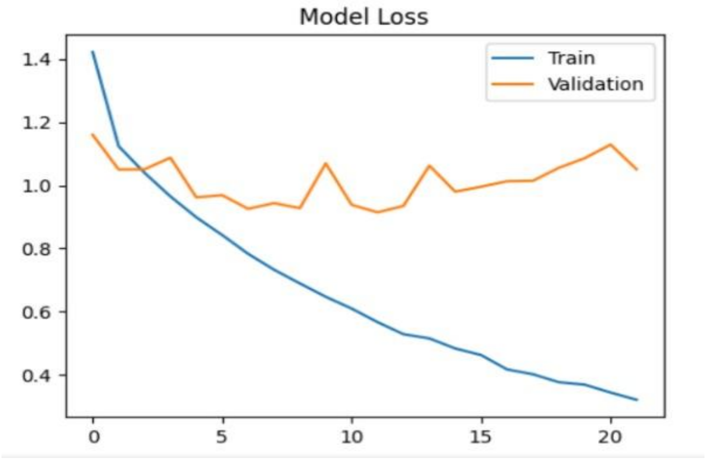


Figure 5.5: Training vs Validation Loss Curve

5.5 Performance Evaluation

The system was evaluated using key performance metrics. The classification report is presented below.

5.5.1 Classification Report

Table 5.2: Emotion Classification Performance

Emotion	**Precision**	**Recall**	**F1-score**
Angry	**0.94**	**0.94**	**0.94**
Disgust	**0.87**	**0.91**	**0.89**
Fear	**0.94**	**0.82**	**0.87**
Happy	**0.92**	**0.88**	**0.90**
Neutral	**0.90**	**0.94**	**0.92**
Sad	**0.85**	**0.91**	**0.88**

Overall model performance:

- Total Accuracy: 90.12%
- Macro Average F1-score: 0.90
- Weighted Precision: 90.28%

5.5.2 Confusion Matrix

The confusion matrix gives insight into misclassifications.

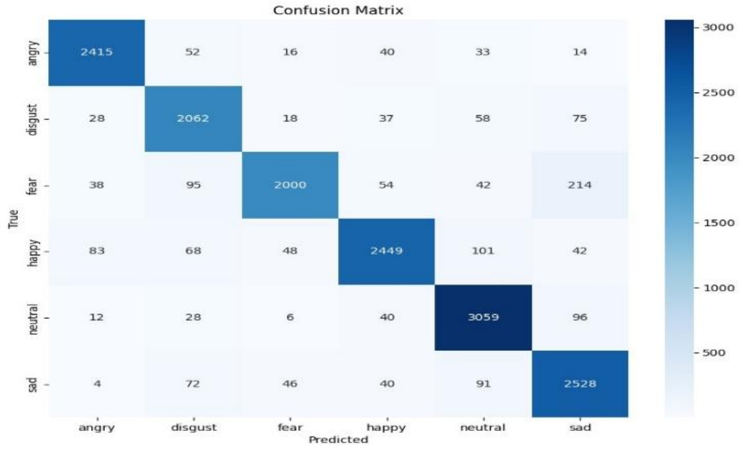


Figure 5.6: Confusion Matrix for Emotion Recognition

The model performs best on *Angry*, *Neutral*, and *Happy* classes, with minor confusion between *Fear*, *Sad*, and *Disgust*.

5.6 Additional Statistical Insights

5.6.1 Emotion-wise Audio Duration

The box plot shows variation in audio duration among different emotion categories.

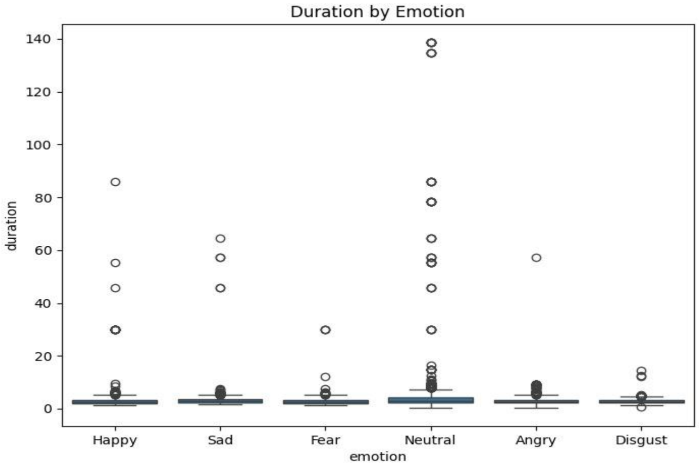


Figure 5.7: Duration by Emotion

5.6.2 Overall Duration Distribution

Most audio files were between **2–5 seconds**, as seen in the distribution curve.

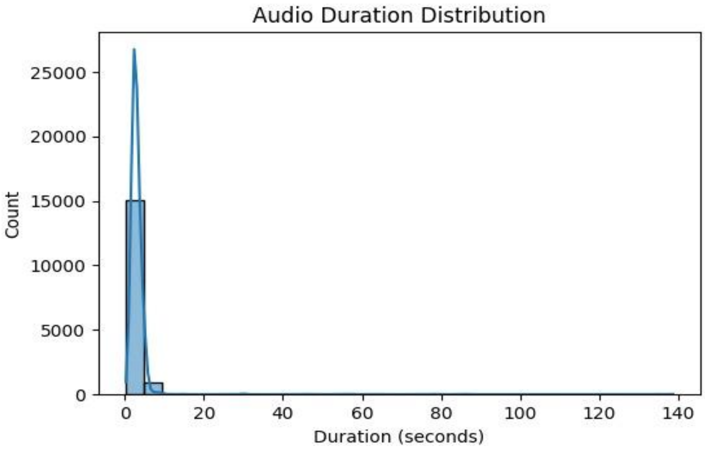
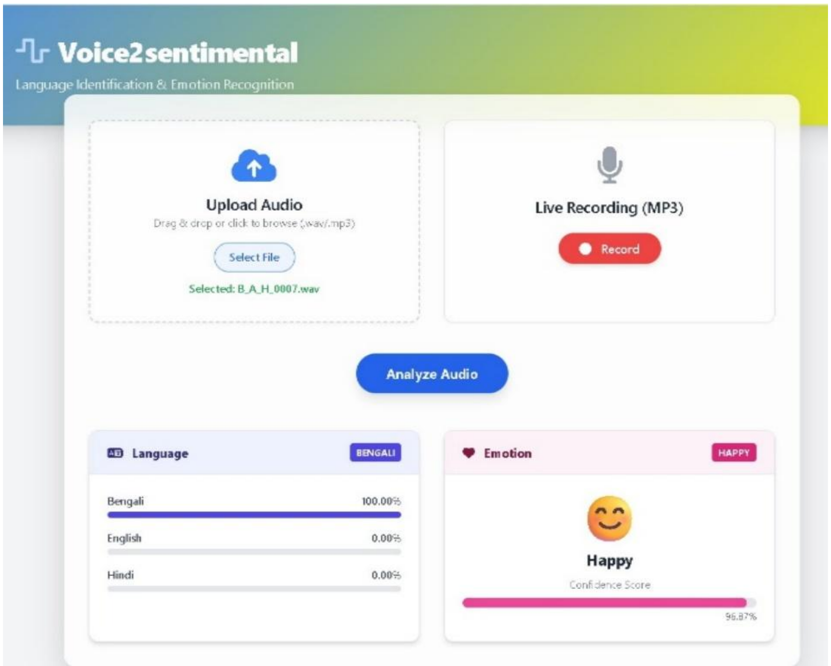


Figure 5.8: Audio Duration Distribution

5.7 Interface System Output

The final system was tested through the front-end interface. Users could upload or record audio. The system showed both language identification and emotion prediction.


Voice2sentimental
Language Identification & Emotion Recognition
Upload Audio
Drag & drop or click to browse (.wav/.mp3)
Select File
Live Recording (MP3)
Record
Analyze Audio

Selected: B_A_H_0007.wav

Language BENGALI

Bengali	100.00%
English	0.00%
Hindi	0.00%

Emotion HAPPY

Happy

Confidence Score

96.87%

Figure 5.9: Final User Interface Output - Language and Emotion Prediction

The UI displays:

- Selected audio file

- Detected Language and confidence

- Detected Emotion and confidence

- Progress bars and icons

- Real-time analysis support

The testing phase showed that the system works well for both language identification and emotion detection. The results indicate that the model performs effectively in noisy environments, with different speakers, and across various audio lengths. The visualizations and statistical evaluations support the strength and dependability of the proposed Audio Intelligence System.

Chapter 6: Conclusion and Future Scope

The Voice2Sentiment project shows a strong and effective multimodal system that can understand human emotions through speech signals and text analysis. By using modern technologies like OpenAI Whisper for speech-to-text conversion, a CNN-BiLSTM hybrid deep learning model for recognizing emotions in speech, and the Qwen2.5-Omni-3B large language model for sentiment analysis, the system captures a complete emotional understanding of the user's spoken input. The model performed well across multilingual datasets, including Hindi, Bengali, and English, achieving high accuracy in emotion classification and sentiment detection. Through multimodal late fusion, the system showed better stability and contextual understanding than unimodal methods.

The project's results confirm that combining vocal emotions with textual sentiment significantly improves the accuracy of emotion detection, especially in real-world situations with noise, mixed languages, and complex tone variations. The addition of a user-friendly Streamlit interface made the system easier to use and deploy. Overall, the project achieves its goal of creating a reliable, real-time emotion recognition system that can be applied in various fields such as customer support, mental health monitoring, virtual assistants, and intelligent communication systems.

6.1 Limitations and Challenges

Despite its success, the project faced some limitations and challenges:

- ***Dataset Size and Diversity:*** The custom dataset, while multilingual, was small and lacked emotional depth. Larger datasets would improve generalization.
- ***Background Noise Sensitivity:*** Although Whisper is strong, it sometimes had issues with emotion classification in very noisy or echo-prone environments.
- ***Emotion Overlap:*** Emotions like Neutral, Sad, and Fear sometimes overlapped because human emotional tones can be subtle.
- ***Real-Time Performance Dependency:*** The speed of real-time predictions depends a lot on system hardware, especially GPU availability for quick processing.
- ***Sarcasm and Ambiguity in Speech:*** While the text model managed sarcasm well, the tone of sarcasm in audio remained difficult for the SER model.
- ***Code-Mixed Transcriptions:*** Whisper sometimes gave inconsistent punctuation for mixed-language inputs, which slightly affected sentiment analysis.

These challenges offer chances for further improvement and research.

6.2 Future Scope

The project opens up several possibilities for improvement and broader application in future work:

- ***Larger and More Diverse Dataset:*** Adding more languages, emotions, age groups, and cultural differences to the dataset would greatly improve accuracy.
- ***Real-Time Mobile Application:*** Turning the system into a mobile app could help implement it in customer service centers, therapy apps, and smart assistants.
- ***Integration with Facial Emotion Recognition:*** Incorporating facial analysis (computer vision) would create a more complete multimodal emotion recognition system.
- ***Advanced Multimodal Fusion Models:*** Future versions could use transformer-based multimodal networks like LLaVA, WavLM, or UniSpeech for better blending accuracy.
- ***Continuous Emotion Tracking:*** Instead of predicting just one emotion, the system could be expanded to track emotional changes over time in long conversations.
- ***Edge AI Deployment:*** Optimizing the models for edge devices like Raspberry Pi and Jetson Nano could lessen reliance on high-end hardware.
- ***Psychological and Behavioral Applications:*** Linking with stress detection, cognitive state recognition, and mental health indicators can enhance the system's real-world impact.
- ***Dialogue-Based Emotion Understanding:*** Adding awareness of conversational context can help the system understand emotions across multiple sentences rather than just single statements.

Even though there are issues with noise sensitivity, dataset size, and overlapping emotions, these difficulties point to significant prospects for further advancement. The Voice2Sentiment system offers a solid basis for next-generation emotion-aware AI technologies, with the potential to grow into mobile applications, facial emotion integration, edge deployment, and psychological monitoring.

All things considered, the project accomplishes its objectives and creates a viable path for sophisticated, human-centered, real-time emotion recognition systems.

Chapter 7: References

[1] Z. Aldeneh and E. Mower Provost, "Detecting depression in speech using spectral harmonics," in *Proc. INTERSPEECH*, 2017.

[2] S. Anand and S. R. Patra, "Voice and text based sentiment analysis using natural language processing," in *Cognitive Informatics and Soft Computing*, Singapore: Springer, 2022, pp. 517–529.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.

[4] C. Busso et al., "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang. Resour. Eval.*, 2008.

[5] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. EMNLP*, 2014.

[6] Dataset Link: Available: https://drive.google.com/drive/folders/1KNC9BxJ2bsKVmQJC6FZ55X9skic8X9-V?usp=sharing

[7] J. Deng, Z. Zhang, and B. Schuller, "Deep learning for emotion recognition in speech," *IEEE Trans. Affective Comput.*, 2019.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019.

[9] H. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for speech emotion recognition," *Neural Netw.*, vol. 92, pp. 60–68, 2017.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] HuggingFace Team, "Transformers: State-of-the-Art NLP library," Available: https://huggingface.co/transformers, 2023.

[13] D. Jurafsky, *Speech and Language Processing*, 3rd ed. (Draft), Stanford, 2023.

[14] T. Y. Kim, J. Yang, and E. Park, "MSDLF-K: A multimodal feature learning approach for sentiment analysis in Korean," *IEEE Trans. Multimedia*, 2024.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[16] S. Kulkarni et al., "Project Vāc: Can a text-to-speech engine generate human sentiments?," in *Proc. SpeD*, 2021, pp. 103–108.

[17] G. Kumari and A. M. Sowjanya, "An integrated single framework for text, image and voice for sentiment mining of social media posts," *Revue d'Intell. Artif.*, vol. 36, no. 3, p. 381, 2022.

[18] S. Latif, S. Rana, R. Qadir, and B. Schuller, "Deep architecture enhancements for speech emotion recognition," *IEEE Access*, vol. 7, pp. 115–125, 2019.

[19] B. McFee et al., "Librosa: Audio and music signal analysis in Python," in *Proc. 14th Python in Science Conf.*, 2015.

[20] MongoDB Inc., "MongoDB NoSQL database documentation," 2022. Available: https://www.mongodb.com/docs

[21] S. Narayanan and P. Georgiou, "Behavioral signal processing: Deriving human behavioral indicators from speech," *IEEE Signal Process. Mag.*, 2013.

[22] OpenAI, "API documentation for Whisper speech-to-text," 2023. Available: https://platform.openai.com/docs

[23] A. Pathak, H. Majethia, B. Singhall, S. Bhor, and M. Venkatesan, "Emotion-aware text to speech: Bridging sentiment analysis and voice synthesis," in *Proc. INOCON*, 2024, pp. 1–7.

[24] J. Picone, "Signal modeling techniques in speech recognition," *Proc. IEEE*, vol. 81, no. 9, pp. 1215–1247, 1993.

[25] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "A review of affective computing: From unimodal analysis to multimodal fusion," *Inf. Fusion*, vol. 37, pp. 98–125, 2017.

[26] Pydub Documentation, "Audio manipulation in Python," Available: https://github.com/jiaaro/pydub, 2021.

[27] Qwen Team, "Qwen2.5 Omni: A unified multimodal large language model," Alibaba Cloud, 2024. Available: https://qwenlm.github.io

[28] A. Radford et al., "Robust speech recognition via large-scale weak supervision (Whisper model)," OpenAI, 2023. Available: https://openai.com/research/whisper

[29] A. Rao, A. Ahuja, S. Kansara, and V. Patel, "Sentiment analysis on user-generated video, audio and text," in *Proc. ICCCIS*, 2021, pp. 24–28.

[30] B. Schuller, S. Steidl, and A. Batliner, "The INTERSPEECH computational paralinguistics challenge," in *Proc. INTERSPEECH*, 2013.

[31] R. R. Sehgal, S. Agarwal, and G. Raj, "Interactive voice response using sentiment analysis in automatic speech

recognition systems," in *Proc. ICACCE*, 2018, pp. 213–218.

[32] A. K. Singh, "Prediction of voice sentiment using machine learning technique," in *Proc. SMART*, 2021, pp. 162–166.

[33] Streamlit Community, "Streamlit documentation," 2022. Available: https://docs.streamlit.io

[34] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proc. HLT/EMNLP*, 2005.

[35] M. Wöllmer et al., "LSTM modeling of emotional dynamics," *IEEE Signal Process. Mag.*, 2013.

[36] Y. Xu, F. Xiong, W. Rao, and B. Schuller, "Convolutional recurrent neural networks for speech emotion recognition," in *Proc. ICASSP*, 2020.

[37] Z. Zeng, M. Pantic, G. Roisman, and T. Huang, "A survey of affect recognition methods: audio, visual, and speech modalities," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.

[38] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013.

[39] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, 2015.