A

PROJECT SCHOOL REPORT

on

SPEECH EMOTION RECOGNIZER USING TRANSFORMERS

Submitted By

Abhijeet Gowlikar	245322733129
Akanksha Rondla	245322733132
Gangula Venkata Raja Vineela	245322733143
Indrakar Gaurav	245322733150
Shaik Ayisha	245322733180
Udata Lekhana Surya Bhanu	245322733188

Under the guidence of

P V N Balarama Murthy

NEIL GOGTE INSTITUTE OF TECHNOLOGY

Kachavanisingaram Village, Hyderabad, Telangana 500058.

NEIL GOGTE INSTITUTE OF TECHNOLOGY
A Unit of Keshav Memorial Technical Education (KMTES)
Approved by AICTE, New Delhi & Affiliated to Osmania University, Hyderabad

CERTIFICATE

This is to certify that the project work entitled "Speech Emotion Recognition" *is a bonafide work carried out by* ABHIJEET GOWLIKAR (245322733129), AKANKSHA RONDLA (245322733132), GANGULA VENKATA VINEELA (245322733143), INDRAKAR GAURAV(245322733150), SHAIK AYISHA(245322733180), UDATA LEKHANA SURYA BHANU(245322733188) of III year V semester Bachelor of Engineering *in* CSE *during the academic year* 2024-2025 and *is a record of bonafide work carried out by them*.

P V N Balarama Murthy

Project Mentor

ABSTRACT

Speech Emotion Recognition (SER) is a vital task in various domains, ranging from human-computer interaction (HCI) and customer service to mental health analysis. Speech Emotion Recognition has become one of the most important and challenging task in real world and it has also gained popularity in recent years. Initially we started with the Deep neural network models which include CNN's (Convolutional Neural Network), ANN's(Artifical Neural Network), LSTM(Long Short-Term Memory), ResNet9((Residual Network) , RNN(Residual neural network) from scratch using pytorch.

Transformers are advanced deep learning architectures well-suited for sequential data processing. While Transformers are widely used in natural language processing, their ability to capture long-range dependencies and complex temporal relationships makes them highly effective for our project when applied to sequential audio data. In our project, we utilized a Transformer-based model to detect and classify emotions from speech signals. The system processes audio signals using librosa to extract features, which are then encoded as sequential inputs for the Transformer. Our model employs an embedding layer followed by a Transformer encoder with two layers, leveraging the multi-head self-attention mechanism to capture intricate temporal dependencies in the speech data. Additionally, the feed-forward network and linear classifier effectively map the learned representations to the target emotion classes. Extensive training and evaluation on publicly available datasets demonstrate the model's ability to achieve robust and accurate emotion classification.

This work paves the way for more robust and adaptive SER systems, which can be applied to a wide range of fields such as virtual assistants, healthcare, automated call

centers providing valuable insights into emotional states and facilitating more personalized user experiences.

TABLE OF CONTENTS

S. NO.	TITLE	PAGE NO
	ABSTRACT	3
	TABLE OF CONTENTS	5
	LIST OF FIGURES	6
1	**Introduction** 1.1 Problem Statement and Objectives 1.2 Motivation 1.3 Scope	8
2	**Literature Survey**	11
3	**Proposed Work, Architecture, Technology Stack & Implementation Details** 3.1 Proposed Work 3.2 Technology Stack 3.3 Implementation Details	15
4	**Results & Discussions** 4.1 Result 4.2 Output screens	27
5	**Conclusion & Future Scope**	32
6	**References**	34

LIST OF FIGURES

Fig. No.	Figure Name	Page No.
1	Accuracy Graph	14
2	Emotion classes in Dataset	20
3	Transformer Model Architecture	26
4:	Confusion Matrix	27
5	Train and validation accuracy curves	28
6	Precision recall and F1 score	28
7:	UI HomePage	29
8:	SignUp Page	30
9	SignIn Page	30
10	Predicted Output	31

LIST OF TABLES

Table No.	Table Name	PageNo.
1	Data preprocessing summary	19

CHAPTER 1

INTRODUCTION

1.1 Problem Statement and Objectives

Statement: Modeling human emotions in speech signals is challenging due to high computational demands and limited emotion-labeled data. This project aims to build a trasformer based model for Speech Emotion Recognition .The focus is on creating a scalable, accurate solution for real-world applications.

Objectives:

1. To build a transformer based model for accurate Speech Emotion Recognition (SER).
2. Overcome data scarcity with techniques like data agumentation.
3. Ensure Scalability
4. Testing the model with the different audio-clips.
5. Optimize the resource utilization for real-world deployment.
6. Evaluate the model using metrics to validate is accuracy and performance.

1.2 Motivation

The ability to accurately recognize emotions from speech is a powerful tool with wide-ranging applications in both personal and professional contexts. Speech Emotion Recognition (SER) enhances the Human-Computer interaction by making systems emotionally aware, enabling more natural communication. It plays a crucial role in mental health monitoring by dectecting emotional patterns in speech. SER improves customer service by analyzing caller emotions and provinding real-time feedback. It aids in security, forensics recognizing stress,

excitement. Additionally, SER supports education and accessibility, personalizing learning and helping those with speeech disorders convey emotions.

The motivation for this work stems from the need to enhance the performance of existing models in recognizing emotions from speech data. With the rapid advancements in artificial intelligence and machine learning, the demand for real-time applications of emotion recognition in various industries is increasing. Virtual assistants, mental health apps, and customer service automation systems all require more accurate and adaptive models to cater to user's emotional needs.

1.3 Scope

The scope of this project is to enhance Speech Emotion Recognition (SER) by refining a Transformer model that directly analyzes raw speech signals, focusing on features such as tone, pitch, rhythm rather than relying on text transcripts. Most existing models in SER typically convert speech into text through speech-to-text (STT) systems before analyzing the emotional content.

Speech-to-text models can often struggle with accuracy, particularly in noisy environments or when the speaker has a non-standard accent or dialect. Errors in transcription can lead to misinterpretation of the emotional content, as emotions in speech are often conveyed through tone, pitch, and rhythm rather than the literal meaning of words.

This model can be optimized for real-time applications where immediate emotional feedback is critical. For instance, virtual assistants, automated customer service systems, or mental health apps can benefit from faster and more

accurate emotion recognition based on tone and speech dynamics rather than relying on the delay introduced by transcription.

The scope of our project can be further expanded by exploring the combination of audio-based emotion recognition facial expression recognition to create a more comprehensive multimodal system. This would provide even deeper insights into emotional states, leading to better user experiences in applications like mental health support, virtual assistants.

CHAPTER 2

LITERATURE SURVEY

1.1 Existing Models

There are several existing models for Speech Emotion Recognition such as **Wav2Vec 2.0. ,HuBERT.**

Wav2Vec.2.0 is a pre-trained self-supervised model developed by Facebook AI, it was orginally used for the speech recognition but adaptable for emotion recognition. It learns fro m directly raw audio, removing the need for extensive labeled data. The model consist of CNN feature extractor follwed by a Transformer encoder, allowing it to capture both local and global dependencies in speech. It handles the noise perfectly and genralizes it across all languages. The model can classify the Emotions like happiness, anger, and sadness effectively. It is easily adaptable for various speech tasks. It is pre-trained on large speech datasets like LibriSpeech and CommonVoice, containing hours of unlabed speech for SER it is fine-tuned on labeled emotion datasets such as RAVDESS and TESS.

HuBERT is also one of the popular model used for the Speech Emotion Recognition which is also developed by the Facebook AI, is a self-supervised model trained to predict hidden units from masked audio segments. It learns speech representations without labeled data by clustering audio fearures iteratively. This model uses the combination of the CNNs and Transformers, making it efficient for capturing both short-term and long-term dependencies in speech. It performes well in the less resource scenarios making it effective for the tasks SER with limited emotion labled datasets. It is designed to deal with the

audio in the noisey environment. It can classify the emotions such as happiness, sadness , anger , fear, surprise, disgust, neutral. It is also pre-trained on dtasets like LibriSpeech and CommonVoice and fine-tunned for SER on datasets like RAVDESS and CREMA-D.

1.2 Key Techniques and Algorithms:

We used ANN where it takes 40 input features, such as Mel-Frequency Cepstral Coeffficients (MFFCCs), and predicts one of five emotional labels: happy,sad, neutral,disgust, or anger. It consist of the 3 fully connected layers. The first layer maps the 40 input features to 128 units, the second layer further processes the output with 128 units, and the third layer generates a probabaility distribution over the five emotion classes. It also uses the ReLU as the actiation function.

We used a RNN model, a deep learning model it consist of 3 bidirectional Recurrent Neural Network (RNN) layers followed by a fully connected layer for output generation. Each RNN layer incorporates dropout regularization to avoid overfitting.The model is designed to handle sequential data, learning features across multiple RNN layers to generate accurate predictions.

We used LSTM Model. The model consists of an LSTM layer followed by a fully connected layer. The LSTM layer is designed to handle sequential inputs, with configurable parameters for input size, number of layers, and a dropout rate of 0.2 to reduce overfitting. The output of the LSTM is passed through the fully connected layer to generate predictions corresponding to emotion classes.

We used a SVC (Support Vector Machine) model, a machine learning model that employs a linear kernel to classify data by finding the optimal hyperplane that separates classes. The model was trained using X_train and y_train, and predictions were made on X_test. It uses the hyperparameter C=1 to balance the margin and misclassification.

We used a RandomForestClassifier, an ensemble learning method that builds multiple decision trees and combines their outputs for classification. The model was trained with 100 estimators and a fixed random state for reproducibility. It leverages the power of multiple trees to improve accuracy and reduce the risk of overfitting.

We used a ResNet9 model, a deep learning architecture that incorporates residual connections to enhance training and mitigate vanishing gradient issues. The model consists of initial convolutional layers followed by three residual blocks, each containing two convolutional layers. These residual blocks allow the network to learn identity mappings, making it easier to train deeper networks. After applying max pooling and dropout for regularization, the output is flattened and passed through a fully connected layer for classification.

1.2 Justification for the Proposed Work:

The selection of the best-performing model was based on extensive experimentation with ten architectures, highlighting a data-driven approach to model selection. Through thourough evaluation process ,we ensured final model's reliability and scalability, addressing gaps such as noise sensitivity, overfitting and misclassification in existing solutions.

As in the below bar graph in contrast to other models the proposed model demonstrated exceptional performance, achieving the highest accuracy (98.04%) among all tested architectures.

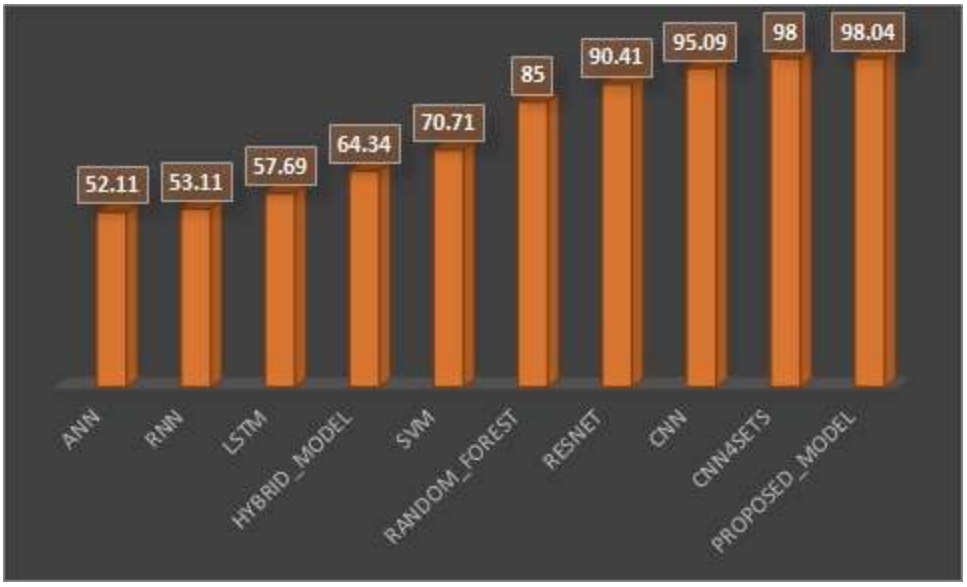

Fig1: Accuracy Graph

CHAPTER 3

PROPOSED WORK , ARCHITECTURE,TECHNOLOGY STACK & IMPEMENTATION DETAILS

3.1 Proposed Work

The primary objective of this proposed work is to enhance the performance of Transformer model in the domain of Speech Emotion Recognition (SER). The focus is on refining the model architecture to improve feature extraction, reduce overfitting, and enable better generalization across various speech emotions. The proposed enhancements aim to address the limitations of the existing model while maintaining both efficiency and scalability. A more robust and efficient model is essential to accurately capture the intricate patterns within speech data, enabling the model to deliver higher classification accuracy.

Although the Transformer's current architecture has proven to be effective, several enhancements can be implemented to better handle the temporal and contextual complexities of speech signals. The following proposed improvements aim to strengthen the model's ability to capture speech characteristics and deliver more precise emotion classifications:

1. **Refined Feature Embedding:**
   The existing input pipeline leverages features such as MFCCs, chroma, and Mel spectrograms. By incorporating additional features the Transformer can enhance its attention mechanism to capture more nuanced and emotionally relevant speech characteristics.

2. **Mitigating Noise and ASR Dependency:**
   The performance of the Transformer model can degrade due to errors in preprocessing or noisy environments. By employing robust preprocessing techniques and noise reduction methods, the model can be made more resilient to such challenges.

3. **Regularization and Overfitting Mitigation:**
   Overfitting remains a significant challenge in deep learning models, including Transformers. Techniques such as dropout within the attention and feed-forward layers, L2 regularization, and data augmentation (e.g., time-stretching, pitch-shifting, and noise addition) can improve generalization. This ensures the model is trained on a diverse set of speech data and performs well on unseen examples.

4. **Model Evaluation and Optimization:**
   Evaluation metrics like accuracy, precision, recall, and F1-score will be meticulously tracked during training. Through extensive cross-validation and hyperparameter tuning using approaches like grid search or Bayesian optimization, the Transformer model will be optimized to achieve a balance between performance and computational demands.

5. **Scalability and Real-Time Processing:**
   Ensuring the model is scalable and efficient for large datasets and real-time applications is critical. Techniques like attention head pruning, model quantization, and memory-efficient Transformer architectures will be explored to optimize inference speed and reduce resource consumption while maintaining performance.

In conclusion, by implementing these enhancements, the Transformer model can evolve into a more robust and efficient system for recognizing and classifying speech emotions. These improvements address current limitations and pave the way for scalable and real-world-ready applications.

3.2 Technology Stack

•**Programming Language:** Python serves as the backbone to our project, chosen for it's extensive support of libraries and tools that are specifically designed for machine learning tasks. With it's rich set of libraries for model training, evaluation, and many more Python is an ideal language for implementing and scaling machine learning solutions efficiently.

• **Libraries and Frameworks**

1. **PyTorch**:

   PyTorch is the core framework used in our project, powering the design and implementation of the Transformer model, as well as the creation of DataLoaders and the execution of training and inference pipelines. The optimization of the model is performed using the Adam optimizer, while loss computation during training is handled by CrossEntropyLoss, a commonly used loss function for classification tasks.

2. **Librosa**:

   Audio signal processing plays a crucial role in feature extraction, and Librosa is the popular library for this purpose. It allows for efficient extraction of Mel-frequency cepstral coefficients (MFCCs), chroma features, and Mel spectrograms, all of which are essential for training

and inference. These features form the input to the neural network, enabling it to learn from the audio data effectively.

3. **NumPy**:

   NumPy is the fundamental library for numerical computations in Python. It has been heavily used in preprocessing and feature extraction stages to manipulate data, perform mathematical operations, and handle arrays efficiently. The integration of NumPy ensures that the data pipeline remains performant and optimized for machine learning tasks.

4. **Pandas**:

   Pandas is the primary library used for data management and preprocessing. It provides powerful tools for handling tabular data, making it easier to clean, filter, and manipulate datasets before they are fed into machine learning models. The use of DataFrames allows for efficient handling of large datasets in a structured format.

5. **Scikit-learn**:

   Scikit-learn has been utilized for various preprocessing tasks, such as splitting the dataset into training and testing sets, as well as encoding target labels. Additionally, it provides robust methods for evaluating the performance of machine learning models, including the calculation of metrics like accuracy, precision, recall, and F1-score.

6. **Matplotlib/Seaborn**:

   For visualizing data and results, Matplotlib and Seaborn have been used for creating informative and insightful plots. These include visualizations of training trends, such as loss and accuracy curves over

epochs, as well as plots like the confusion matrix to assess model performance in terms of true and false classifications.

3.3 Implementation Details

Load Dataset: We analyzed a total of **783 audio files** after applying data augmentation techniques. These audio files are distributed across **6 emotion classes**: happy, fear, sad, angry, disgust, and neutral. The original dataset consisted of **269 audio files**, but through data augmentation, which included adding **white noise** and applying **spectral shifts**, the number of files was increased to **783**.

The below table gives data processing summary

Step	Details
Original Dataset Size	269 audio files
Augmented Dataset Size	783 audio files (after noise addition and spectral shifting)
Feature Types	MFCC, Chroma, Mel Spectrogram
Train-Test Split	- **Training Set**: 626 samples (80%) - **Testing Set**: 157 samples (20%)
Final Dataset Dimensions	- **x_train**: 626 samples - **x_test**: 157 samples

Table1: Data Processing summary

To implement this project we used telugu emotion dataset saved inside 'emotions_dataset' folder and below screenshot shows various classes of emotions that have been used in our project.

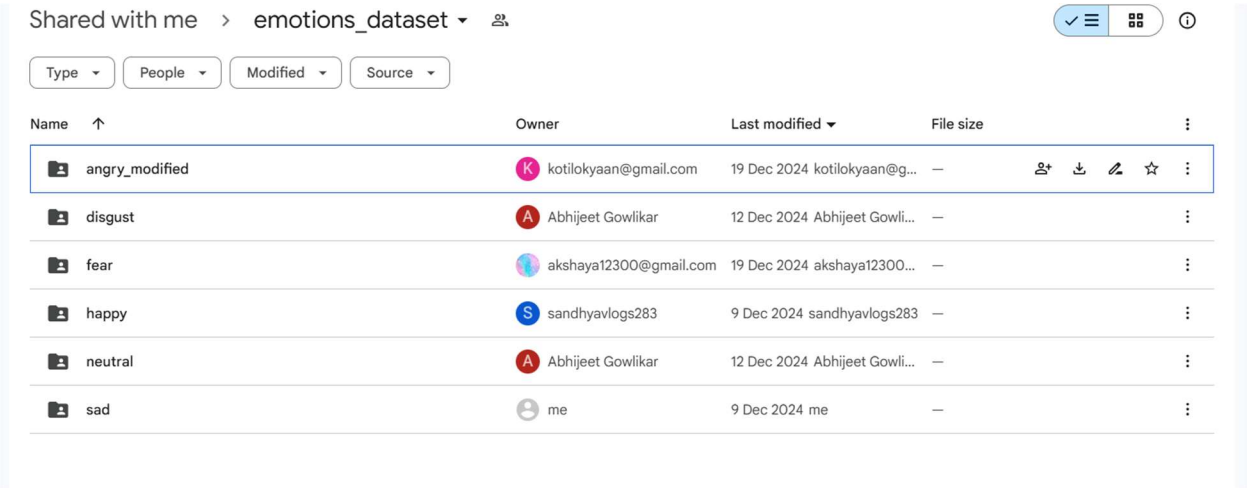

Fig2: Emotion classes in Dataset

1. Model Architecture

The Transformer-based Model has been built to process sequential data, such as feature sequences derived from audio signals, for tasks like Speech Emotion Recognition (SER). The architecture follows a structured approach that leverages self-attention mechanisms to capture global dependencies. Below are the key components:

Transformer Encoder Layers

- **Embedding Layer:**
  Converts the input feature vectors into a higher-dimensional space, preparing them for processing by the Transformer layers.
    - Input Dimension: Matches the input feature size.
    - Output Dimension: 256-dimensional vector space, ensuring a richer representation of the data.

- **Multi-Head Self-Attention:**
  Captures relationships between all positions in the sequence, allowing the model to focus on relevant parts of the sequence.
    - Number of Heads: 4 heads to ensure diverse representation learning.
- **Feed Forward Network:**
  Each Transformer encoder layer includes a feedforward network for nonlinear transformations of the attention outputs.
    - Layer Structure: Composed of two linear transformations separated by a ReLU activation.
- **Stacked Transformer Layers:**
  Two layers of Transformer encoders refine the feature representations.
    - Hierarchical Representation: Helps the model learn complex relationships within the data.

Classifier

Maps the learned sequence representation to the emotion classes.

- Output Dimension: Equal to the number of emotion classes (6).

2. Data Preparation

The feature extraction process is essential to convert raw audio data into a form that the model can process. Several audio features are extracted and preprocessed to capture different aspects of the audio signal:

- **MFCC (Mel-Frequency Cepstral Coefficients):**
    - **Purpose:** MFCCs capture the short-term power spectrum of sound, which is highly effective for speech and emotion recognition tasks.

- **Extraction:** The model extracts 40 MFCC coefficients from the audio and computes the mean of each coefficient across the time frames.

- **Chroma Features:**
  - **Purpose:** Chroma features reflect the intensity of the 12 pitch classes and are often used in music analysis and audio classification tasks.
  - **Extraction:** These features are computed using the short-time Fourier transform (STFT), which represents the frequency content of the audio over time.

- **Mel Spectrogram:**
  - **Purpose:** The Mel spectrogram captures the frequency content of the audio signal using the Mel scale, which aligns more closely with human perception of pitch.
  - **Extraction:** The mean of the Mel spectrogram is calculated to reduce the dimensionality and ensure a fixed-size representation for each audio file.

- **Feature Concatenation:**
  - The MFCC, chroma, and Mel spectrogram features are concatenated into a single feature vector, which serves as the input to the convolutional layers.

- **Reshaping Data:**
  - The feature vectors are reshaped to match the model's expected input format: (batch_size, channels, sequence_length).

3. Data Augmentation

To improve the robustness and generalization of the model, data augmentation techniques are applied to the audio data:

- **Noise Addition:**
    - **Function:** noise(data, noise_factor)
    - **Description:** Random Gaussian noise is added to the audio signal, simulating real-world conditions where background noise can affect the quality of recordings. The amount of noise is controlled by the noise_factor parameter.
- **Time Shifting:**
    - **Function:** shift(data, sampling_rate, shift_max, shift_direction)
    - **Description:** This technique shifts the audio in time, either left or right, which helps the model become more invariant to slight variations in timing. The maximum shift amount is controlled by the shift_max parameter, and the direction of the shift is specified by the shift_direction.

4. Data Loading

- **Function:** load_data(save=False)
    - **Description:** This function traverses the directories containing audio files, extracts the relevant features (MFCC, chroma, Mel spectrograms), applies the necessary augmentations, and prepares the dataset for training.
    - The function ensures that the features are stored in a format compatible with the model, either in memory or saved to disk for later use.

5. Train-Test Split

- **Function:** train_test_split from scikit-learn
    - **Purpose:** The dataset is divided into training and testing subsets to evaluate the model's performance. 80% of the data is allocated to the training set, and 20% is reserved for testing. This split ensures that the model is trained on a large enough dataset while being evaluated on an independent set to check for overfitting.

6. Evaluation Metrics

The model's performance is assessed using several standard metrics:

- **Accuracy:** The proportion of correct predictions compared to the total number of predictions. It is calculated as the number of correct predictions divided by the total number of samples.
- **Loss:** The loss function (cross-entropy loss in this case) measures the error between the predicted class probabilities and the actual class labels. The model aims to minimize this loss during training.
- **Confusion Matrix:** A confusion matrix provides a detailed breakdown of the model's predictions, showing the number of true positives, false positives, true negatives, and false negatives for each emotion class. This matrix is useful for identifying which classes the model performs well on and which ones it struggles with.

7. Model Training

- **Training Loop:** The model undergoes training for a specified number of epochs (e.g., 50 epochs). In each epoch:

- The model's parameters are updated based on the loss computed from the training data.
- After every epoch, the model is evaluated on the validation set to track performance.
- The train_loss, train_accuracy, val_loss, and val_accuracy metrics are logged to monitor the training progress.

- **Optimization:** The model uses the Adam optimizer with a learning rate of 0.001 and a weight decay of 1e-6 to optimize the parameters. This optimizer is well-suited for tasks with sparse gradients, like the ones encountered in deep learning models.
- **Evaluation:** The model's performance on the validation set is evaluated after every epoch, and the training and validation metrics are printed to track the progress.

8. Inference Pipeline

- **Prediction Function:** A separate inference pipeline is provided to make predictions on unseen audio data. The process involves:
  - **Audio Loading:** The audio file is loaded, and features are extracted.
  - **Feature Transformation:** The extracted features are reshaped to match the model's expected input format.
  - **Model Evaluation:** The model is set to evaluation mode, and the features are passed through the model to obtain predictions.
  - **Class Mapping:** The predicted class is mapped to the corresponding emotion label using a label encoder.

Model Architecture

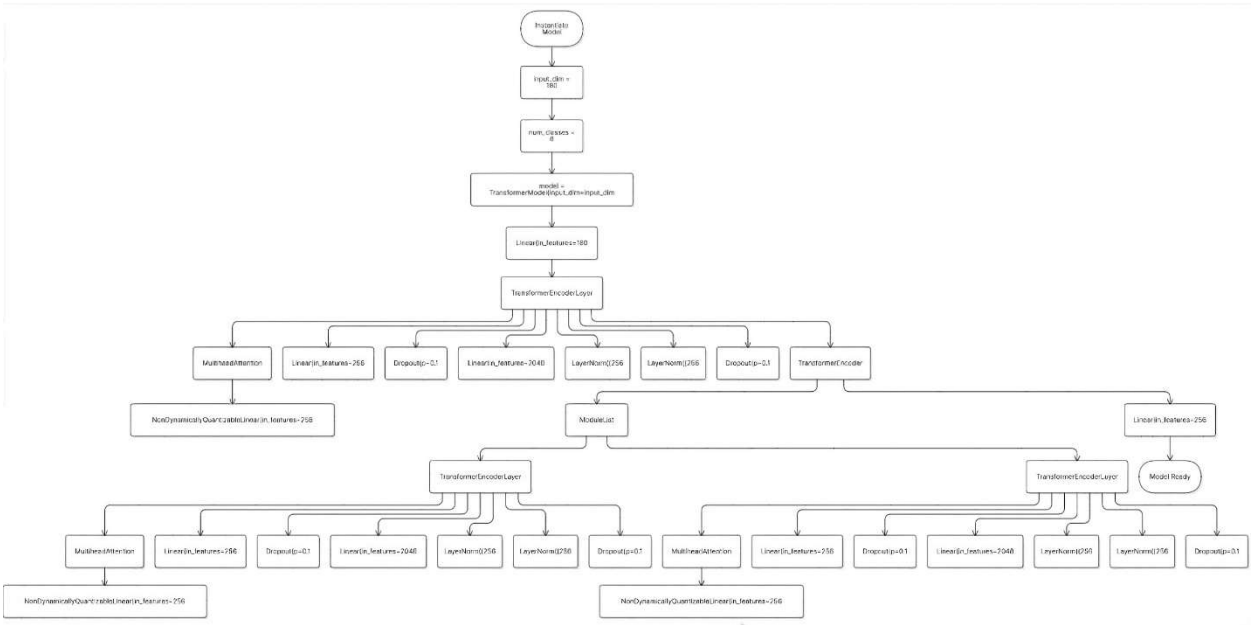

Fig3: Transformer Model Architecture

The above diagram represents the architecture of a Transformer-based model designed for emotion recognition tasks. This model takes input features of dimension input_dim and processes them to predict num_classes. It begins with a linear embedding layer, which maps the input features into a 256-dimensional space to prepare them for the Transformer. The core of the architecture consists of two TransformerEncoder layers, each comprising submodules such as MultiHeadAttention for capturing dependencies between different parts of the input sequence, and FeedForward networks (Linear layers) for feature transformation. Each encoder layer incorporates essential components like layer normalization to stabilize training, dropout for regularization, and residual connections to facilitate gradient flow. These elements work together to extract hierarchical and contextual information from the input sequence. The output of the final Transformer layer is passed through a fully connected linear layer that maps it to the six output classes.

Chapter 4

Results and Discussions

The Transformer Speech Emotion Recognition model demonstrates strong performance across all metrics, achieving high accuracy in emotion classification.

The screenshot of confusion matrix highlights minimal misclassifications, with particularly strong performance in predicting classes like "angry," "disgust," and "happy," where there are no false positives or negatives.

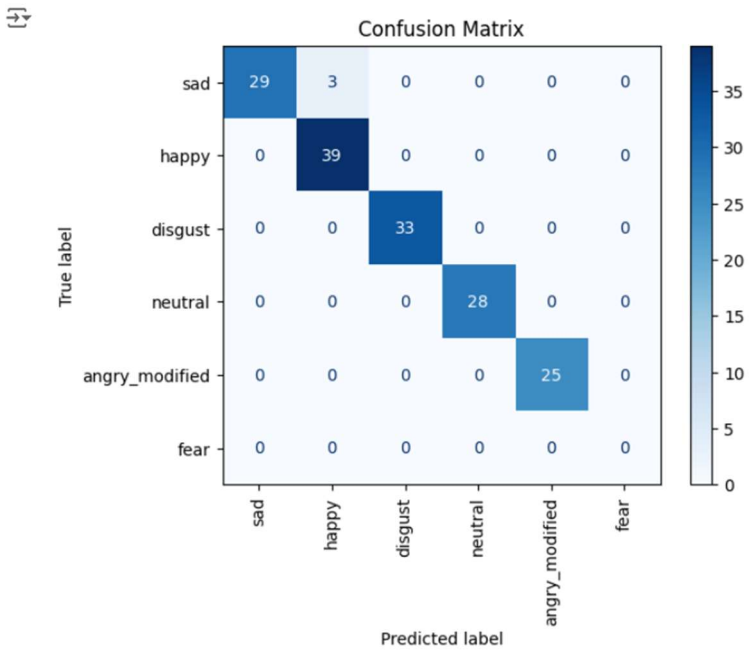

Fig4: Confusion Matrix

The training and validation accuracy curves as in below screenshot demonstrate the transformer model's effective learning and convergence, with both curves reaching high accuracy over 50 epochs, indicating minimal overfitting and consistent performance.

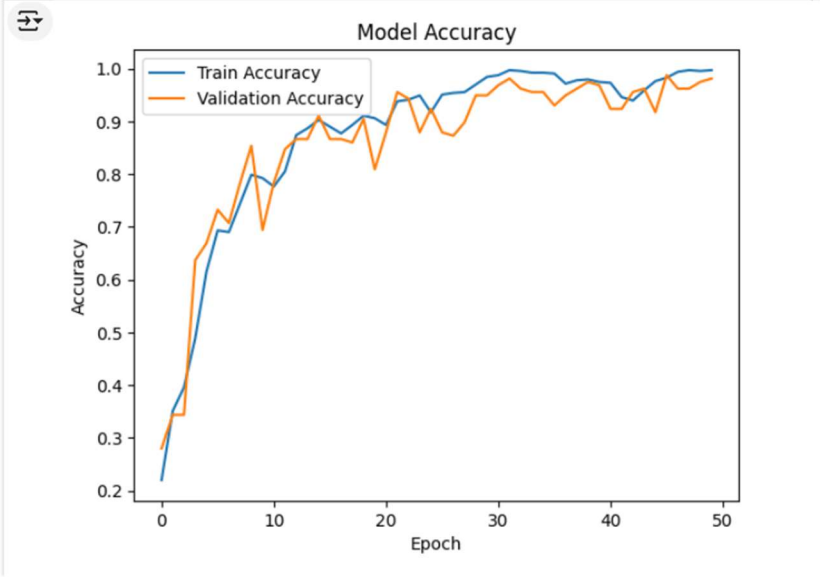

Fig5: Train and Validation Accuracy Curve

The architecture, consisting of Transformer encoder layers, multi-head self-attention mechanisms, and feedforward networks, ensures the model's robustness and ability to capture global dependencies in sequential data. The classification results confirm that the model maintains high precision, recall, and F1-scores across all emotion classes, making it a reliable tool for practical applications in SER tasks. This consistency across metrics and class distributions establishes the model as an effective and dependable solution for emotion detection in audio data.

```
Number of predictions: 157
Number of true labels: 157
Sample predictions: [3 2 1 4 2 4 2 3 1 0]
Sample true labels: [3 2 1 4 2 4 2 3 1 0]
Confusion Matrix:
[[29  3  0  0  0]
 [ 0 39  0  0  0]
 [ 0  0 33  0  0]
 [ 0  0  0 28  0]
 [ 0  0  0  0 25]]
Precision: 0.9823
Recall: 0.9809
F1 Score: 0.9808
```

Fig6: Precision Recall and F1 score

4.2 Output Screens

Our application NeuraWave provides an intuitive and user-friendly interface for users to interact with the application. It includes a main page, a sign-in page, and a sign-up page.

This is the **home page** of our UI where a user can login or signup to use our services

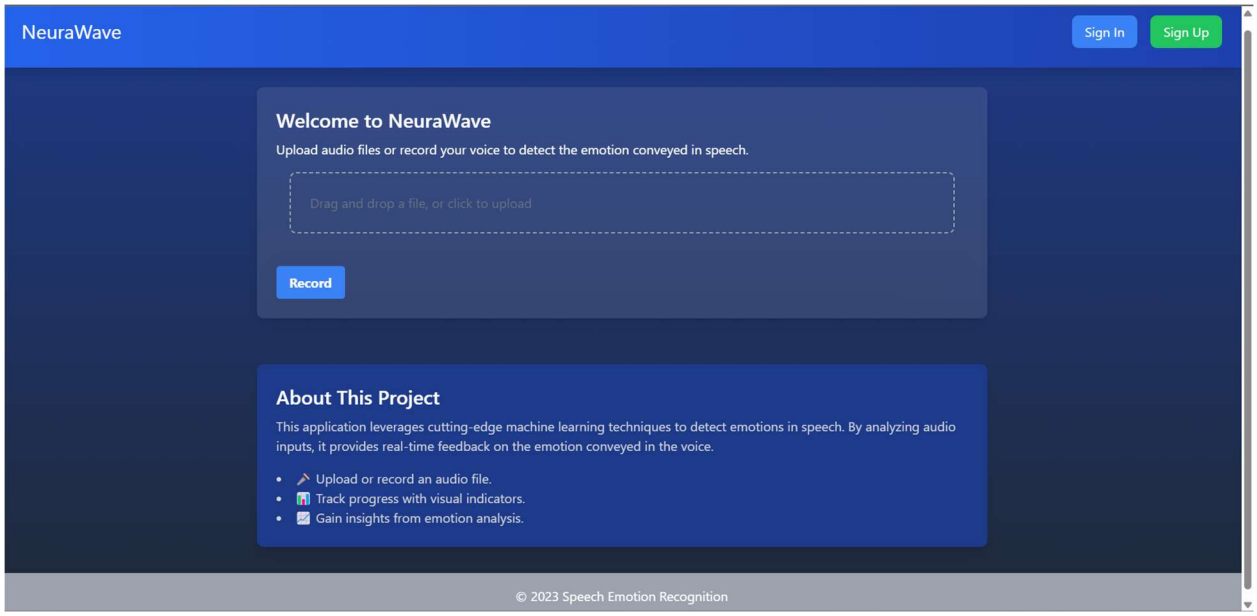

Fig7: UI HomePage

Sign Up Page:

When a user clicks on signup they will get navigate to this page where user has to provide information like name, email, password

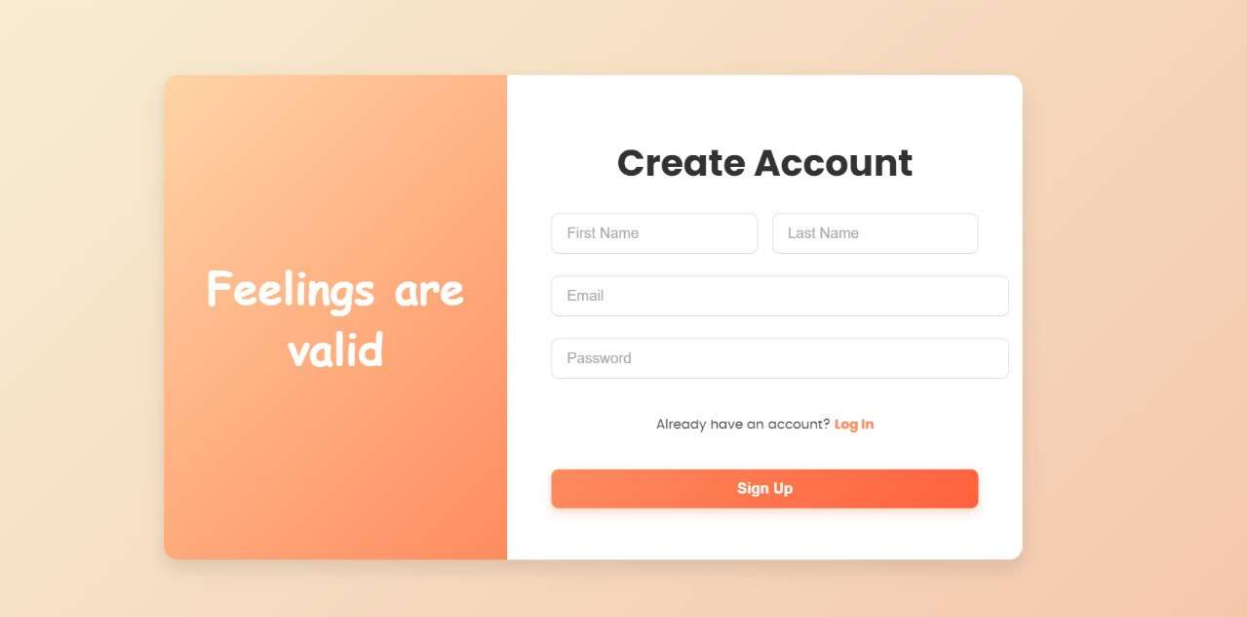

Fig8: SignUp Page

Sign In page:

After signup user has to login by providing the same email and password which the user provided while signup.

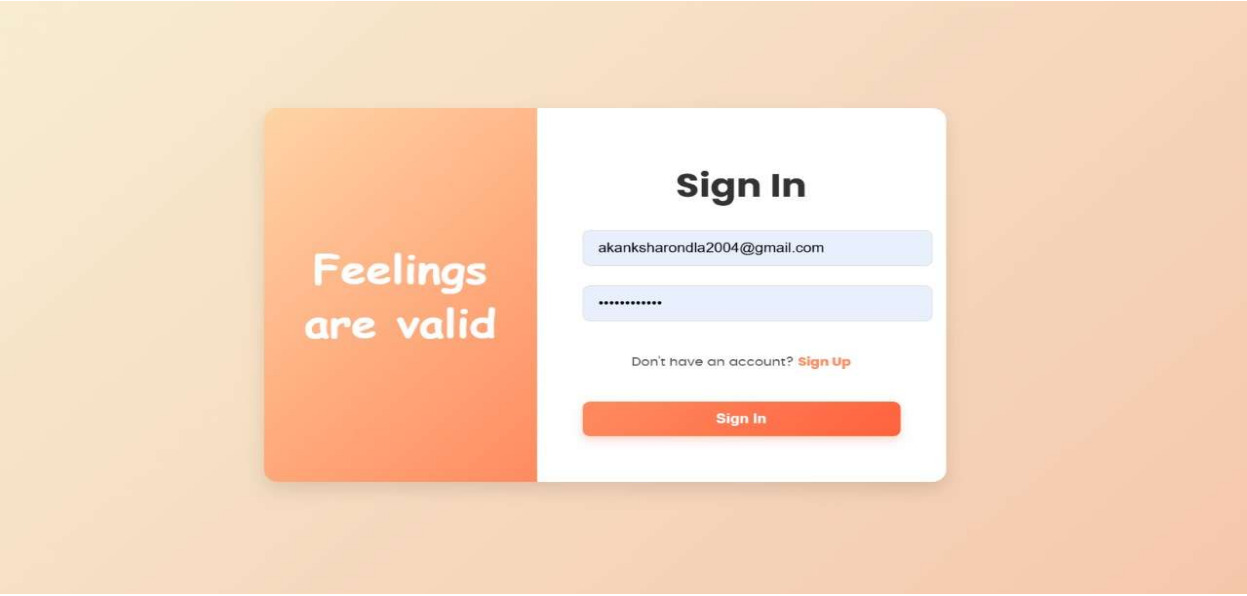

Fig9: SignIn Page

After signing in user can either drag and drop audio files or upload files from their device. User can also choose to record their voice. Upon the file being successfully uploaded, we give the predicted emotion.

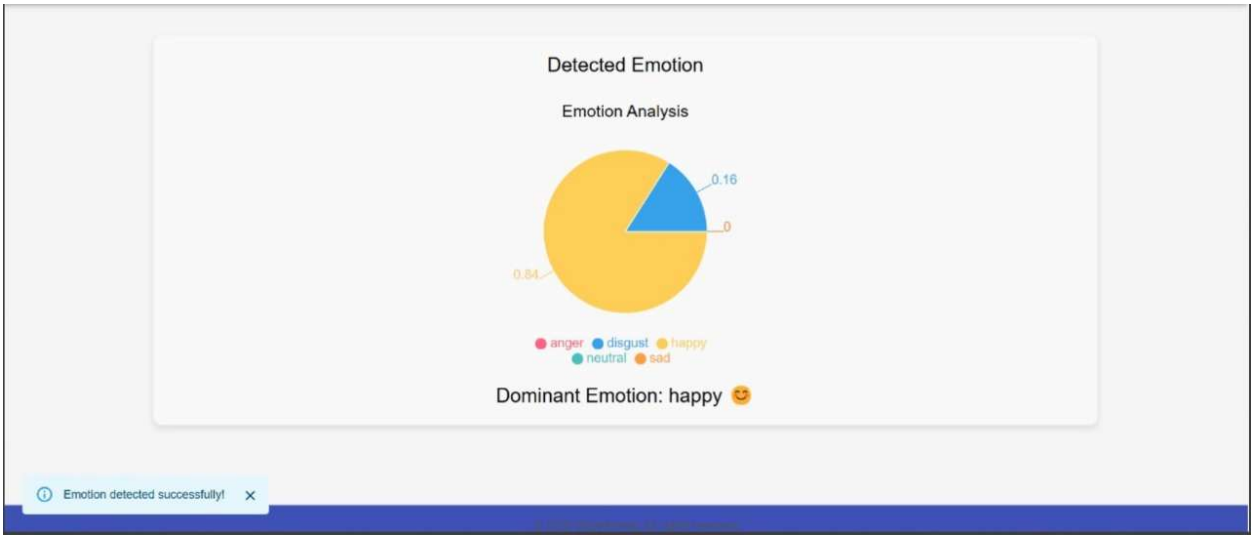

Fig10: Emotion Detection Output

Chapter 5

Conclusion & Future Scope

In this proposed work, we aim to enhance the performance of the Transformer model in the domain of Speech Emotion Recognition (SER) by addressing key limitations of the current architecture. By refining the feature extraction process, implementing model architecture improvements, reducing overfitting, and optimizing for better generalization, we anticipate significant improvements in the model's ability to accurately classify emotions from speech data. These enhancements will provide the Transformer model with a more robust and scalable framework to handle the intricate patterns of speech, ensuring better classification performance and generalization across different speech emotions.

Through these proposed changes, the Transformer model will become more efficient and adaptable, enabling it to be deployed in real-world applications that require accurate and real-time emotion recognition. The improvements outlined in this work not only aim to enhance model performance but also ensure that it remains computationally feasible and scalable for large-scale datasets and practical use cases.

Future Scope:

While the current Speech Emotion Recognition (SER) model provides a strong foundation, several areas remain for further exploration and enhancement.

Incorporating datasets from different languages will enable the model to recognize emotions in multilingual contexts. This would be crucial for applications in global markets and diverse user bases.

Combining speech data with facial expression recognition could improve emotion classification accuracy. This multi-modal approach would allow the model to utilize visual cues (from videos or images) along with audio features, offering more reliable emotion recognition in real-world settings.

Model architecture can be further improved by using hybrid models like combination of Transformers for feature extraction with sequential modeling using RNNs or CNNs.

While six basic emotions can be effective for general applications, a larger set of emotion categories could improve **emotion granularity**. Recognizing emotional nuances, such as distinguishing between **surprise** and **confusion**, or between **happiness** and **pride**, can provide more accurate insights into a person's emotional state, enhancing the performance of **human-computer interaction** and **mental health monitoring** systems.

The current system could be adapted to work in real-time applications such as virtual assistants (e.g., Siri, Alexa), mental health monitoring tools, or customer service chatbots. Efficient inference pipelines, optimized for lower latency, could allow the model to predict emotions during ongoing conversations.

CHAPTER 5

References:

1. **Han K, Yu D, and Tashev**. This reference discusses speech emotion recognition using deep neural networks (DNNs) and extreme learning machines, aligning well with your CNN and ANN focus.
2. **Lin Zhen-Tao et al.** Explores SER based on formant characteristics and phoneme type convergence, which may provide insights into feature extraction techniques relevant to CNN and Transformer-based models.
3. **Farooq Misbah et al.** Focuses on the impact of feature selection algorithms in SER using deep convolutional neural networks, directly relevant to CNNs.
4. **Vaswani A, Shazeer N, Parmar N, et al.** This reference introduces the Transformer model ("Attention is All You Need"), critical for applying Transformers in SER tasks.
5. **Issa D, Demir M F, and Yasici A.** Focuses specifically on speech emotion recognition using deep convolutional neural networks, directly relevant tANN:52.11