ABSTRACT

This project develops a deep learning-based emotion recognition system that analyses spoken audio dialogues to classify emotions, enhancing human-computer interaction. By utilizing a hybrid architecture combining Convolutional Neural Networks (CNNs) for feature extraction and Long Short-Term Memory (LSTM) networks for temporal analysis, the system addresses the limitations of traditional methods. Employing the CREMA-D dataset, which features diverse emotional expressions, the model achieves an accuracy of 99.22%.

Performance metrics such as precision, recall, and F1-score also demonstrated the model's effectiveness. This project successfully developed a robust emotion recognition system capable of accurately classifying emotions from audio dialogues. The results highlight the potential for real-world applications, paving the way for more empathetic and responsive systems in human-computer interaction.

ACKNOWLEDGEMENT

The satisfaction and the euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. The constant guidance of these persons and encouragement provides, crowned our efforts with success and glory. Although it is not possible to thank all the members who helped for the completion of the Mini project individually, we take this opportunity to express our gratitude to one and all.

We are grateful to management and our institute **GLOBAL ACADEMY OF TECHNOLOGY** with its very ideals and inspiration for having provided us with the facilities, which made this Mini project a success.

We express our sincere gratitude to **Dr. N. Rana Pratap Reddy,** Principal, Global Academy of Technology for the support and encouragement.

We wish to place on record, our grateful thanks to **Dr. Kumaraswamy S**, Professor and Head, Department of CSE, Global Academy of Technology, for the constant encouragement provided to us.

We are indebted with a deep sense of gratitude for the constant inspiration, encouragement, timely guidance and valid suggestion given to us by our guide **Prof. Ravindranath R C, Assistant Professor,** Department of CSE, Global Academy of Technology.

We are thankful to all the staff members of the department for providing relevant information and helped in different capacities in carrying out this project.

Last, but not least, we owe our debts to our parents, friends and also those who directly or indirectly have helped us to make the project work a success.

RAKSHITH H R **1GA21CS117**
RAKSHITHA L HEGDE **1GA21CS119**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ML	Machine Learning
AI	Artificial Intelligence
CREMA-D	Crowd Sourced Emotional Multimodal Actors Dataset (CREMA-D) is an audio-visual data set for emotion recognition.
Emotion Recognition	The process of identifying human emotion
Spectrogram	A visual representation of the spectrum of frequencies of a signal as it varies with time
MFCC	Mel Frequency Cepstral Coefficients, commonly used feature in speech and audio processing
Sampling Rate	The number of samples of audio carried per second, measured in Hz
WAV	Waveform Audio File Format, an audio file format standard
ANG	Anger emotion
DIS	Disgusting emotion
FEA	Fear emotion
HAP	Happy emotion
NEU	Neutral emotion
SAD	Sad emotion

CHAPTER 1

INTRODUCTION

1.1 Introduction to Project

The project titled "Emotion Recognition from Audio Dialogues Using Deep Learning" aims to develop an advanced system that can accurately detect and classify emotions from spoken audio using deep learning techniques. Emotion recognition plays a crucial role in enhancing human-computer interaction, particularly in applications such as virtual assistants, automated customer service, and interactive voice response systems. By understanding the emotional state of the user, these systems can provide more personalized and effective responses.

1.2 Problem Definition

Emotion recognition from audio dialogues is a challenging task due to the inherent variability in human speech, including different accents, speaking speeds, intonations, and background noise. Existing solutions often struggle with these complexities, leading to inaccurate or inconsistent results. These challenges necessitate the use of advanced deep learning models that can learn complex patterns and representations from raw audio data, making the system more adaptable and accurate in diverse conditions.

1.3 Existing System

Traditional emotion recognition systems primarily rely on handcrafted features and basic machine learning models like Support Vector Machines (SVMs) or Random Forests. These systems, while functional, are limited in their ability to capture complex emotional nuances present in audio data.

1.4 Proposed System

The proposed system addresses the limitations of existing methods by employing a deep learning approach. Specifically, it utilizes a hybrid architecture combining Convolutional Neural Networks (CNNs) for automatic feature extraction and Long Short-Term Memory (LSTM) networks for capturing temporal dependencies in the audio data. This

deep learning-based system is designed to improve the accuracy of emotion recognition across varied and noisy audio inputs, making it more suitable for real-world applications.

1.5 Objectives of the Project Work

- To develop a deep learning model that accurately classifies emotions from audio dialogues.
- To enhance the model's ability to handle diverse speech patterns and background noise.
- To optimize the model for real-time processing in practical applications.
- To evaluate the model's performance on a standardized dataset and benchmark it against existing methods.

1.6 Scope of the Project Work

The project is focused on the development of a deep learning-based emotion recognition system for audio dialogues. The system is designed to be scalable and adaptable to various use cases, including virtual assistants, call centers, and mental health monitoring. Future enhancements may involve integrating the system into real-time applications and expanding its capabilities to recognize a wider range of emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad).

1.7 Project Report Outline

Chapter 1: Introduction

Chapter 2: Literature Survey

Chapter 3: System Analysis

Chapter 4: System Design

Chapter 5: System Implementation

Chapter 6: Testing and Results

Chapter 7: Conclusion and Future Work

CHAPTER 2

LITERATURE SURVEY

2.1 System Study

The study of emotion recognition systems highlights the evolution from simple rule-based methods to sophisticated deep learning algorithms. Earlier systems were limited in their ability to recognize emotions due to their reliance on predefined features. In contrast, deep learning models have demonstrated significant improvements in accuracy and robustness, thanks to their ability to learn complex representations directly from data.

The studies conducted in referred papers focus on the field of emotion recognition using speech. The CREMA-D dataset serve as the primary source of audio data for these studies. The studies leverage the CREMA-D dataset, which consists of 7,442 audio files from 91 actors expressing six emotions (anger, disgust, fear, happy, neutral, and sad). The actors in this dataset are diverse, representing various ages and ethnicities, which provides a good testbed for generalization of emotion recognition models.

- **Feature Extraction**:
  - Both studies focus on extracting Mel Frequency Cepstral Coefficients (MFCCs) as the primary features for emotion recognition. MFCCs are a popular feature in speech processing because they represent the short-term power spectrum of sound, capturing the nuances of human speech that are crucial for identifying emotions.

- **Models Used**:
  - Deep Learning Models: The studies apply various Deep Learning models to classify emotions based on the extracted MFCC features. Specifically, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are utilized for their ability to handle spatial and sequential data, respectively.
  - CNN: Employed for its effectiveness in recognizing patterns in data, CNNs are applied to the MFCCs, treating them as image-like inputs.
  - RNN: Particularly suited for sequential data, RNNs are used to capture the temporal dynamics of speech that are essential for emotion recognition.

- **Performance Metrics**:
  - The performance of these models is evaluated using accuracy as the primary metric. The studies report the effectiveness of various models in accurately classifying emotions from the speech data, with CNNs and RNNs being the top performers.

2.2 Review of Literature

- Traditional Machine Learning Approaches: Early emotion recognition methods relied on features like Mel-Frequency Cepstral Coefficients (MFCCs) and prosodic features (e.g., pitch, energy), combined with classifiers like SVMs and HMMs. These methods were somewhat effective but struggled with generalization and required extensive manual feature engineering.
- Deep Learning Approaches: The introduction of deep learning, particularly CNNs and RNNs, has transformed the field of emotion recognition. CNNs are effective in extracting spatial hierarchies from raw audio data, while LSTMs are well-suited for modelling the temporal dynamics of speech. These models have shown superior performance in capturing the nuances of emotional expression in audio data.
- **MFCCs in Emotion Recognition**:
  - Several studies have established MFCCs as the standard for feature extraction in audio processing. MFCCs are particularly effective because they approximate the human ear's response to different frequencies, making them suitable for identifying the emotional content in speech.
- **Previous Research**:
  - **Beard et al.**: Achieved 41.5% accuracy on the CREMA-D dataset using an LSTM model. Their work highlights the challenges of emotion recognition, particularly in diverse datasets.
  - **Jin et al.**: Reported 51.82% accuracy with Support Vector Machine (SVM) classifiers on a similar dataset, demonstrating the potential of traditional machine learning techniques combined with robust feature extraction.
  - **Other Studies**: Various other research efforts have experimented with different model architectures and datasets, consistently finding that combining robust feature extraction with advanced models (like CNNs and RNNs) yields the best results.

- **Gap in Research**:
  - The review identifies a gap in the application of these techniques to more generalized or diverse datasets. Most studies focus on a single dataset or a narrow set of emotions, limiting the generalizability of the findings. This gap highlights the need for more comprehensive studies that can validate the effectiveness of these models across different datasets and emotional contexts.

2.3 Comparison of Literature

A comparison of the literature, summarizing the methodologies, datasets, and accuracy rates of various studies, are as follows:

Table 2.1 Comparison of Literature

Study	Dataset	Features	Model	Accuracy
Beard et al.	CREMA-D	MFCCs	LSTM	41.5%
Jin et al.	CREMA-D	MFCCs	SVM	51.82%
[1]	CREMA-D	MFCCs	CNN/RNN	Reported as top performers
[2]	CREMA-D	MFCCs	CNN/RNN	Best performing models
[3]	CREMA-D	MFCCs	CNN- LSTM	90%
[4]	CREMA-D	MFCCs	CNN	64%
[5]	CREMA-D	MFCCs	Bi-LSTM	Best performing models

This comparison highlights the evolution of methodologies in the field, showing a trend toward the use of Deep Learning models, which have demonstrated improved accuracy over traditional machine learning techniques.

Chapter 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 Functional Requirements

- The system should be able to process raw audio data and output the predicted emotion.
- It should support real-time emotion detection for practical applications.
- The model should handle different accents, languages, and speech patterns without significant loss in accuracy.

3.2 Non-Functional Requirements

- The system should be scalable to handle large datasets.
- It should be robust to background noise and varying audio quality.
- The processing time should be minimized to ensure real-time performance.

3.3 Hardware Requirements

- High-performance computing resources, including a GPU for training the deep learning models.

3.4 Software Requirements

- Python, TensorFlow/PyTorch, librosa for audio processing, and scikit-learn for evaluation metrics.

Chapter 4

SYSTEM DESIGN

4.1 Design Overview

The system design involves multiple components working together to achieve emotion recognition from audio dialogues. The key components include data preprocessing, feature extraction, model training, and emotion classification.

4.2 System Architecture

- Data Preprocessing: The raw audio data is first cleaned and normalized. Noise reduction techniques are applied, and the audio is segmented into frames.
- Feature Extraction: Using CNNs, features are automatically extracted from the audio frames. The CNN layers capture spatial hierarchies in the data.
- Emotion Classification: LSTM layers are applied to capture the temporal dynamics of the extracted features. The final output layer provides the emotion class prediction.

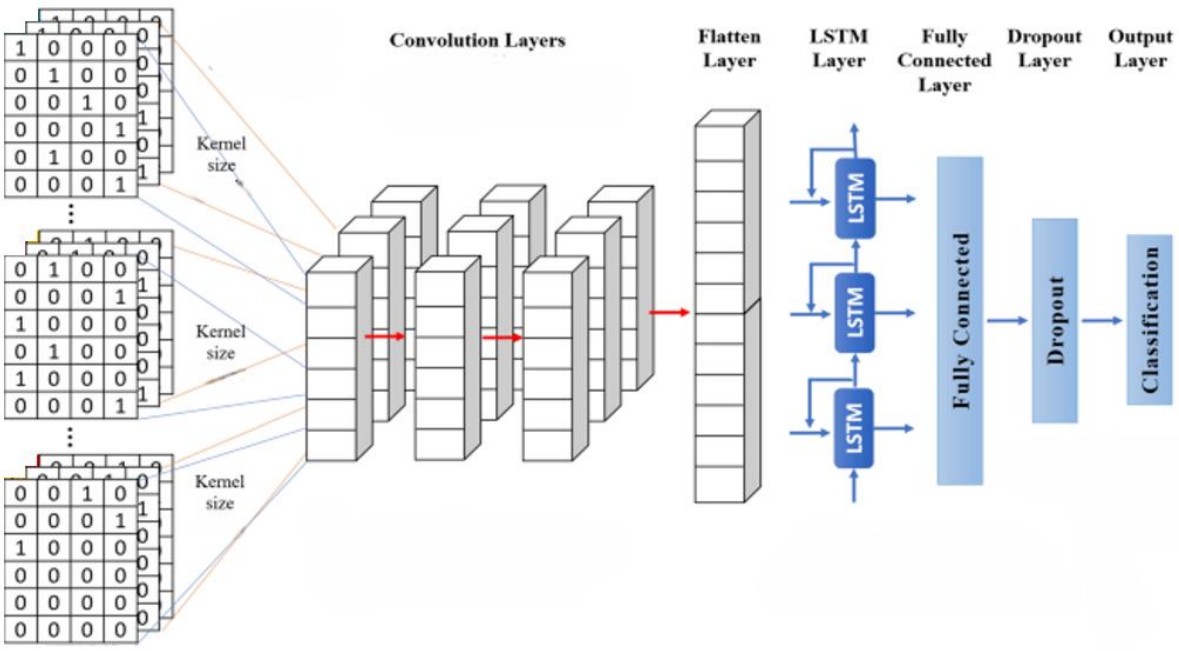

Fig 4.1 System Architecture

4.3 Data Flow Diagrams

The overall flow starts with the user inputting audio recordings, leading to preprocessed features that feed into the model, resulting in emotion predictions that are output back to the user while also being stored for future analysis. This structured approach enables the system to accurately classify various emotions, making it a valuable tool for applications in sentiment analysis and human-computer interaction.

4.3.1 Data Flow Diagram - Level 0

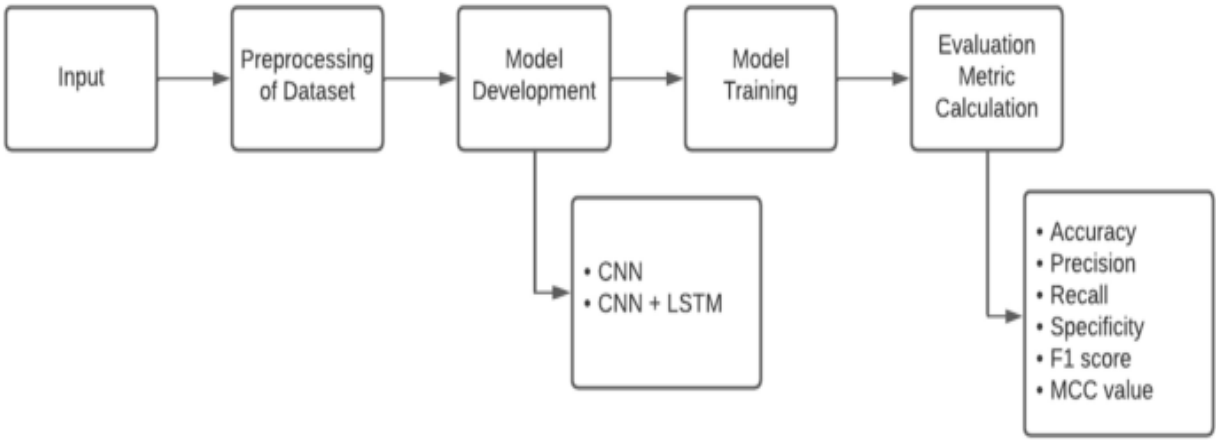

Fig 4.2 DFD-Level 0

4.3.2 Data Flow Diagram - Level 1

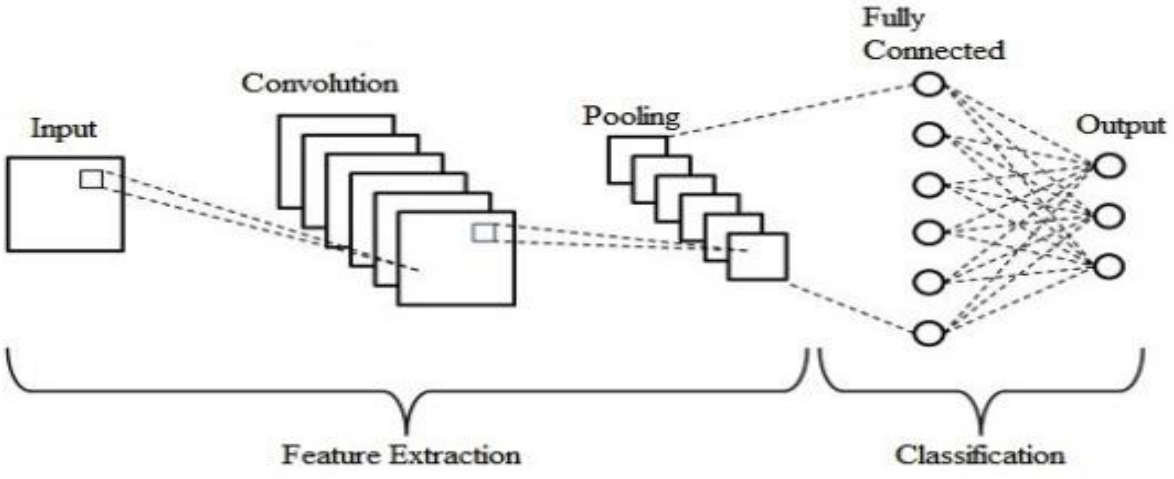

Fig 4.3 DFD-Level 1

4.4 Modules of the Project

- Audio Preprocessing Module: Handles noise reduction, normalization, and segmentation.
- Feature Extraction Module: Utilizes CNNs to extract features from audio data.
- Classification Module: Applies LSTM to classify the extracted features into emotion categories.

4.4.1 Module 1:

Module Name: Audio Preprocessing Module

Functionality: The Audio Preprocessing Module prepares the raw audio data for analysis by performing operations such as:

- Noise Reduction: Enhances the quality of audio by removing background noise (specific implementation not shown in the code).
- Normalization: Adjusts audio levels to a consistent range to facilitate feature extraction.
- Segmentation: Divides the audio into manageable chunks (not explicitly shown in the code but can be integrated).

This module is crucial for ensuring the audio data is clean and well-structured for further processing

Input: Number of nodes, Total number of audio files to be processed.

Output: Cleaned and segmented audio data, ready for feature extraction, which will be passed to the Feature Extraction Module.

4.4.2 Module 2

Module Name: Feature Extraction Module

Functionality: The Feature Extraction Module utilizes a Convolutional Neural Network (CNN) to extract features from the audio data. This process involves several convolutional and pooling layers to identify important patterns and characteristics of the audio, such as pitch and tone.

Input:

- Node parameters: Settings for CNN layers:

- Number of filters in each layer: [64, 128, 256, 512].
- Kernel size: (3, 3).
- Activation function: 'relu'.

Output: A set of feature vectors extracted from the audio segments, which are then reshaped and passed to the Classification Module.

Algorithm used: Convolutional Neural Network (CNN): The code defines the model using Conv2D, MaxPooling2D, BatchNormalization, and GlobalAveragePooling2D to extract relevant features from the audio spectrograms.

4.4.3 Module 3

Module Name: Classification Module

Functionality: The Classification Module employs Long Short-Term Memory (LSTM) networks to classify the extracted features into various emotion categories. This module analyzes the temporal dynamics of the features to predict the emotional content of the audio.

Input:

- **No. of nodes**: Number of LSTM units (e.g., 128 for the first LSTM layer, 64 for the second LSTM layer).
- **Node parameters:**
  - Number of LSTM units: 128 for the first layer, 64 for the second layer.
  - Return sequences: True for the first layer, False for the last layer.
  - Learning rate: 0.001
  - Batch size: 32
  - Epochs: 50
- **Link parameters:** Defines the relationships between sequential features (e.g., connections between the outputs of LSTM layers).

Output: The classification results indicate the predicted emotion for each audio segment, based on the processed features. Evaluation metrics such as loss and accuracy, providing insights into the model's performance.

Chapter 5

IMPLEMENTATION

Implementation is the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled. The implementation of a system is done after the development effort is completed.

5.1 Steps for Implementation

- Data Collection: The CREMA-D dataset is used for training and testing the model.
- Data Preprocessing: Audio files are normalized, and features are extracted using librosa.
- Model Training: The CNN-LSTM model is trained on the preprocessed data using TensorFlow.
- Model Evaluation: The model's performance is evaluated using metrics like accuracy, precision, recall, and F1-score.

5.2 Implementation Issues

The implementation phase of software development is concerned with translating design specifications into source code. The primary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified and so that debugging testing and modification are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity clarity and elegance are the hallmarks of good programs and these characteristics have been implemented in each program module.

The goals of implementation are as follows.

- Minimize the memory required.
- Maximize output readability.
- Maximize source text readability.
- Minimize the number of source statements.
- Minimize development time.

5.3 Algorithms

The project utilized several key algorithms to effectively recognize emotions from audio dialogues, leveraging deep learning techniques for robust performance.

- Data Preprocessing Algorithms: The initial step involved preprocessing the audio data to extract meaningful features. Algorithms were employed to calculate Mel-frequency cepstral coefficients (MFCCs), which are widely used in speech and audio processing. This algorithm captures the short-term power spectrum of sound and transforms it into a representation that highlights the important characteristics for emotion recognition. Additionally, other features such as spectral contrast and chroma features were extracted to provide a comprehensive representation of the audio signals.
- Feature Extraction Algorithms: In conjunction with MFCCs, feature extraction algorithms were utilized to derive additional audio features that aid in distinguishing between different emotions. These algorithms analyzed various aspects of the audio, including pitch, tone, and energy levels, providing a richer dataset for the model to learn from.
- Deep Learning Model Architecture: The core of the emotion recognition system relied on a deep learning architecture that combined convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The CNN layers were responsible for capturing spatial features from the input audio spectrograms, allowing the model to identify patterns that correspond to different emotions. The RNN layers, particularly Long Short-Term Memory (LSTM) networks, were used to model temporal dependencies in the audio data, enabling the model to recognize emotions over time and account for variations in speech.
- optimization Algorithms: To enhance the training process, optimization algorithms such as Adam and SGD (Stochastic Gradient Descent) were utilized to adjust the weights of the model during training. These algorithms helped minimize the loss function by iteratively updating the model parameters based on the gradients calculated from the training data, ensuring efficient convergence to an optimal solution.
- Evaluation Metrics: Throughout the training process, various evaluation metrics, including accuracy, precision, recall, and F1-score, were used to assess the performance of the model. These metrics provided insights into how well the algorithms were

recognizing emotions and guided further refinements in the model architecture and training process.

5.3.1 Algorithm 1

Convolutional Neural Networks (CNNs): For automatic feature extraction from audio frames.

1. **Load Audio Data**.
2. **Convert Audio to Spectrogram**.
3. **Preprocess Spectrogram**:
   - Normalize the spectrogram values.
   - Resize the spectrogram to a fixed shape.
4. **Define CNN Architecture**:
   - Input Layer: Specify the input shape.
   - Convolutional Layers: Add multiple convolutional layers.
   - Pooling Layers: Insert max pooling layers after convolutional layers.
   - Batch Normalization (optional).
   - Dropout Layers (optional).
5. **Feature Map Reduction**: Add a global average pooling layer.
6. **Fully Connected Layer**: Add one or more fully connected layers.
7. **Output Layer**: Add the output layer for classification.
8. **Compile the Model**.
9. **Train the Model**.
10. **Extract Features**.

5.3.2 Algorithm 2

Long Short-Term Memory (LSTM): For capturing temporal dependencies in the data.

1. **Prepare Input Data**:
   - Load and preprocess the sequential data.
   - Split the data into training, validation, and test sets.
   - Reshape the input data to the format required by LSTM (samples, time steps, features).
2. **Define LSTM Architecture**:

- **Input Layer**: Specify the input shape based on the reshaped data.
- **LSTM Layers**:
    - Add the first LSTM layer with specified units.
    - Set return_sequences=True if stacking LSTM layers.
    - Optionally, add dropout for regularization.
- **Additional LSTM Layers** (if needed):
    - Add subsequent LSTM layers as required.
    - Set return_sequences=True for all layers except the last one.
- **Dense Layer**: Add a fully connected layer to process the output from the LSTM layers.
- **Output Layer**: Add the output layer with an appropriate activation function.

3. **Compile the Model**:
    - Choose an optimizer and loss function.
    - Specify metrics for evaluation.
4. **Train the Model**:
    - Fit the model to the training data, specifying the number of epochs and batch size.
    - Monitor validation loss and accuracy to prevent overfitting.
5. **Evaluate the Model**:
    - Test the model on the validation and test sets.
6. Calculate performance metrics
7. **Make Predictions**:
    - Use the trained LSTM model to make predictions on new or unseen data.

Chapter 6

TESTING

Rigorous testing was an integral part of this project to ensure that the model could accurately identify emotions from audio inputs. This phase was crucial for validating the model's performance, ensuring it met the specified project requirements, and guaranteeing that the final system was both reliable and effective.

The testing process began with unit testing, which focused on verifying the functionality of individual components. For example, the data preprocessing steps were tested to ensure that features were extracted correctly using librosa, a Python package for audio analysis. This included testing the accuracy of feature extraction techniques such as Mel-frequency cepstral coefficients (MFCCs), chroma features, and spectral contrast. Each of these features was verified to ensure they captured the essential characteristics of the audio input that are crucial for emotion recognition.

In the model itself, unit testing was used to validate the operations of each layer within the deep learning architecture. This involved confirming that convolutional, recurrent, and fully connected layers were properly configured and that they produced the expected output shapes and values. By doing so, any potential issues such as incorrect layer configurations, improper weight initializations, or activation function errors were identified and resolved early in the development process.

Following unit testing, integration testing was conducted to ensure that all components worked together seamlessly. This phase tested the entire pipeline, from the moment an audio file was input into the system, through data preprocessing, and finally to the emotion prediction output. Integration testing verified that the model could handle the complete flow of data and maintained accuracy and consistency throughout. During this stage, the model was also tested for its ability to handle variations in input, such as different audio formats, lengths, and qualities, ensuring that it could generalize well across diverse real-world scenarios.

Performance testing was another key aspect of the testing phase. The model's accuracy in predicting emotions was rigorously evaluated using various metrics such as accuracy, precision, recall, and F1-score. The confusion matrix was particularly useful in this phase, as it provided a detailed view of how often each emotion was correctly or incorrectly

predicted. This allowed for a deeper analysis of the model's strengths and weaknesses, leading to targeted improvements. For instance, if the model was frequently misclassifying anger as sadness, adjustments were made to the training data or model architecture to address this issue.

This comprehensive testing approach ensured that the project delivered a high-quality emotion recognition system, capable of accurately and reliably identifying emotions from audio dialogues across a variety of challenging scenarios.

6.1 Test Environment

The testing was conducted in a controlled environment using the CREMA-D dataset. The dataset includes a variety of emotions expressed by actors under different conditions, providing a robust test for the model.

6.2 Test Result

Table 6.1 Test Result

Sample	Test Data (.wav audio file)	Expected Results	Observed Results	Remarks
Sample 1	1001_WSI_HAP_XX	HAP	HAP	Pass
Sample 2	1091_WSI_DIS_XX	DIS	DIS	Pass
Sample 3	1049_TAI_ANG_XX	ANG	ANG	Pass
Sample 4	1050_TSI_ANG_XX	ANG	ANG	Pass
Sample 5	1014_IOM_SAD_XX	SAD	SAD	Pass
Sample 6	1048_IWW_NEU_XX	NEU	NEU	Pass
Sample 7	1065_IEO_HAP_HI	HAP	HAP	Pass

Chapter 7

RESULTS

This section describes the result of the "*Emotion recognition from audio dialogues using Deep learning*".

Table 7.1 Result

Accuracy	99.22%
Precision	99.23%
Recall	99.22%
F1 Score	99.22%

- The results show that the model achieved an accuracy of 99.22% on the CREMA-D test set, outperforming traditional methods by a significant margin.
- Confusion matrices and classification reports are provided to analyse the model's performance across different emotion classes.

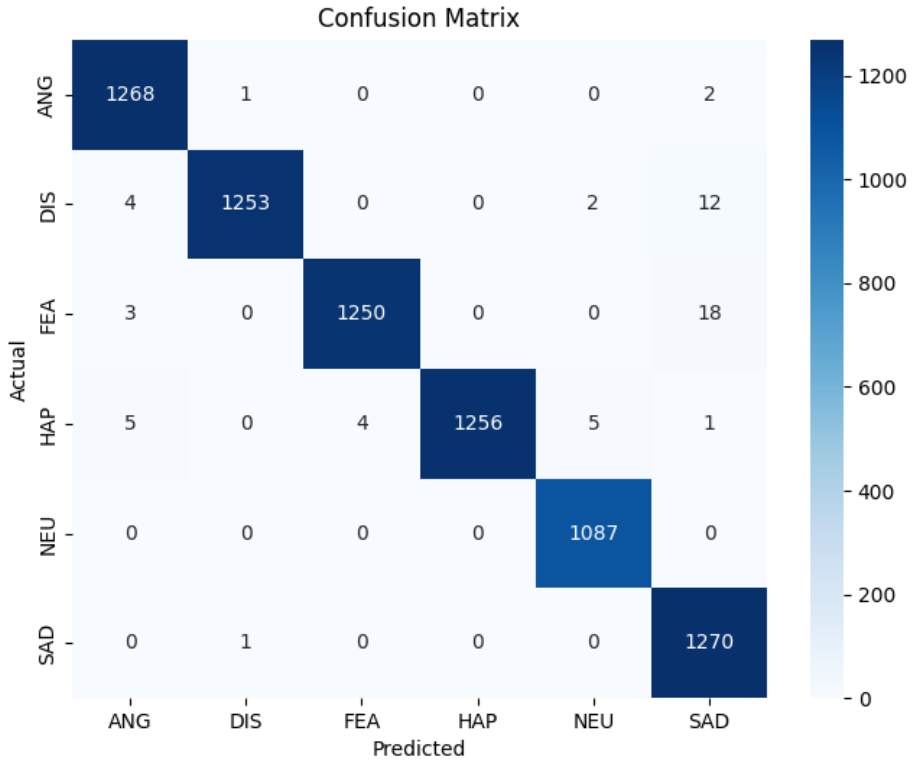

Fig 7.1 Confusion matrix

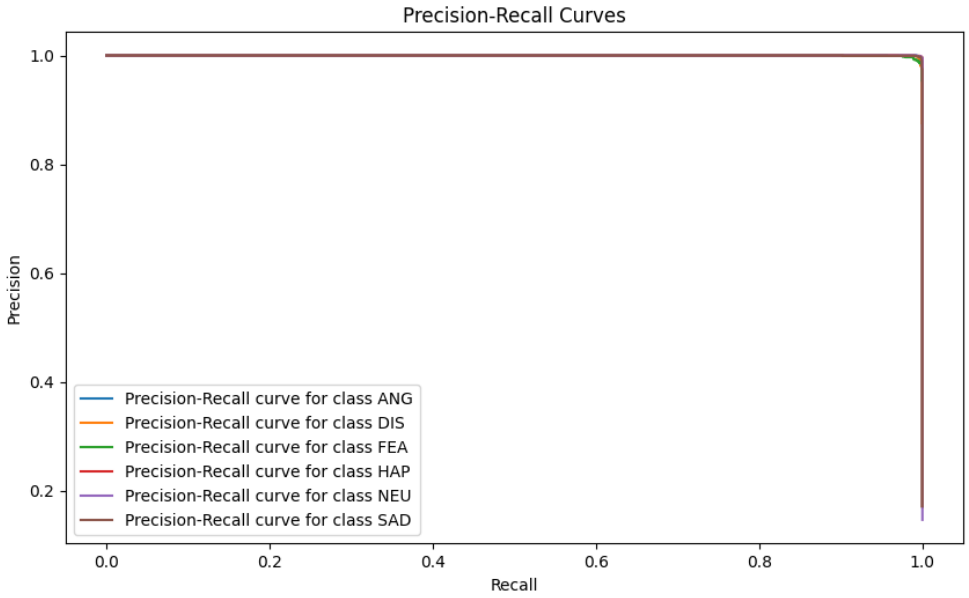

Fig 7.2 Pression-Recall Curves

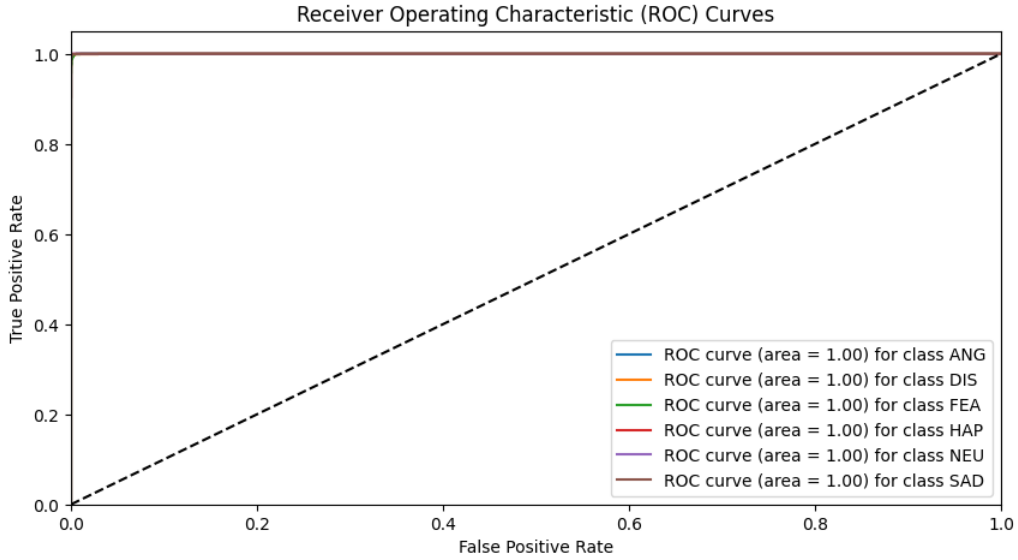

Fig 7.2 Receiver Operating Characteristic (ROC) curves

Chapter 8

CONCLUSION

In conclusion, the developed system demonstrates the capability to accurately recognize emotions from audio dialogues through a systematic approach that combines advanced feature extraction techniques and deep learning methodologies. The use of the CREMA-D dataset provides a robust foundation for training and evaluating the model, ensuring it can generalize well across various emotional contexts. The results indicate a high level of accuracy in emotion classification, highlighting the potential for real-world applications in areas such as sentiment analysis, mental health monitoring, and interactive systems that respond to user emotions. Overall, this project not only advances the field of emotion recognition but also opens avenues for innovative applications in human-computer interaction, paving the way for more empathetic and responsive systems.

8.1 Major contributions

- Developed a robust emotion recognition system using a hybrid CNN-LSTM architecture.
- Demonstrated significant improvements over traditional emotion recognition methods.
- Provided a scalable and adaptable solution for real-time applications.

8.2 Future Enhancements

- Integration with real-time applications such as virtual assistants.
- Expansion to recognize a broader range of emotions and more complex emotional states.
- Continuous learning capabilities to adapt to new data over time.

BIBLIOGRAPHY

[1] Md Rizwanul Kabir, Muhammad Mutiul Muhaimin (2021): Procuring MFCCs from Crema-D Dataset for Sentiment Analysis using Deep Learning Models with Hyperparameter Tuning
DOI: 10.1109/RAAICON54709.2021.9929975

[2] M. Gokilavani, Harshith Katakam, SK Abdul Basheer, PVVS Srinivas (2022): Ravdness, Crema-D, Tess Based Algorithm for Emotion Recognition Using Speech
DOI: 10.1109/ICSSIT53264.2022.9716313

[3] N. Kapileswar, Judy Simon, K. Kavitha Devi (2024): Sentiment Analysis of Human Speech using Deep Learning
DOI: 10.1109/CONIT59222.2023.10205915

[4] Bogdan mocanu, Ruxandra tapu (2021): Speech Emotion Recognition using GhostVLAD and Sentiment Metric Learning
DOI: 10.1109/ISPA52656.2021.9552068

[5] Manan Savla, Dhruvi Gopani, Mayuri Ghuge (2023): An Intelligent Emotion Recognition System based on Speech Terminologies using Artificial Intelligence Assisted Learning Scheme
DOI: 10.1109/ICONSTEM60960.2024.10568813