

A FIELD PROJECT REPORT

on

**“SPEECH EMOTION RECOGNITION”**

**Submitted**

by

221FA04112

G.Sathvika

221FA04435

K.Niharika

221FA04382

D.Yojitha

221FA04437

CH.Nithya sri

**Under the guidance of**

***Ms. Sajida Sultana. Sk***


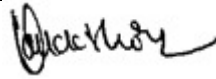
*Assistant Professor*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH**  
**Deemed to be UNIVERSITY**  
**Vadlamudi, Guntur.**  
**ANDHRA PRADESH, INDIA, PIN-522213.**

### **CERTIFICATE**

This is to certify that the Field Project entitled “**Speech Emotion Recognition**” that is being submitted by 221FA04112(Sathvika), 221FA04382(Yojitha), 221FA04435(Niharika), 221FA04437(Nithya Sri) for partial fulfilment of Field Project is a bonafide work carried out under the supervision of Sajida Sultana.Sk , Assistant Professor, Department of CSE

Sajida Sultana. Sk	 Dr. S.V. Phani Kumar	 Dr.K.V. Krishna Kishore
Assistant Professor, CSE	HOD,CSE	Dean, SoCI

## **DECLARATION**

We hereby declare that the Field Project entitled “**Speech Emotion Recognition**” is being submitted by 221FA04112 (Sathvika), 221FA04382(Yojitha), 221FA04435(Niharika) and 221FA04437(Nithya Sri) in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervision of **Ms. Sajida Sultana. Sk** ,Assistant Professor, Department of CSE.

By

**221FA04112 (Sathvika)**

**221FA04382(Yojitha)**

**221FA04435 (Niharika)**

**221FA04437 (Nithya Sri)**

Date:

# ABSTRACT

Speech is a commonly used signal for interaction between humans, this leads to the usage of speech for human and machine interactions as well. Improvements in this interactive system reach toward speech emotion recognition (SER) system. SER gives sufficient intelligence for efficient natural communication between humans and machines. SER system classifies emotional states such as sadness, angry, neutral, and happiness from the speaker's utterances. This paper describes speech features and machine learning models that can be used for SER. For effective classification and to learn multidimensional complex data, a deep learning algorithm is used in this system. This paper also presents the preliminary results of a system with an MFCC feature and an LSTM algorithm. To enhance the system's performance, data augmentation techniques and feature extraction methods like spectral contrast and chroma features are considered. The proposed system's architecture is optimized through hyperparameter tuning, which improves the classification accuracy. The results demonstrate the efficacy of combining MFCC with LSTM for accurate and robust emotion detection across varying speech patterns.

## TABLE OF CONTENTS

1.Introduction	2
1.1What is Speech Emotion Recognition?	2
1.2Importance of SER	2-3
1.3Applications of SER	3-4
1.4Objectives of the Paper	4
2.Literature Survey	
2.1Literature review	6-8
2.2 Motivation	8
3.Proposed System	48
3.1Input data	48-49
3.2Data pre-processing	49-51
3.3Feature extraction	51-52
3.4Model building	52-53
3.5Methodology of the system	53-55
3.6Model evaluation	55-56
3.7Constraints	56-57
4.Implementation	59-60
5.Result analysis	62-66
6.Conclusion	68
7. References	69

## TABLE OF FIGURES

Figure 3.1 Various features in the dataset	49
Figure 3.2 Block diagram of MFCC process	52
Figure 3.3 Architecture of LSTM	53
Figure 4.1 Data Collection	59
Figure 4.2 Accuracy	60
Figure 4.3 Loss	60
Figure 5.1 OUTPUT of Accuracy, Loss	64
Fig-5.2 output of training accuracy and validation loss	65
Fig-5.3 output of training and validation accuracy	66

## **LIST OF TABLES**

Table 1. Results for the performance of Model evaluation	55
--	----

# **CHAPTER-1**

## **INTRODUCTION**



# **1. INTRODUCTION**

Human commonly uses a vocal language for communication. This vocal language motivates researchers to think for speech communication with a machine. Multiple machines are developed based on this topic like assistance applications in a smartphone, speech to text converter and voice command operated machines. But this system lags in natural communication with humans; this activity can be improved by giving some intelligence to a machine. A machine can understand humans more efficiently when it can recognize human perception. Speech emotion recognition (SER) helps a machine to identify human emotions and react accordingly.

Humans naturally convey a wealth of emotional information through their speech patterns, including variations in pitch, intonation, rhythm, and other acoustic features. SER aims to capture and interpret these subtle cues to infer the underlying emotional state of the speaker. By analyzing speech signals, SER systems can categorize emotions into discrete categories such as happiness, sadness, anger, fear, surprise, disgust, or neutrality

## **1.1 What is Speech Emotion Recognition?**

Speech Emotion Recognition (SER) is the process of automatically identifying the emotional state of a speaker based on their speech signal. It involves analyzing various acoustic features present in the speech signal to infer the speaker's emotional state accurately. These emotional states can include happiness, sadness, anger, fear, surprise, disgust, or neutrality.

## **1.2 Importance of SER**

Speech Emotion Recognition (SER) is pivotal due to its multifaceted contributions across various domains:

1. Enhanced Human-Computer Interaction: SER empowers machines to comprehend human emotions, fostering more intuitive interactions. This advancement is vital for applications like virtual assistants and customer service bots, enhancing user experience and satisfaction.
2. Insight into Emotional States: Understanding emotions is crucial for deciphering human behavior and decision-making processes. SER provides valuable insights into emotional states, benefiting fields such as psychology, market research, and social sciences.

3. Personalization and Adaptation: By discerning emotional cues, SER enables tailored experiences. This personalization extends to recommendation systems, educational software, and adaptive learning platforms, enriching user engagement and learning outcomes.

4. Mental Health Monitoring: SER aids in the early detection and management of mental health disorders by analyzing speech patterns indicative of conditions like depression and anxiety. This contributes to personalized healthcare and early intervention strategies.

5. Improved Safety and Security: Integrating SER into security systems enables the detection of distress signals or abnormal emotional states in emergency situations. For instance, SER technology can assist in identifying callers in distress in call center environments, prioritizing urgent assistance.

6. Market Research and Customer Insight: SER facilitates the analysis of customer feedback and sentiment, guiding businesses in making informed decisions. By understanding customer emotions, companies can refine products, services, and marketing strategies to enhance customer satisfaction and loyalty.

7. Entertainment and Gaming: In the realm of entertainment, SER enhances gaming experiences by adapting game dynamics based on players' emotional responses. This real-time adaptation creates more immersive and engaging gaming experiences.

### **1.3 Applications of SER**

Speech Emotion Recognition (SER) finds diverse applications across numerous domains:

1. Customer Service: SER assists in analyzing customer interactions to gauge satisfaction levels, detect frustration, and enhance service quality in call centers and customer support services.

2. Healthcare: SER aids in mental health monitoring by detecting speech patterns indicative of mood disorders, facilitating early intervention and personalized treatment plans.

3. Education: SER enhances educational software and adaptive learning systems by tailoring content and activities based on students' emotional responses, improving engagement and learning outcomes.
4. Market Research: SER contributes to market research by analyzing customer feedback and sentiment, guiding businesses in product development, marketing strategies, and customer satisfaction initiatives.
5. Entertainment: SER enhances gaming and interactive media experiences by adapting content and game dynamics based on players' emotional responses, creating more immersive and emotionally resonant experiences.
6. Security and Safety: SER can be integrated into security systems to detect distress signals or abnormal emotional states in emergency situations, enhancing safety and security measures.

#### **1.4 Objectives of the Paper**

The objective of the paper is to develop a Speech Emotion Recognition (SER) system using machine learning techniques, specifically focusing on the utilization of Mel- Frequency Cepstral Coefficients (MFCC) and Long Short-Term Memory (LSTM) algorithm. The paper aims to address the importance of SER in improving human-machine interaction by enabling machines to recognize and respond to human emotions conveyed through speech. It discusses the challenges associated with speech emotion recognition, such as variations in speech due to speakers, speaking styles, and environmental factors. Additionally, the paper provides a literature survey highlighting existing methods and research in the field of SER.

# **CHAPTER-2**

## **LITERATURE SURVEY**

# 1. LITERATURE SURVEY

## 2.1 Literature review

A literature survey is a systematic examination of existing research on a particular topic. It serves as the foundation for any scholarly investigation, offering insights into current knowledge, identifying research gaps, and providing context for new studies. By synthesizing and summarizing relevant literature, researchers can formulate precise research questions, build upon existing work, and avoid duplication. In essence, a literature survey is an essential tool for ensuring the validity and relevance of new research within the broader academic landscape.

Yang et al. developed a Speech Emotion Recognition model combining short-term and rhythmic features, achieving 98.47% accuracy on the CASIA dataset. This LSTM-based approach shows promise for applications in HCI and mental health. Future work may focus on cross-cultural adaptability.

Nandini developed an SER system using MFCC and machine learning, reaching 93% accuracy on the RAVDESS dataset. The system, leveraging CNN and SVM, effectively classifies emotions for applications in HCI. Future work may integrate text analysis for enhanced precision.

Nath explored Speech Emotion Recognition with SAVEE and IEMOCAP datasets, using MFCC and ZCR features. Bi-LSTM and Rotation Forest models achieved top accuracies of 76% and 72%, respectively. The study suggests further model optimization for improved SER applications.

Madanian reviewed machine learning in Speech Emotion Recognition (SER), highlighting challenges like data imbalance and speaker independence, and recommending data augmentation and ensemble models to boost accuracy.

Dwivedi explore Speech Emotion Recognition (SER) using LSTM models with MFCC features, demonstrating effective emotion classification on the TESS dataset. The study highlights the model's high recognition accuracy but notes limitations due to dataset size, suggesting future work with more diverse data for improved generalizability.

Singh et al. introduce constant-Q transform-based modulation spectral features (CQT-MSF) for improved Speech Emotion Recognition (SER), showing superior performance over standard features by emphasizing low-frequency information. Their method effectively combines hand-crafted and deep neural network features, aligning closely with human auditory processing. This approach offers robust SER potential for use in healthcare, autonomous systems, and customer service.

Alluhaidan introduce a Speech Emotion Recognition (SER) approach combining MFCC and time-domain features (MFCCT) with a 1D CNN, achieving up to 97% accuracy on Emo-DB, SAVEE, and RAVDESS datasets. This hybrid model outperforms traditional methods, showing promise for real-time applications in SER. The approach provides an efficient and accurate solution for emotion detection from speech.

Pagidirayi and Anuradha analyze Speech Emotion Recognition (SER) using LSTM networks, demonstrating improved emotion classification accuracy over traditional models through temporal pattern recognition and MFCC feature extraction. Their study emphasizes LSTM's advantage in processing sequential audio data for applications in human-computer interaction. Future research may enhance SER performance by combining CNN with LSTM for hybrid modeling approaches.

de Lope and Manuel Graña review advancements in Speech Emotion Recognition (SER), focusing on machine learning and deep learning models like CNN and LSTM. Their study categorizes developments in datasets, feature extraction, and classification methods, with applications in human-computer interaction, healthcare, and IoT. They recommend expanding

datasets and enhancing model architectures to increase SER adaptability in real-world scenarios.

## **2.2 Motivation**

Speech Emotion Recognition (SER) is vital for natural human-computer interaction, allowing machines to understand and respond appropriately to human emotions conveyed through speech. SER finds applications in diverse fields like affective computing, healthcare, psychology, and entertainment. It enhances user experiences in human-computer interaction by personalizing responses based on emotional cues. Moreover, SER supports assistive technologies, aids psychological research, and drives innovation in healthcare by monitoring patients' emotional well-being. Additionally, in entertainment, SER enhances gaming experiences by adapting content to users' emotional states. Overall, SER's significance lies in its ability to enable technology to empathize and connect with humans on a deeper emotional level.

# **CHAPTER-3**

## **PROPOSED SYSTEM**



### **3. PROPOSED SYSTEM**

The proposed system for speech emotion recognition (SER) aims to develop an intelligent model capable of accurately detecting and classifying emotions expressed in human speech. The system leverages machine learning techniques and signal processing methodologies to analyze audio recordings and infer the underlying emotional states of the speakers. Key components of the proposed system include data collection and preprocessing, feature extraction using techniques such as Mel-Frequency Cepstral Coefficients (MFCC), selection of appropriate machine learning algorithms, model training, and evaluation. The dataset used for training and testing consists of diverse audio samples annotated with corresponding emotional labels. During preprocessing, techniques like noise removal, normalization, and feature scaling are applied to enhance the quality of the audio data. Feature extraction extracts relevant acoustic features from the audio signals, which serve as input to the machine learning models. Various machine learning algorithms, such as deep learning models like Long Short- Term Memory (LSTM) networks or traditional classifiers like Support Vector Machines (SVM), are evaluated for their effectiveness in emotion recognition tasks. The performance of the SER model is assessed using metrics like accuracy, precision, recall, and F1-score.

Overall, the proposed system aims to provide an efficient and reliable solution for automatically recognizing emotions from speech signals, with potential applications in areas such as human-computer interaction, sentiment analysis, and affective computing.

#### **3.1 Input dataset**

The dataset used in the presented system is Toronto emotional speech set (TESS). There are a set of 200 target words were spoken in the carrier phrase "Say the word \_" by two actresses (aged 26 and 64 years) and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). There are 2800 data points (audio files) in total. The dataset is organized such that each of the two female actor and their emotions are contain within its own folder. And within that, all 200 target words audio file can be found. The format of the audio file is a WAV format.

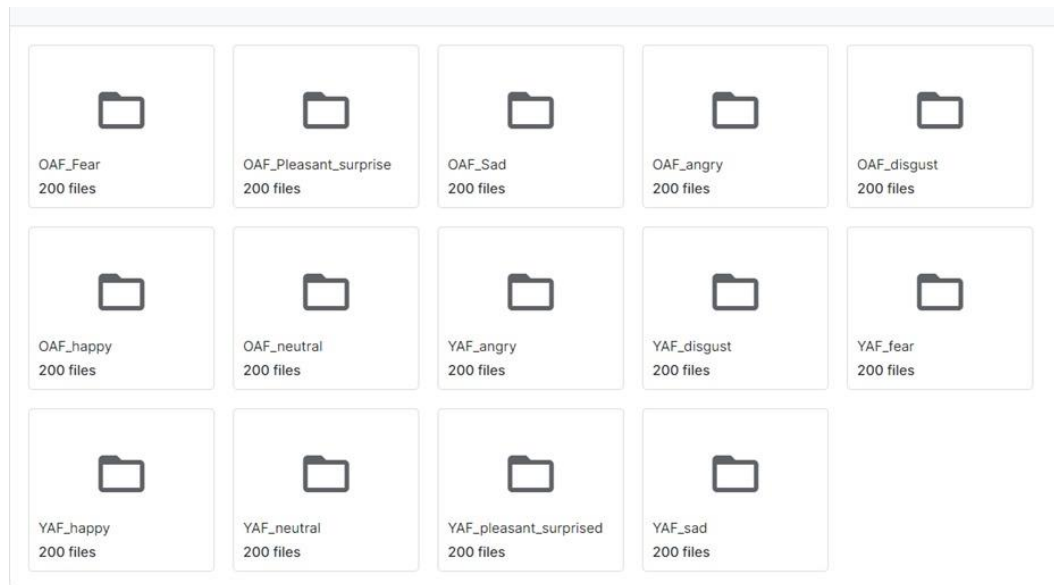


Figure 3.1 Various features in the dataset

### 3.2 Data Pre-processing

Data pre-processing is the essential process of preparing raw data for analysis and modelling by cleaning, transforming, and structuring it to enhance data quality and utility. It involves tasks like handling missing values, correcting errors, encoding features, and scaling data to ensure it's in an optimal form for further analysis. It encompasses a range of operations and transformations designed to refine raw data, ensuring that it is clean, structured, and amenity subsequent analysis. This process is driven by its manifold significance in data science and analysis.

Through meticulous data cleaning, transformation, feature engineering, dimensionality reduction, outlier handling, scaling, and data splitting, it prepares raw data for more accurate and reliable analysis and modelling. Ultimately, the goal is to obtain more meaningful insights, make informed decisions, and optimize predictive models for a wide range of applications in data science and analysis.

Here are some common techniques used for pre-processing sound data in the context of sound emotion recognition (SER):

#### 1. Noise Reduction:

- Goal: Eliminate unwanted background noise that can interfere with emotion-related features.
- Techniques:

- Filtering: Applying filters to remove specific frequencies associated with the noise, like power line hum or traffic rumble.
- Spectral subtraction: Estimating and subtracting the noise spectrum from the original signal.
- Statistical methods: Utilizing statistical properties of the noise and speech to separate them.

## 2. Silence Removal:

- Goal: Remove silent segments irrelevant to emotion recognition, improving efficiency and avoiding feeding irrelevant information to the model.
- Techniques:
  - Energy-based detection: Identifying and removing segments with energy levels below a predefined threshold.

## 3.Feature Scaling:

- Goal: Ensure all features are within a similar range, preventing one feature from dominating the model's learning.
- Techniques:
  - Normalization: Rescaling features to a range between 0 and 1.
  - Standardization: Subtracting the mean and dividing by the standard deviation of each feature.

## 4.Data Augmentation (Optional):

- Goal: Artificially increase the size and diversity of the training data to prevent overfitting and improve model generalization.
- Techniques:
  - Adding noise: Introducing controlled levels of background noise to simulate real-world scenarios.
  - Speed perturbation: Slightly increasing or decreasing the playback speed of the audio to create variations in emotional expression.
  - Pitch shifting: Adjusting the fundamental frequency of the audio to simulate different speaker characteristics.

#### 5. Missing Value Handling:

- Goal: Address missing values in the audio data that can occur due to various reasons.
- Techniques:
  - Interpolation: Estimating missing values based on surrounding data points.
  - Deletion: Removing rows or columns with a high percentage of missing values, especially if they don't significantly impact the overall data.
  - Mean/median imputation: Replacing missing values with the average or median value of the feature.

#### 6. Data Windowing:

- Goal: Divide the audio signal into smaller segments (windows) to capture the temporal dynamics of emotion within an utterance.
- Techniques:
  - Hanning window: Commonly used, smoothens the edges of the window to reduce spectral leakage.
  - Rectangular window: Simplest approach, but can introduce abrupt discontinuities at the window edges.

#### 7. Framing:

- Goal: Extract short-term features from each windowed segment.
- Techniques:
  - Overlapping windows: Ensures continuity between adjacent frames and captures information across window boundaries.

### 3.3 Feature extraction

MFCC provides a high level of perception of the human voice and achieving high recognition accuracy. Mel Frequency Cepstral Coefficient (MFCC) is a popular and powerful analytical tool in the field of speech recognition. MFCC reduces the computational complexity of the approach, gives better ability to extract the features and can find the different parameters like pitch and energy. Mel Frequency Cepstral Coefficient (MFCC) reduces the frequency information of speech signal into the small number of coefficients which is easy to compute and extract the features. It represents the short-term power spectrum of sound, based on linear cosine transform of a log power spectrum on a non-linear Mel scale of frequency. A survey shows that the MFCC gives good results as compared to other features for a speech-based emotion recognition system.

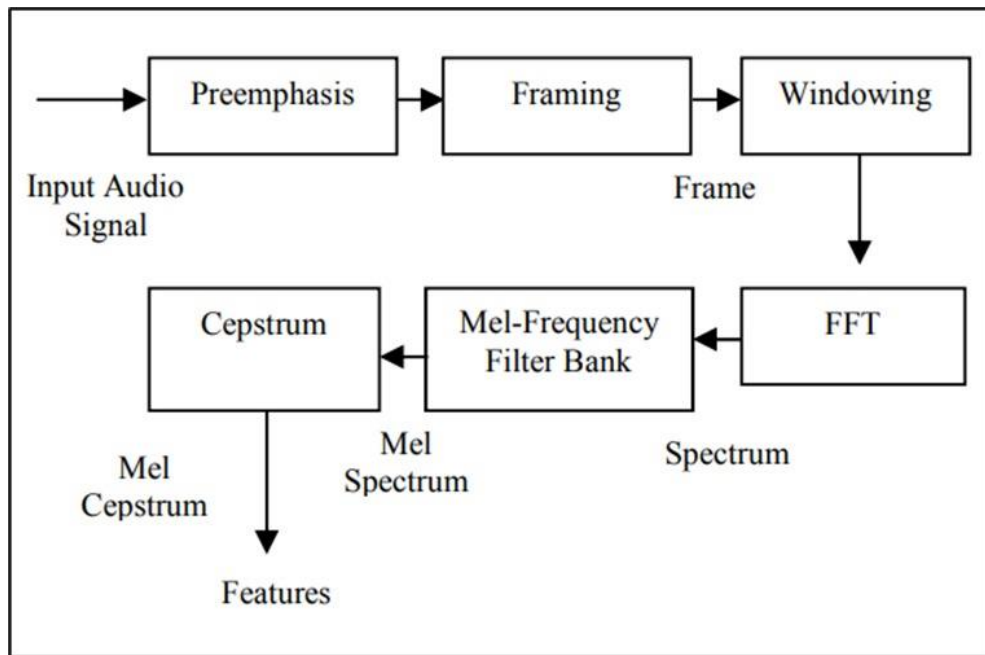


Figure 3.2 Block diagram of MFCC process

Above figure shows the block diagram of the MFCC process. Pre-emphasis is the process of the speech signal in which pre-emphasis filter is used to achieve a smoother spectral. In the frame blocking process, the sound signal is segmented into multiple small overlapped frames in the presented methodology frame size of 20ms and the step between successive frames is also 20ms. Windowing is a process required for analyzing a section of long signals. This process removes the aliasing. Fast Fourier Transform (FFT) is used to convert a time-domain signal into a frequency spectrum. Mel-Frequency filter bank used for converting a linear frequency scale to the Mel-frequency scale. Mel-frequency scale is designed according to the perception of the human ear against the sound frequency. The scale of Mel Frequency is a logarithmic scale, so it is sensitive to a lower frequency than a higher frequency. In the cepstrum process, Mel-spectrum will be converted into the time domain by using a Discrete Cosine Transform (DCT) to get the Mel frequency Cepstrum coefficient (MFCC).

### 3.2 Model building

ML model used in the presented methodology is based on LSTM architecture. Long short-term memory (LSTM) is a modified version of artificial recurrent neural network (RNN) architecture. LSTM works better with a huge amount of data and enough training data. The main advantage of RNN over ANN is in the case of a sequence of data it gives better performance. In the case of speech processing signal is framed in small pieces this small section, for emotion detection the dependency of each section with the previous one should be considered. So in this case LSTM gives better performance.

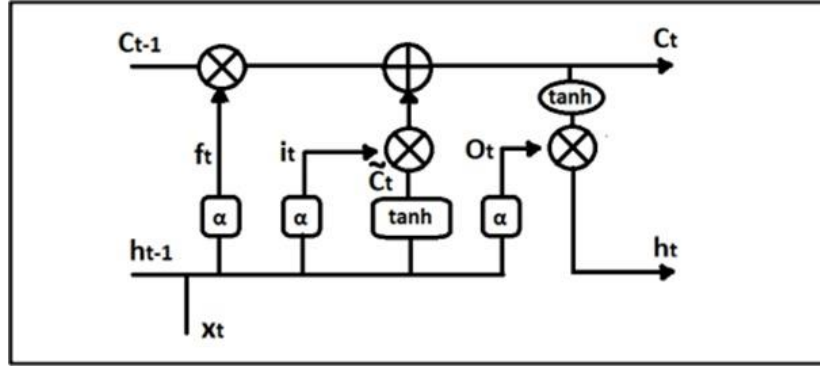


Figure 3.3 LSTM STRUCTURE

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \quad (1)$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \quad (2)$$

$$O_t = \sigma(x_t U^o + h_{t-1} W^o) \quad (3)$$

$$\tilde{C}_t = \tanh(x_t U^g + h_{t-1} W^g) \quad (4)$$

$$C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t) \quad (5)$$

$$h_t = \tanh(C_t) * O_t \quad (6)$$

The above figure shows the structure of the LSTM cell and equations describing the LSTM model.  $U$  is the weight matrix contains the inputs to the hidden layer,  $W$  is the connection between current and previous layer.  $C$  is the internal memory of the unit, which is a combination of the previous memory, multiplied by the for-get gate, and the newly computed hidden state, multiplied by the input gate.  $\tilde{C}_t$  is a candidate hidden state that is computed based on the current input and the previous hidden state .

### 3.4 Methodology of the system

Having discussed the foundational elements in the preceding sections, we now venture into the core of our traffic congestion prediction system. In this section, we embark on a journey through the inner workings of our model, unveiling the methodology that drives our system's ability to forecast traffic congestion. Just as a well-orchestrated symphony requires each instrument to play its part harmoniously, our methodology combines data, pre-processing, modelling, and evaluation to create a seamless and efficient prediction system.

The proposed speech emotion recognition (SER) system utilizes a combination of Mel- Frequency Cepstral Coefficients (MFCC) features and a Long Short-Term Memory (LSTM) network architecture. Here's a breakdown of the likely architectural components:

1. Pre-processing:

- Input: Raw audio signal.
- Steps:
- May involve noise reduction, silence removal, and other techniques to improve data quality.
- Not explicitly mentioned in the abstract, but likely present in a complete system.

2. Feature Extraction:

- Input: Pre-processed audio signal.
- Method: MFCC extraction:
- Segmenting the audio into short frames.
- Applying windowing functions to smooth the signal.
- Calculating the Mel-frequency cepstrum, a representation of the short-term power spectrum of the audio on a Mel scale (perceptually relevant scale for human hearing).

3. Machine Learning Model:

- Input: Extracted MFCC features.
- Architecture: LSTM network:
- A type of recurrent neural network (RNN) capable of capturing long-term dependencies in sequential data like speech.
- Likely structure based on the abstract:
  - Input layer: Receives the extracted MFCC features.
  - Hidden layers: One or more LSTM layers responsible for learning temporal relationships in the features.
  - Output layer: Produces predictions for the emotions present in the speech (e.g., happy, angry, sad, etc.).

4. Output:

- The model outputs the predicted emotions based on the processed audio signal.

## 5. Machine Learning Model:

- Input: Extracted MFCC features.
- Architecture: LSTM network:
- A type of recurrent neural network (RNN) capable of capturing long-term dependencies in sequential data like speech.
- Likely structure based on the abstract:

Input layer: Receives the extracted MFCC features.

Hidden layers: One or more LSTM layers responsible for learning temporal relationships in the features.

Output layer: Produces predictions for the emotions present in the speech (e.g., happy, angry, sad, etc.).

## 6. Output:

- The model outputs the predicted emotions based on the processed audio signal.

Model: "sequential\_8"

Layer (type)	Output Shape	Param #
lstm_8 (LSTM)	(None, 123)	61,500
dense_24 (Dense)	(None, 64)	7,936
dropout_16 (Dropout)	(None, 64)	0
dense_25 (Dense)	(None, 32)	2,080
dropout_17 (Dropout)	(None, 32)	0
dense_26 (Dense)	(None, 7)	231

Total params: 71,747 (280.26 KB)  
Trainable params: 71,747 (280.26 KB)  
Non-trainable params: 0 (0.00 B)

## 3.5 Model evaluation

This model summary describes a neural network built with Keras, featuring an LSTM (Long Short-Term Memory) layer and several Dense (fully connected) and Dropout layers. The network is named "sequential\_8" and consists of six layers:

1. LSTM layer with 123 units, contributing 61,500 trainable parameters. This layer is designed to capture temporal dependencies in sequence data.
2. Dense layer with 64 units, followed by a \*Dropout layer\* (which has no trainable parameters) to prevent overfitting.
3. Another Dense layer with 32 units, followed by a second \*Dropout layer\*.



4. Final Dense layer with 7 output units, likely representing a multi-class classification with 7 classes.

The model has a total of 71,747 trainable parameters and no non-trainable parameters. The compact architecture and dropout layers suggest it is designed for efficient learning and reduced overfitting in sequential data tasks.

### **3.6 CONSTRAINTS**

Constraints in speech emotion recognition (SER) systems can arise from various factors, impacting their effectiveness and accuracy:

1. **Variability in Speech:** Natural speech exhibits considerable variability in terms of accent, dialect, speaking rate, pitch, and tone. This variability poses a challenge for SER systems to generalize across different speakers and contexts.
2. **Limited Data Availability:** Building robust SER models requires large and diverse datasets encompassing various emotions, languages, and speaking styles. However, acquiring such datasets may be challenging due to privacy concerns, data annotation costs, and the need for balanced emotional representations.
3. **Ambiguity in Emotional Expression:** Emotions are complex and multidimensional, often expressed through subtle variations in speech prosody, intonation, and linguistic content.
4. **Disentangling these subtle cues and accurately recognizing emotions poses a significant challenge for SER systems.**
5. **Noise and Environmental Factors:** Environmental noise, microphone quality, and recording conditions can degrade the quality of speech signals, affecting the performance of SER systems.
6. **Cultural and Individual Differences:** Emotion expression can vary across cultures and individuals, making it challenging to develop universal SER models that generalize across diverse populations. Cultural norms, social context, and individual personality traits influence how emotions are expressed and perceived.

7. Real-time Processing: In applications requiring real-time emotion recognition, such as human interactions or virtual assistants, SER systems must operate with low latency and high efficiency. Real-time processing constraints impose limitations on model complexity and computational resources.

8. Addressing these constraints requires interdisciplinary research efforts spanning signal processing, machine learning, psychology, linguistics, and ethics to advance the capabilities and reliability of SER systems for practical applications.

# **CHAPTER-4**

## **IMPLEMENTATION**

## 4.Implementation

### 4.1 Configuration Environment

In this project, To configure the environment for extracting MFCC features and training a neural network model, install the necessary libraries: librosa for audio processing, tensorflow for model building, pandas for data manipulation, scikit-learn for evaluation metrics, and matplotlib for visualization. Run the following command: `pip install numpy pandas librosa tensorflow scikit-learn matplotlib`. In your Python script or Jupyter notebook, import these libraries to handle tasks like loading audio, extracting MFCCs, splitting data, building LSTM-based neural networks, and evaluating the model's performance. This setup provides a comprehensive environment for audio-based machine learning tasks, particularly for speech or sound classification.

```
from keras.models import Sequential
from keras.layers import Dense,LSTM,Dropout

# Assuming y has 10 classes based on the error message,
# adjust the final Dense layer to have 10 units
model=Sequential([
    LSTM(123,return_sequences=False,input_shape=(40,1)),
    Dense(64,activation='relu'),
    Dropout(0.2),
    Dense(32,activation='relu'),
    Dropout(0.2),
    Dense(10,activation='softmax') # Changed to 10 units
])
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
model.summary()

# Train the model
history = model.fit(x,y,validation_split=0.2,epochs=100,batch_size=512,shuffle=True)
```

**Figure-4.1 Data Collection**

```
epochs=list(range(100))
acc=history.history['accuracy']
val_acc=history.history['val_accuracy']
plt.plot(epochs,acc,label='train accuracy')
plt.plot(epochs,val_acc,label='val accuracy')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.legend()
plt.show()
```

**Figure-4.2 Accuracy**

```
loss=history.history['loss']
val_loss=history.history['val_loss']
plt.plot(epochs,loss,label='train accuracy')
plt.plot(epochs,val_loss,label='val loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

**Figure-4.3 Loss**

# **CHAPTER-5**

## **TESTING AND OUTCOME ANALYSIS**

```

Epoch 1/100
9/9 ————— 7s 453ms/step - accuracy: 0.1342 - loss: 2.2465 - val_accuracy: 0.3799 - val_loss: 1.9997
Epoch 2/100
9/9 ————— 4s 417ms/step - accuracy: 0.3337 - loss: 1.9317 - val_accuracy: 0.3852 - val_loss: 1.7564
Epoch 3/100
9/9 ————— 4s 284ms/step - accuracy: 0.4574 - loss: 1.5977 - val_accuracy: 0.5703 - val_loss: 1.2877
Epoch 4/100
9/9 ————— 5s 275ms/step - accuracy: 0.5744 - loss: 1.2093 - val_accuracy: 0.6032 - val_loss: 0.9873
Epoch 5/100
9/9 ————— 4s 481ms/step - accuracy: 0.7235 - loss: 0.8562 - val_accuracy: 0.7180 - val_loss: 0.7129
Epoch 6/100
9/9 ————— 4s 330ms/step - accuracy: 0.8129 - loss: 0.6235 - val_accuracy: 0.8265 - val_loss: 0.5115
Epoch 7/100
9/9 ————— 5s 272ms/step - accuracy: 0.8832 - loss: 0.4495 - val_accuracy: 0.8790 - val_loss: 0.3639
Epoch 8/100
9/9 ————— 3s 296ms/step - accuracy: 0.9069 - loss: 0.3648 - val_accuracy: 0.9030 - val_loss: 0.2935
Epoch 9/100
9/9 ————— 6s 652ms/step - accuracy: 0.9337 - loss: 0.2759 - val_accuracy: 0.9626 - val_loss: 0.1364
Epoch 10/100
9/9 ————— 7s 269ms/step - accuracy: 0.9460 - loss: 0.2275 - val_accuracy: 0.9528 - val_loss: 0.1422
Epoch 11/100
9/9 ————— 3s 282ms/step - accuracy: 0.9447 - loss: 0.2062 - val_accuracy: 0.9555 - val_loss: 0.1520
Epoch 12/100
9/9 ————— 4s 461ms/step - accuracy: 0.9594 - loss: 0.1824 - val_accuracy: 0.9591 - val_loss: 0.1194
Epoch 13/100
9/9 ————— 3s 353ms/step - accuracy: 0.9643 - loss: 0.1513 - val_accuracy: 0.9635 - val_loss: 0.1249
Epoch 14/100
9/9 ————— 4s 285ms/step - accuracy: 0.9648 - loss: 0.1620 - val_accuracy: 0.9511 - val_loss: 0.1461
Epoch 15/100
9/9 ————— 3s 276ms/step - accuracy: 0.9692 - loss: 0.1302 - val_accuracy: 0.9742 - val_loss: 0.0793
Epoch 16/100
9/9 ————— 3s 323ms/step - accuracy: 0.9764 - loss: 0.1143 - val_accuracy: 0.9609 - val_loss: 0.1120
Epoch 17/100
Epoch 45/100
9/9 ————— 3s 313ms/step - accuracy: 0.9882 - loss: 0.0522 - val_accuracy: 0.9973 - val_
Epoch 46/100
9/9 ————— 3s 371ms/step - accuracy: 0.9937 - loss: 0.0367 - val_accuracy: 0.9920 - val_
Epoch 47/100
9/9 ————— 4s 444ms/step - accuracy: 0.9908 - loss: 0.0364 - val_accuracy: 0.9964 - val_
Epoch 48/100
9/9 ————— 4s 273ms/step - accuracy: 0.9941 - loss: 0.0325 - val_accuracy: 0.9982 - val_
Epoch 49/100
9/9 ————— 3s 282ms/step - accuracy: 0.9929 - loss: 0.0325 - val_accuracy: 0.9964 - val_
Epoch 50/100
9/9 ————— 2s 271ms/step - accuracy: 0.9940 - loss: 0.0380 - val_accuracy: 0.9973 - val_
Epoch 51/100
9/9 ————— 4s 423ms/step - accuracy: 0.9922 - loss: 0.0394 - val_accuracy: 0.9947 - val_
Epoch 52/100
9/9 ————— 4s 274ms/step - accuracy: 0.9927 - loss: 0.0354 - val_accuracy: 0.9982 - val_
Epoch 53/100
9/9 ————— 3s 290ms/step - accuracy: 0.9916 - loss: 0.0350 - val_accuracy: 0.9947 - val_
Epoch 54/100
9/9 ————— 5s 320ms/step - accuracy: 0.9923 - loss: 0.0376 - val_accuracy: 0.9947 - val_

```

```

Epoch 45/100
9/9 ————— 3s 313ms/step - accuracy: 0.9882 - loss: 0.0522 - val_accuracy: 0.9973 - val_loss: 0.0086
Epoch 46/100
9/9 ————— 3s 371ms/step - accuracy: 0.9937 - loss: 0.0367 - val_accuracy: 0.9920 - val_loss: 0.0198
Epoch 47/100
9/9 ————— 4s 444ms/step - accuracy: 0.9908 - loss: 0.0364 - val_accuracy: 0.9964 - val_loss: 0.0105
Epoch 48/100
9/9 ————— 4s 273ms/step - accuracy: 0.9941 - loss: 0.0325 - val_accuracy: 0.9982 - val_loss: 0.0073
Epoch 49/100
9/9 ————— 3s 282ms/step - accuracy: 0.9929 - loss: 0.0325 - val_accuracy: 0.9964 - val_loss: 0.0116
Epoch 50/100
9/9 ————— 2s 271ms/step - accuracy: 0.9940 - loss: 0.0380 - val_accuracy: 0.9973 - val_loss: 0.0115
Epoch 51/100
9/9 ————— 4s 423ms/step - accuracy: 0.9922 - loss: 0.0394 - val_accuracy: 0.9947 - val_loss: 0.0238
Epoch 52/100
9/9 ————— 4s 274ms/step - accuracy: 0.9927 - loss: 0.0354 - val_accuracy: 0.9982 - val_loss: 0.0081
Epoch 53/100
9/9 ————— 3s 290ms/step - accuracy: 0.9916 - loss: 0.0350 - val_accuracy: 0.9947 - val_loss: 0.0190
Epoch 54/100
9/9 ————— 5s 320ms/step - accuracy: 0.9923 - loss: 0.0376 - val_accuracy: 0.9947 - val_loss: 0.0304
Epoch 55/100
9/9 ————— 5s 276ms/step - accuracy: 0.9917 - loss: 0.0474 - val_accuracy: 0.9982 - val_loss: 0.0089
Epoch 56/100
9/9 ————— 2s 271ms/step - accuracy: 0.9928 - loss: 0.0400 - val_accuracy: 0.9973 - val_loss: 0.0116
Epoch 57/100
9/9 ————— 2s 267ms/step - accuracy: 0.9941 - loss: 0.0327 - val_accuracy: 0.9973 - val_loss: 0.0096
Epoch 58/100
9/9 ————— 3s 266ms/step - accuracy: 0.9940 - loss: 0.0311 - val_accuracy: 1.0000 - val_loss: 0.0058
Epoch 59/100
9/9 ————— 4s 490ms/step - accuracy: 0.9953 - loss: 0.0295 - val_accuracy: 0.9991 - val_loss: 0.0064
Epoch 60/100
9/9 ————— 3s 258ms/step - accuracy: 0.9954 - loss: 0.0285 - val_accuracy: 1.0000 - val_loss: 0.0054
Epoch 61/100
9/9 ————— 2s 267ms/step - accuracy: 0.9954 - loss: 0.0250 - val_accuracy: 1.0000 - val_loss: 0.0062
Epoch 62/100
9/9 ————— 2s 267ms/step - accuracy: 0.9954 - loss: 0.0250 - val_accuracy: 1.0000 - val_loss: 0.0062
Epoch 63/100
9/9 ————— 3s 265ms/step - accuracy: 0.9951 - loss: 0.0281 - val_accuracy: 0.9982 - val_loss: 0.0092
Epoch 64/100
9/9 ————— 4s 467ms/step - accuracy: 0.9905 - loss: 0.0444 - val_accuracy: 0.9867 - val_loss: 0.0564
Epoch 65/100
9/9 ————— 3s 320ms/step - accuracy: 0.9905 - loss: 0.0401 - val_accuracy: 0.9893 - val_loss: 0.0493
Epoch 66/100
9/9 ————— 5s 272ms/step - accuracy: 0.9872 - loss: 0.0528 - val_accuracy: 0.9947 - val_loss: 0.0228
Epoch 67/100
9/9 ————— 2s 273ms/step - accuracy: 0.9930 - loss: 0.0349 - val_accuracy: 0.9973 - val_loss: 0.0095
Epoch 68/100
9/9 ————— 3s 365ms/step - accuracy: 0.9912 - loss: 0.0430 - val_accuracy: 0.9938 - val_loss: 0.0169
Epoch 69/100
9/9 ————— 4s 269ms/step - accuracy: 0.9941 - loss: 0.0328 - val_accuracy: 0.9938 - val_loss: 0.0183
Epoch 70/100
9/9 ————— 3s 274ms/step - accuracy: 0.9923 - loss: 0.0371 - val_accuracy: 0.9991 - val_loss: 0.0070
Epoch 71/100
9/9 ————— 3s 270ms/step - accuracy: 0.9940 - loss: 0.0313 - val_accuracy: 0.9982 - val_loss: 0.0076
Epoch 72/100
9/9 ————— 3s 324ms/step - accuracy: 0.9943 - loss: 0.0299 - val_accuracy: 1.0000 - val_loss: 0.0051
Epoch 73/100
9/9 ————— 4s 474ms/step - accuracy: 0.9955 - loss: 0.0255 - val_accuracy: 0.9991 - val_loss: 0.0058
Epoch 74/100
9/9 ————— 3s 275ms/step - accuracy: 0.9938 - loss: 0.0339 - val_accuracy: 0.9991 - val_loss: 0.0061
Epoch 75/100
9/9 ————— 2s 265ms/step - accuracy: 0.9959 - loss: 0.0265 - val_accuracy: 1.0000 - val_loss: 0.0046
Epoch 76/100
9/9 ————— 3s 263ms/step - accuracy: 0.9964 - loss: 0.0231 - val_accuracy: 1.0000 - val_loss: 0.0053

```



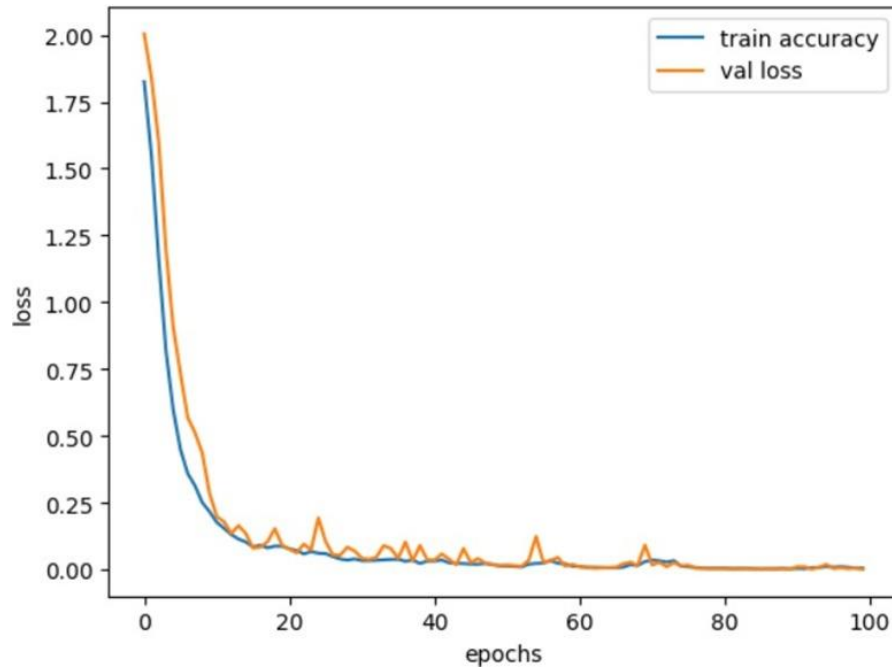
```

9/9 ————— 3s 263ms/step - accuracy: 0.9964 - loss: 0.0231 - val_accuracy: 1.0000 - val_loss: 0.0053
Epoch 77/100
9/9 ————— 3s 374ms/step - accuracy: 0.9955 - loss: 0.0254 - val_accuracy: 0.9982 - val_loss: 0.0077
Epoch 78/100
9/9 ————— 4s 442ms/step - accuracy: 0.9951 - loss: 0.0246 - val_accuracy: 1.0000 - val_loss: 0.0052
Epoch 79/100
9/9 ————— 3s 259ms/step - accuracy: 0.9939 - loss: 0.0305 - val_accuracy: 0.9929 - val_loss: 0.0318
Epoch 80/100
9/9 ————— 2s 266ms/step - accuracy: 0.9942 - loss: 0.0274 - val_accuracy: 1.0000 - val_loss: 0.0048
Epoch 81/100
9/9 ————— 2s 262ms/step - accuracy: 0.9949 - loss: 0.0254 - val_accuracy: 0.9964 - val_loss: 0.0176
Epoch 82/100
9/9 ————— 3s 344ms/step - accuracy: 0.9917 - loss: 0.0403 - val_accuracy: 0.9920 - val_loss: 0.0333
Epoch 83/100
9/9 ————— 4s 263ms/step - accuracy: 0.9912 - loss: 0.0387 - val_accuracy: 0.9982 - val_loss: 0.0087
Epoch 84/100
9/9 ————— 2s 270ms/step - accuracy: 0.9956 - loss: 0.0241 - val_accuracy: 0.9991 - val_loss: 0.0061
Epoch 85/100
9/9 ————— 2s 269ms/step - accuracy: 0.9930 - loss: 0.0343 - val_accuracy: 0.9982 - val_loss: 0.0092
Epoch 86/100
9/9 ————— 2s 266ms/step - accuracy: 0.9939 - loss: 0.0305 - val_accuracy: 1.0000 - val_loss: 0.0054
Epoch 87/100
9/9 ————— 4s 443ms/step - accuracy: 0.9948 - loss: 0.0288 - val_accuracy: 0.9956 - val_loss: 0.0247
Epoch 88/100
9/9 ————— 3s 323ms/step - accuracy: 0.9957 - loss: 0.0230 - val_accuracy: 0.9991 - val_loss: 0.0057
Epoch 89/100
9/9 ————— 4s 271ms/step - accuracy: 0.9925 - loss: 0.0432 - val_accuracy: 0.9973 - val_loss: 0.0129
Epoch 90/100
9/9 ————— 2s 278ms/step - accuracy: 0.9935 - loss: 0.0303 - val_accuracy: 1.0000 - val_loss: 0.0053
Epoch 91/100
9/9 ————— 4s 401ms/step - accuracy: 0.9935 - loss: 0.0355 - val_accuracy: 0.9920 - val_loss: 0.0318
Epoch 92/100
9/9 ————— 4s 271ms/step - accuracy: 0.9887 - loss: 0.0456 - val_accuracy: 0.9973 - val_loss: 0.0116
Epoch 93/100
9/9 ————— 4s 271ms/step - accuracy: 0.9887 - loss: 0.0456 - val_accuracy: 0.9973 - val_loss: 0.0116
Epoch 94/100
9/9 ————— 2s 271ms/step - accuracy: 0.9942 - loss: 0.0298 - val_accuracy: 1.0000 - val_loss: 0.0075
Epoch 95/100
9/9 ————— 3s 280ms/step - accuracy: 0.9940 - loss: 0.0274 - val_accuracy: 0.9991 - val_loss: 0.0069
Epoch 96/100
9/9 ————— 3s 281ms/step - accuracy: 0.9948 - loss: 0.0281 - val_accuracy: 1.0000 - val_loss: 0.0054
Epoch 97/100
9/9 ————— 5s 278ms/step - accuracy: 0.9939 - loss: 0.0307 - val_accuracy: 1.0000 - val_loss: 0.0036
Epoch 98/100
9/9 ————— 3s 278ms/step - accuracy: 0.9971 - loss: 0.0164 - val_accuracy: 1.0000 - val_loss: 0.0045
Epoch 99/100
9/9 ————— 2s 279ms/step - accuracy: 0.9970 - loss: 0.0188 - val_accuracy: 1.0000 - val_loss: 0.0060
Epoch 100/100
9/9 ————— 3s 283ms/step - accuracy: 0.9949 - loss: 0.0252 - val_accuracy: 1.0000 - val_loss: 0.0047
Epoch 100/100
9/9 ————— 5s 590ms/step - accuracy: 0.9939 - loss: 0.0304 - val_accuracy: 1.0000 - val_loss: 0.0040

```

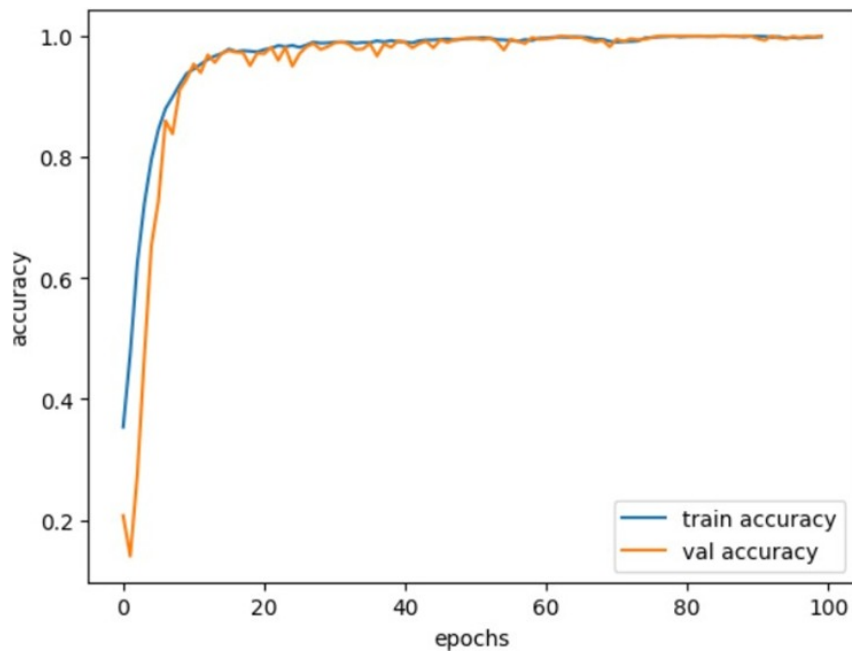
**Figure-5.1 OUTPUT of Accuracy, Loss**

This output shows the training progress of a neural network over multiple epochs, with metrics including training accuracy, loss, validation accuracy, and validation loss. The model's accuracy is consistently high (above 98%) and improves slightly over time. Both training and validation losses are generally decreasing, indicating the model is learning effectively. Validation metrics are stable, suggesting the model is nearing optimal performance. Consider using early stopping to prevent overfitting and save training time.



**Figure-5.2 Output of training accuracy and validation loss**

This plot shows the training accuracy and validation loss of a model over 100 epochs. Initially, both training accuracy improves and validation loss decreases sharply, indicating effective learning. Around epoch 20, both metrics stabilize at low values, suggesting the model has learned well and is not overfitting. The nearly flat curves from epoch 20 onward imply diminishing returns from further training. Early stopping could be considered around this point to save resources.



**Figure-5.3 output of training and validation accuracy**

This plot shows training and validation accuracy over 100 epochs. Initially, both accuracies increase rapidly, with validation accuracy catching up to training accuracy by around epoch 10. After that, both training and validation accuracies remain close to 1.0, indicating that the model performs well on both the training and validation datasets. The similar trajectories of both lines suggest minimal overfitting. This stability after early epochs implies the model has reached optimal performance.

# **CHAPTER-6**

## **CONCLUSION**

## **6.1 Conclusion**

In this work, a speech emotion recognition system with the LSTM model and MFCC feature is presented. It is observed that MFCC is a popularly used feature and gives better results for emotion detection in SER. There is a future scope in ROC curve improvement. The area observed under the ROC curve is 0.55. 67.21% loss is observed in this model, which needs to improve. The positive point observed in the implemented model is that the system achieves 84.81% accuracy. Still, there is a scope for improvement using a combination of different feature and optimizing ML model for a better true positive rate.

## **6.2 Future Enhancement:**

Future work will optimize feature selection and refine model architecture to boost positive rates and robustness. Key focus areas include adapting to diverse environments, handling background noise, and accommodating speaker variations like accents and styles. Enhanced feature combinations will help the model detect patterns more precisely, reducing false negatives. Testing across challenging scenarios will further fine-tune resilience. Ultimately, these improvements aim to create a more reliable, adaptable system for real-world applications.

# **CHAPTER – 7**

## **REFERENCES**

## References:

- [1] Yang, Z., Li, Z., Zhou, S., Zhang, L., & Serikawa, S. (2024). Speech emotion recognition based on multi-feature speed rate and LSTM. *Neurocomputing*, 601, 128177.
- [2] Panda, S. K., Jena, A. K., Panda, M. R., & Panda, S. (2023). Speech emotion recognition using multimodal feature fusion with machine learning approach. *Multimedia Tools and Applications*, 82(27), 42763-42781.
- [3] Nandini, K., Divya, T., Subhani, S., Mounika, S., & Nandhan, T. V. (2024, February). Recognition of emotional speech using MFCC and Machine Learning Technique. In *2024 International Conference on Emerging Systems and Intelligent Computing (ESIC)* (pp. 416-421). IEEE.
- [4] Nath, S., Shahi, A. K., Martin, T., Choudhury, N., & Mandal, R. (2024). Speech Emotion Recognition Using Machine Learning: A Comparative Analysis. *SN Computer Science*, 5(4), 390.
- [5] Shetty, K. J., Shetty, S., & Shetty, M. (2024, April). Speech Emotion Recognition Using LSTM. In *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)* (pp. 1-6). IEEE.
- [6] Patnaik, S. (2023). Speech emotion recognition by using complex MFCC and deep sequential model. *Multimedia Tools and Applications*, 82(8), 11897-11922.
- [7] Madanian, S., Chen, T., Adeleye, O., Templeton, J. M., Poellabauer, C., Parry, D., & Schneider, S. L. (2023). Speech emotion recognition using machine learning—A systematic review. *Intelligent systems with applications*, 200266.
- [8] Paul, B., Bera, S., Dey, T., & Phadikar, S. (2024). Machine learning approach of speech emotions recognition using feature fusion technique. *Multimedia Tools and Applications*, 83(3), 8663-8688.

- [9] Dwivedi, S., Srivastava, N., Rawal, V., Deshwal, P., & Dev, D. (2023, September). Analysing the Impact of LSTM and MFCC on Speech Emotion Recognition Accuracy. In 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET) (pp. 55-58). IEEE.
- [10] Hema, C., & Marquez, F. P. G. (2023). Emotional speech recognition using cnn and deep learning techniques. *Applied Acoustics*, 211, 109492.
- [11] de Lope, J., & Graña, M. (2023). An ongoing review of speech emotion recognition. *Neurocomputing*, 528, 1-11.
- [12] Pagidirayi, A. K., & Anuradha, B. (2023). An efficient Speech Emotion Recognition using LSTM model. *NeuroQuantology*, 21(1), 117.
- [13] Alsabhan, W. (2023). Human–computer interaction with a real-time speech emotion recognition with ensembling techniques 1D convolution neural network and attention. *Sensors*, 23(3), 1386.
- [14] Alluhaidan, A. S., Saidani, O., Jahangir, R., Nauman, M. A., & Neffati, O. S. (2023). Speech emotion recognition through hybrid features and convolutional neural network. *Applied Sciences*, 13(8), 4750.
- [15] Singh, P., Sahidullah, M., & Saha, G. (2023). Modulation spectral features for speech emotion recognition using deep neural networks. *Speech Communication*, 146, 53-69.