Enhancing Speech Emotion Recognition using

Long Short Term Memory Network

Peddi Ajay Kumar  
Department of Electronics and Communication Engineering  
 *R.V.R & J.C College of Engineering*Guntur, India  
peddiajaykumar13@gmail.com

Yericharla Sayomi  
Department of Electronics and Communication Engineering  
*R.V.R & J.C College of Engineering*Guntur, India  
yericharlasayomi@gmail.com

Ramavath Mallikharjuna Naik  
Department of Electronics and Communication Engineering  
*R.V.R & J.C College of Engineering*Guntur, India  
ramavathmallikharjunan@gmail.com

*Abstract*— Speech emotion recognition (SER) plays a crucial role in enhancing human-computer interaction by enabling machines to understand and respond to human emotions. In our project, we explore the limitations of the previously employed method that combines Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks for SER tasks. While this fusion approach has shown promise, it is hindered by the inherent limitation of LSTM cells, which retain the output of the CNN for a specific time instant. This characteristic is inadequate for capturing the complex temporal dependencies present in time-series data, leading to suboptimal performance in recognizing emotions. To address this drawback, we propose a novel method that leverages Three-layer stacked LSTM networks, which are designed to capture temporal features more effectively by processing the input data. This approach allows the model to utilize contextual information from time steps, enhancing its ability to discern emotional nuances in speech. We utilize the Toronto Emotional Speech Set (TESS) datasets, which provides a rich source of emotional speech data for training and evaluation. Additionally, we incorporate Mel-frequency cepstral coefficients (MFCCs) as supplementary features to enrich the representation of spoken words. MFCCs are known for their effectiveness in capturing the spectral characteristics of audio signals, which are vital for emotion recognition. Our experimental results may demonstrate that the proposed Multiple layer LSTM-based method significantly outperforms the traditional CNN-LSTM fusion approach, achieving higher accuracy in classifying emotional states. This advancement not only contributes to the field of speech emotion recognition but also paves the way for more sophisticated emotion-aware systems in various applications.

Keywords—LSTM, CNN, TESS, RAVDESS, EMO-DB Speech Emotion Recognition, HCI, MFCC.

# Introduction

The purpose of the exciting and quickly developing field of speech emotion recognition is to automatically determine a speaker's emotional state from their voice. Because human emotions are subtle and complicated, accurate speech emotion recognition (SER) is a difficult endeavour. The capacity of machines to comprehend and react to human emotions is essential in our increasingly linked society, where human-computer interaction is become more common. By examining and processing audio signals, Speech Emotion Recognition (SER), a fundamental tool in artificial intelligence and human-computer interaction, seeks to identify the emotional states of speakers. Enhancing SER model accuracy and real-time performance is crucial since the naturalness of human-computer interaction and user experience are determined by how well emotions are recognized [1]. Today, there is more interaction between humans and machines thanks to the growth of web and mobile applications. Human-machine communication now uses voice channels. Voice help technologies are becoming more and more involved in providing prompt, accurate answers to queries and requests. as they appear more frequently in our day-to-day encounters. For instance, the use of virtual personal assistants (VPAs) like Amazon Alexa, Google Assistant, Apple's Siri, and Cortona is spreading. These virtual personal assistants are capable of deducing the essential orders from words, but they are not proficient at recognizing people's emotions and responding appropriately [2]. People use their language to communicate their feelings. It is clear that people all across the world speak different languages. Accurately identifying emotions in real-time situations has numerous applications in various domains, enhancing system capabilities and enhancing communications between humans and machines. Since emotional reactions are an essential component of human interactions, they have logically emerged as a key component in the creation of applications that rely on human-machine connections. To achieve more spontaneous and transparent interactions between humans and machines, feelings conveyed through auditory signals need to be regularly recognized and appropriately maintained. Many machine learning techniques have been proposed and altered throughout the past 20 years of research focused on emotion interpretation. Consequently, the Speech Emotion Recognition Model (SER) was established [3].

Mel-Frequency Cepstral Coefficients (MFCC), pitch, and rhythm were examples of handcrafted elements that were frequently used in traditional emotion recognition techniques. However, when processing long-term dependencies in emotional features, where handcrafted features have limited expressive power, these features are unable to adequately capture complex emotional information. The subtleties and complexity of human emotions were difficult for these methods to convey. Deep Learning transformed the study of speech emotion recognition with its potent capacity to create hierarchical representations from unprocessed data. It also allows us to achieve notable improvements in accuracy and resilience. Traditional feature engineering has gradually been supplanted by autonomous feature extraction as deep learning has grown in popularity. Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) are being used by researchers for SER tasks. CNN's convolutional structure is good at collecting short-term emotional information from audio input because it can catch local features, but it is not very good at capturing long-term dependencies. Long Short-Term Memory (LSTM) networks, on the other hand, excel at speech recognition and emotion analysis tasks thanks to their special gating mechanisms that efficiently capture long-term relationships in time-series data. In emotion recognition, LSTM network architectures are now the standard method [1]. Multi-layer LSTM structures have been found to have better feature extraction capabilities for handling complex emotional data. Models can improve their ability to recognize emotions by extracting and processing emotional data layer by layer by stacking LSTM layers. However, too many LSTM layers can impact model training by increasing computing overhead and causing issues with gradient disappearing or exploding. Therefore, selecting the appropriate number of layers is crucial for improving model performance.

Due to the shortcomings of the previously used approach that combines Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks for SER tasks, this paper explores the application of an LSTM network for improving Speech Emotion Recognition, a difficult problem. Although this fusion method has demonstrated potential, it is hampered by LSTM cells' intrinsic restriction, which is that they only store the CNN's output for a single point in time. This feature does less well when it comes to identifying emotions since it is unable to capture the intricate temporal correlations found in time-series data. We suggest a sequential model architecture that is intended to produce excellent performance in speech emotion recognition by efficiently capturing temporal patterns. Additionally, we incorporate Mel-frequency cepstral coefficients (MFCCs) as supplementary features to enrich the representation of spoken words. MFCCs are known for their effectiveness in capturing the spectral characteristics of audio signals, which are vital for emotion recognition .

The necessity for efficient emotion classification is what drives this concept. It is especially well-suited for Speech Emotion Recognition (SER), where the advantages of using LSTM for Speech Emotion Recognition (SER) offer several important advantages, thanks to its capacity to process sequential data and learn long-term dependencies. First off, speech signals' natural temporal dependencies are well captured by LSTMs. Second, LSTMs can withstand changes in speech duration and tempo. Moreover, hierarchical representations of speech features can be learned by LSTMs. Finally, LSTMs are capable of handling incomplete and noisy speech data with ease. LSTM cells are more resilient to real-world situations where speech signals may deteriorate because of their gating processes, which enable them to selectively respond to pertinent information and filter out noise. The use of dropout after both the LSTM and dense layers mitigates overfitting, a common issue in deep learning models, particularly when dealing with limited datasets. This architecture aims to strike a balance between model complexity and generalization ability.

The following sections will detail the Literature Survey, Methodology, Experiments, and Results and Discussion, demonstrating the efficiency of our proposed LSTM-based approach for Speech Emotion Recognition (SER).

# Literature Survey

Xiaoran Yang et al. [1] builds an existing LSTM model and added a new layer to increase efficiency and accuracy in emotion classification on the RAVDESS dataset. This resulted in an overall average recognition rate of 87.33% and an improvement of 2% in accuracy when compared to single layer LSTM.

In order to capture the complex emotional content tones found in RAVDESS, EMO-DB, and IEMOCAP, Fatma Harby et al. [2] proposed a Bi-LSTM model that is used with multiple audio cues features such as MFCCs, Chroma, Mel-Spectrogram, Contrast, and Tonnetz extracted from spectrogram sequentially. The models have an accuracy rate of 90.2%, 93%, and 32%, respectively.

On the RAVDESS dataset, R. Leelavathi et al. [3] developed an LSTM model employing MFCCs as features. It achieved an overall accuracy of 78.2% and will become better by removing random silence from audio clips and increasing the volume of data by locating more annotated audio clips.

Using a hybrid CNN+LSTM model, Neha Prerna Tigga [4] et al. suggested a multiclass classification model for speech emotion recognition that uses MFCC features to deploy it across a corpus and recognizes gender-biased emotions based on the combination of TESS, RAVDESS, and SAVEE. The findings indicate that the accuracy provided by the female corpus is superior to that of the male corpus.

For the purpose of classifying emotions using the TESS dataset, Faith Sengul et al. [5] developed two distinct CNN and LSTM models. When MFCC is used, it outperforms the current CNN models with an accuracy of 99.5%.

A convolutional neural network with recurrent neural network (CNN+LSTM) model employing acceleration and velocity features is used by S. Basu et al. [6] to create an emotion recognition system with an accuracy rate of 80% on the EMO-DB dataset.

By achieving an overall average of 53.35% on the TESS dataset, J Parry et al. [7] provided a variety of models, including CNN, LSTM, and CNN-LSTM fusion, using Mel filter bank coefficients as the features.

By attaining 91.5% accuracy on the TESS dataset, K. Sankara Pandiammal et al. [7] presented an LSTM model that provides an efficient method for voice emotion recognition and addresses resource restrictions.

Md Imran Hossain et al. [9] achieved an overall accuracy of 98.21% on the TESS dataset by combining CNN, LSTM, and KNN to leverage spectral and temporal information on a range of speech samples.

On the TESS, RAVDESS, SAVEE, and CREMA-D datasets, Muralidharan et al. [8] offered various models employing CNN, LSTM, and SVM models for a bot to interact with humans using Vocal Tract, Prosodic, and Non-Linear characteristics. The CNN+LSTM model performed the best, with an accuracy of 72.66%.

Using a CNN with two layers of Bi-LSTM, Passricha et al. [9] suggested a new technique for emotion recognition in human speech and obtained an 86.43% testing accuracy on the TESS dataset.

The audio samples are compiled from the TESS, RAVDESS, and SAVEE datasets and are further enhanced by adding noise. Beena Salian et al. [10] used a different hybrid neural network that consists of four blocks of convolutional layers followed by a layer of LSTM and Mel-Spectrograms as features. The accuracy of the model's testing was 89.26%.

Zhao et al. [13] proposed an approach by constructing two models namely one 1D CNN LSTM and one 2D CNN LSTM network to learn local and global emotion-related features from speech and log-Mel spectrograms, respectively. Berlin EMO-DB speaker dependent and speaker independent sets provide 95.33% and 95.89% identification rates, respectively, whereas IEMOCAP speaker dependent and speaker independent sets provide 89.16% and 52.14% recognition accuracy, respectively.

# Proposed Methodology

## TESS Dataset

Our research on speech emotion recognition (SER) benefited greatly from the Toronto Emotional Speech Set (TESS) dataset, which we acquired from Kaggle. Its primary purpose was to train our suggested model, which includes 2800 audio files in.wav format that are narrated by two female actors known as OAF and YAF. 200 target words were recorded using the carrier phrase "Say the word \_\_\_." Seven emotional states were included in the dataset: anger, disgust, fear, happiness, pleasant surprise, sadness, and neutrality. To guarantee excellent audio quality, the recordings were made in a controlled setting. In the below figure we can see how the audio files are classified into their respective emotion labels each emotion consists of 400 audio files from two female actors for each 200 audio files respectively. The entire dataset is perfectly balanced as shown in below Figure 1.

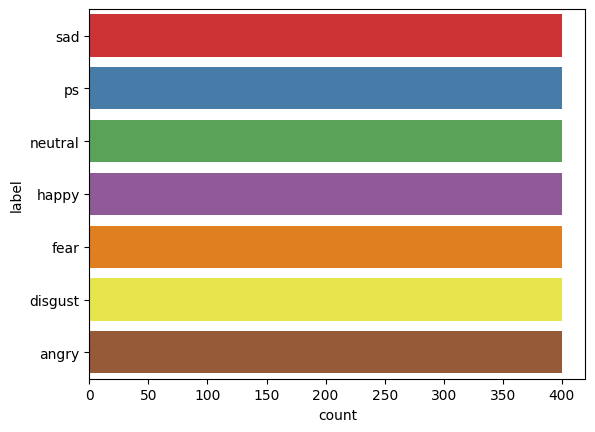


Figure 1 Distribution of voice flies in TESS Dataset

## Data Preprocessing

The raw audio data from the Toronto Emotional Speech Set (TESS) dataset underwent several preprocessing steps to prepare it for model training. First, we extracted emotion labels from the filenames and organized the audio file paths and corresponding labels into a Pandas DataFrame. The DataFrame contains two primary columns “speech” which stores the file paths to the audio recordings and “label” which holds the corresponding emotion label. It organizes the file paths and labels into a structured DataFrame, simplifying data manipulation and analysis. The analysis and visualization of audio data play a crucial role and we utilize the waveplot and spectrogram functions for generating the visual representations of audio signals, leveraging the capabilities of the librosa and matplot.pyplot libraries. The waveplot function is used to depict the audio waveform over time, which shows the insights into the signal’s amplitude variations. The core functionality of waveplot visualization is achieved by importing librosa module and sampling rate of 22.05kHz. We can clearly observe these in the audio files with different emotions as shown in the below figures.

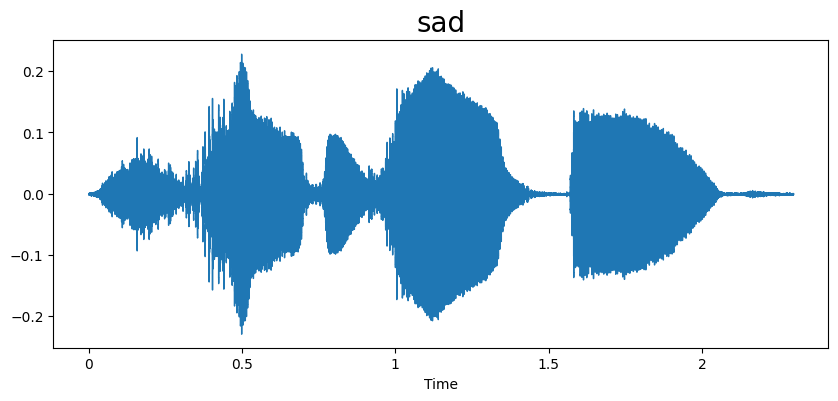


Figure 2 Visualization of sad emotion audio signal

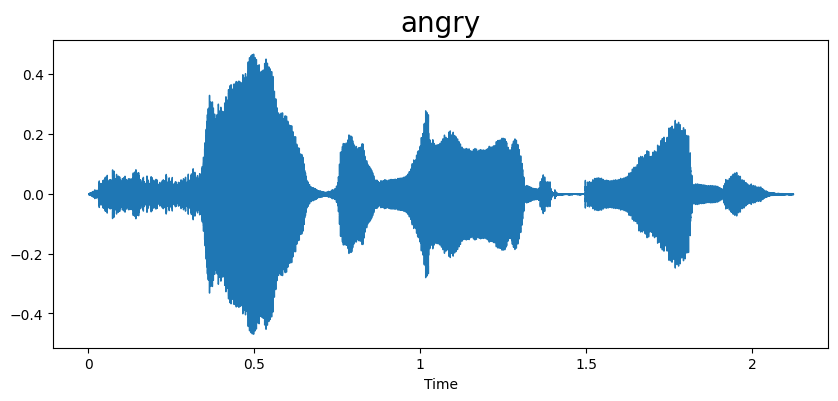


Figure 3 Visualization of angry emotion audio signal

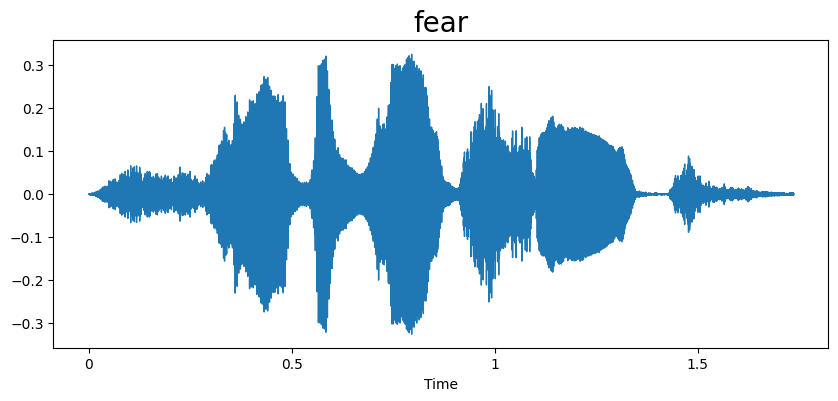


Figure 4 Visualization of fear emotion

Similarly, the spectrogram function also depicts the distribution of energy across different frequencies at various time points. It utilizes the Short-Time Fourier Transform (STFT) to divide the signals into short overlapping time frames and applies Fourier transform to each frame using the librosa module then the resulting STFT is converted into decibel scale (dB) which gives logarithmic representation of the signal intensity.

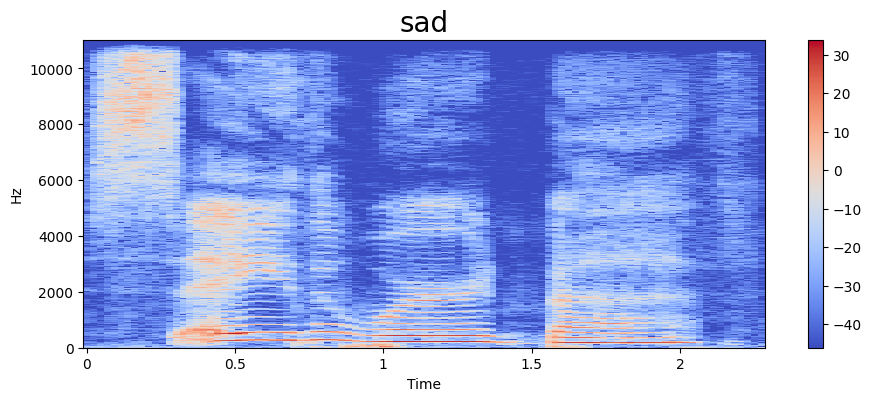


Figure 5 Visualization of sad emotion spectrogram

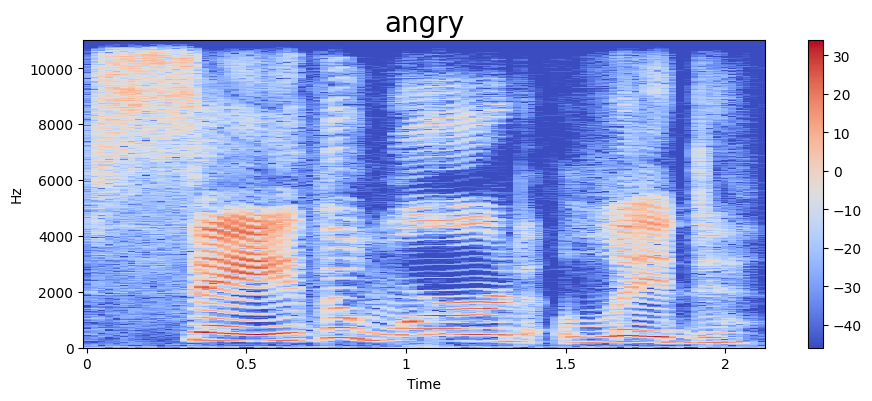


Figure 6 Visualization of angry emotion spectrogram

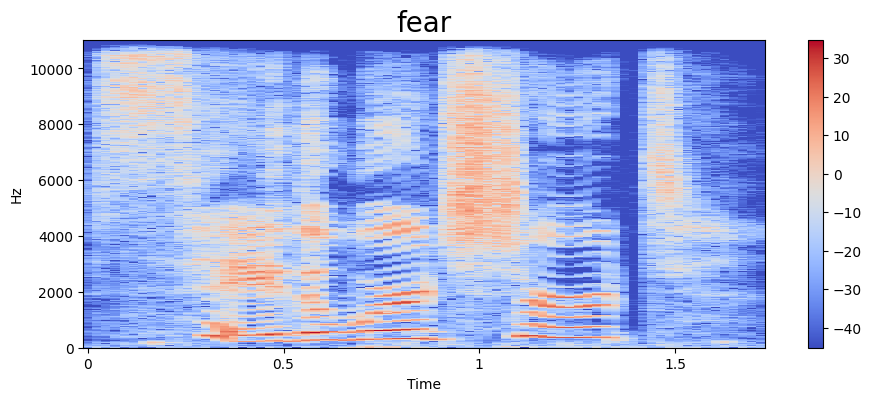


Figure 7 Visualization of fear spectrogram

By incorporating these visualization techniques, we can easily understand the spectral characteristics and complex temporal dependencies in time series data for speech emotion recognition tasks. Feature scaling and normalization are crucial preprocessing steeps for deep learning models, especially those involving distance-based calculation or gradient descent optimization which discussed briefly in the below section.

## Feature Extraction and Scaling

To capture the salient characteristics and complex temporal dependencies of the audio signals for emotion recognition, we employed Mel-frequency cepstral coefficients (MFCCs) as our primary features. MFCCs are widely used in speech and audio analysis due to their ability to effectively represent the spectral envelope of sound, which is crucial for distinguishing emotions conveyed through vocal cues.

The MFCC extraction process involves six steps as follows:

1. To increase the signal-to-noise ratio, a pre-emphasis filter is applied to each audio file once it has been loaded using the Librosa library.
2. To examine the audio signal's spectral content across time, it is split up into brief, overlapping frames. To lessen spectral leakage, a window function (such as the Hamming window) is multiplied by each frame.
3. The energy distribution across various frequencies is shown by applying the FFT to each frame, which transforms the time-domain signal into the frequency domain.
4. The Mel scale, a perceptual scale that more nearly mimics how people perceive pitch, is created by warping the frequency spectrum onto it. Frequencies that are important to human hearing are highlighted by this distortion. The following formula provides the change from the real frequency unit, Hertz, to the frequency unit, Mel.



1. The Mel filter-bank, a collection of triangle filters, is applied to the Mel-frequency spectrum. Each filter creates a series of Mel-frequency filter-bank energies by integrating the energy within a particular frequency range.
2. To compress the dynamic range and approximate the response of the human auditory system, the logarithm is applied to the filter bank energies. Lastly, the MFCCs are created by decorrelation of the filter-bank energies using the DCT.

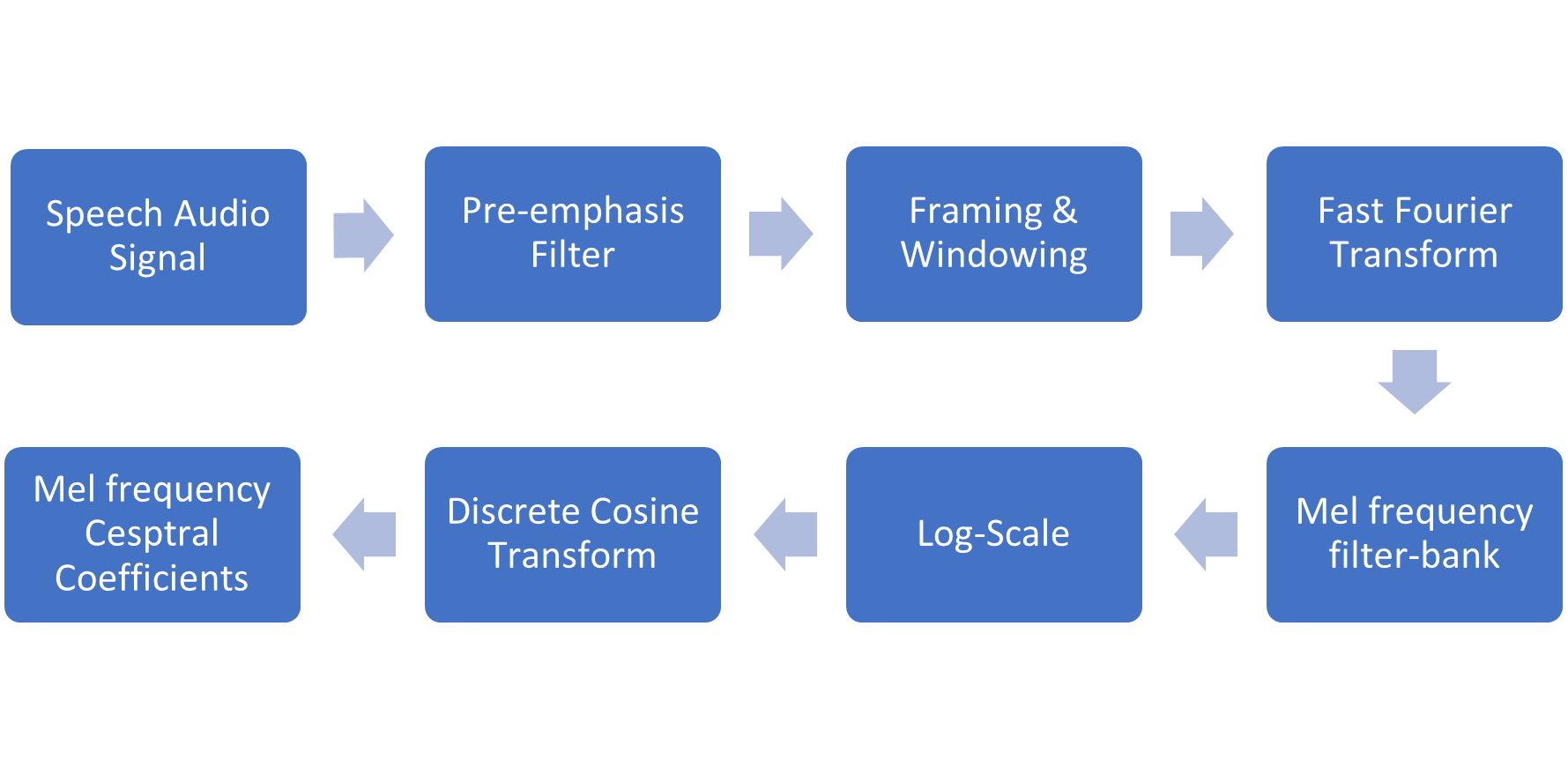


Figure 8 Illustration of MFCC Extraction

In our implementation, we extracted 40 MFCCs per audio frame, providing a comprehensive representation of the spectral envelope. To further reduce dimensionality and focus on the most relevant information, we calculated the mean of the MFCCs across all frames within each audio file.

Within the extract\_mfcc function we applied initial normalization it means the librosa.util.normalize function is applied each MFCC frame. This ensures that the MFCC values within each frame are scaled to a similar range.

After extracting MFCCs for all audio samples, StandardScaler from sklearn. preprocessing is used to standardize the features. The Scaling process involves a method known as fit\_transform which will apply scaling to the MFCC data using StandardScaler. This centers the data by subtracting the mean of each feature and then scales it by dividing by the standard deviation.

Standard Scaling ensures that features with larger ranges don’t dominate the learning process. It always prevents feature dominance. Scaling can help the model converge faster during training. Proper scaling can improve model’s ability to generalize the unseen data. Later, we apply Data Reshaping means temporal information by reshaping the data into a 3D format suitable for an LSTM network.

X.reshape(num\_samples, num\_timesteps, num\_features) converts the data into a shape where num\_samples is length of the temporal sequence, and num\_featuress is the number of features per timestep.

## Model Architecture

In our model, the model performance can be influenced by the size of the data being used. Handling large dataset can impact the efficiency and effectiveness of the model for that purpose we employed the TESS dataset for model training, evaluation and testing. Working with small or flat data can lead to overfitting problems in our model. To address the challenge and improve our model’s performance we employed the Scaling and normalization techniques for data preprocessing and we also employed One-Hot Encoding for emotion classification. Firstly, we created a Pandas DataFrame called mfcc\_df it takes the MFCC features (presumably the extracted mfccs earlier) stored in the X\_mfcc object which is like a Pandas Series and converts into a list using tolist(). This list of lists is then used to construct the DataFrame. mfcc\_df[‘label’] = df[‘label’] This line adds a new column named ‘label’ to the mfcc\_df DataFrame. It takes the emotion labels from the original DataFrame df and assigns them to the corresponding rows int the mfcc\_df. This combines the MFCC features with their respective emotion labels in a single DataFrame. By importing the OneHotEncoder class from the sklearn. preprocessing module which is used for one-hot encoding categorical variables. We create an instance of the OneHotEncoder class named as enc. By using the fit\_transform method of the enc object to learn the unique emotion labels from the ‘label’ column of the df DataFrame and transform them into a one-hot encoded representation. Let the resulting one-hot encoded labels are stored in the variable y. From this by creating the mfcc\_df DataFrame organizes the MFCC features and their corresponding emotion labels in a structured format suitable for deep learning algorithms. The One-hot encoding is essential for handling categorical features like emotion labels. It converts the labels into numerical format that can be effectively used by deep learning models preventing the model from interpreting original relationships between categories that might not exist. The model can better learn the relationship between features and the emotion labels, leading to improved classification accuracy.

### Data Spliting

This section outlines the crucial steps involved in the preparation data for the LSTM model training, evaluation and testing. Firstly, we imported the necessary libraries such as TensorFlow, sklearn. model\_selection, and NumPy. It then proceeds to split the dataset into three subsets namely training, validation and testing. This is accomplished by using the train\_test\_split function from the sklearn\_selection library. We applied stratification to the original dataset by using stratify parameter which ensures the distribution of classes in original dataset is maintained in each subset. The random\_state parameter ensures reproducibility of the split. We set split ratios as the dataset is initially split into 80% for training and 20% for testing. Subsequently, the training set is further divided into 80% for training and 20% for validation.

### LSTM Model

An expansion of the RNN (Recurrent Neural Networks) architecture and model, the LSTM technique offers a larger memory span. The LSTM model efficiently makes use of the prior information in the current neural network, whereas RNNs have a restricted "short-term memory" that does so. The hidden layer of artificial neural networks constructed using the LSTM approach generates the output signal, which is initialized and utilized as one of the values in the subsequent input. An adaptation of the artificial recurrent neural network (RNN) architecture is long short-term memory (LSTM). 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. The LSTM method is a versatile technique used in several implementations, like speech recognition, anomaly detection in the time-series data, handwriting recognition, grammar learning and music composition.

In this study, after obtaining the features from the TESS dataset, the LSTM method is used for classification. The proposed LSTM model consists of three stacked LSTM layers, two Dense layers and an output layer as shown in the figure.

Let the proposed LSTM model is comprised of stacked LSTM layers followed by two dense layers and an output layer. This architecture is designed to capture complex temporal dependencies an extract complex patterns from the input speech data. The core building blocks of the model are the LSTM layers, each consisting of multiple LSTM cells. The distinctive cell structure of the LSTM layers, which includes three crucial gates, allows them to accomplish this:

1. **Forget Gate:** This gate decides what information to discard from the cell’s memory. It takes the previous hidden state (ht-1) and the current input (xt) as input and outputs a value between 0 and 1 for each element in the cell state (ct-1). A value of 0 indicates complete forgetting while 1 means complete retention.

**ft = σ(Wf · [ht-1, xt] + bf)**

1. **Input Gate:** This gate determines what new information to store in the cell state. It consists of two parts namely a sigmoid layer that decides which values to update and a tanh layer that creates a vector of new candidate values (ĉt) that could be added to the state

**it = σ(Wi · [ht-1, xt] + bi)**

1. **Candidate Cell State:** The Candidate Cell State equation is essential for the LSTM’s ability to selectively update its memory. The Candidate Cell State represents the potential new value of the cell state, which plays a crucial role in determining what information should be added to the cell’s memory at the current time step. The input gate then determines how much of this candidate cell state should be integrated into the actual cell state.

**ĉt = tanh(Wc · [ht-1, xt] + bc)**

1. **Cell State Update:** The Cell State Update equation is crucial for the LSTM’s ability to selectively remember and forget information over time. It is the core of the LSTM’s memory mechanism. It describes how the cell state, which acts as the LSTM’s internal memory, is updated at each time step. The cell state is a vector that stores information over time, allowing thee LSTM to capture long-term dependencies in the input sequence. By controlling the input gate and forget gate, which parts of the previous cell state retain and which parts of the candidate cell state to incorporate into the current cell state. This selective memory mechanism allows the LSTM to learn long-term dependencies and avoid the vanishing gradient problem that can hinder traditional RNNs.

**ct = ft \* ct-1 + it \* ĉt**

1. **Output Gate:** This controls what information from the cell state is outputted as the hidden state (ht). It uses a sigmoid layer to filter the cell state (ct) and then multiplies the filtered state by a tanh -squashed version of the cell state to produce the output.

**ot = σ(Wo · [ht-1, xt] + bo)**

1. **Hidden State:** The Hidden state is the output of the LSTM cell and carries information about the input sequence up to the current time step. It serves as the cell’s memory and is passed to the next time step as well as other parts of the network. It is very crucial for the LSTM’s ability to selectively output information from its memory. By controlling the output gate, the LSTM can determine which parts of the cell state are relevant for the current task and should be passed on to the next time step or other parts of the network.

**ht = ot \* tanh(ct)**

Where σ represents the sigmoid activation function, tanh represents the hyperbolic tangent activation function, Wf, Wi, Wc, Wo are the weight matrices for the forget, input, cell state, and output gates, respectively, bf, bi, bc, bo are the bias vectors for the respective gates, ht-1 is the hidden state from the previous time step, xt is the input at the current time step, ct-1 is the cell state from the previous time step, ct is the updated cell state at the current time step, ht is the output hidden state at the current time step.

The above equations illustrate how the gates within a LSTM cell regulate the flow of information, allowing the network to selectively remember or forget information over time. The stacking of LSTM layers enhances the model’s ability to learn hierarchical representations of the sequential data. The first LSTM layer initially extracts emotional features from the audio signal, while the second and third LSTM layers further process and enriches the emotional information, enhancing the understanding of emotional transitions and long-term dependencies. The LSTM cell is represented in the below figure-5.

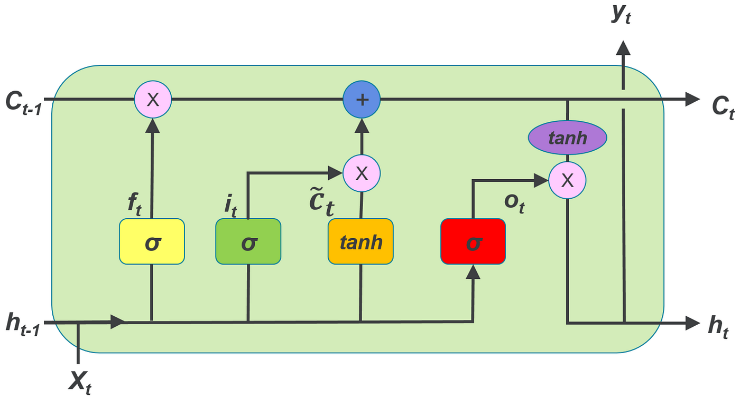


Figure 9 LSTM cell structure

Following the LSTM layers, two dense layers with ReLU activation functions are incorporated to further process the extracted features. The mathematical operation of a dense layer can be expressed as given below.

at = ReLU(Wd · ht + bd)

where ReLU represents the rectified linear unit activation function, Wd is the weight matrix of the dense layer, bd is the bias vector of the dense layer, ht is the input to the dense layer (output from the previous layer), at is the output of the dense layer. The dense layers introduce non-linearity and help the model learn complex relationships between the features.

Finally, an output layer with a SoftMax activation function is used to produce the probability distribution over the different emotion classes. The SoftMax function is given by the below equation.

pi = exp(zi) / Σj exp(zj)

Where pi is the probability of the i-th class. zi is the input to the SoftMax function for the i-th class (output from the previous layer). The output layer provides the final classification predictions for the input speech data. The LSTM model effectively learn to recognize the patterns in the speech data and predict the corresponding emotion labels. The stacked LSTM layers capture the complex temporal dependencies, the dense layers extract the complex features, and the output layer provides the final classification.

The proposed LSTM model is composed of 3 LSTM layers, 5 dropout layers, 2 dense layers, 1 SoftMax layer. The use of Dropout layers is for regularization and to prevent overfitting. The model incorporates two dense layers with L2 regularization to enhance its robustness and generalization capabilities. These layers are strategically positioned after the LSTM layers to process the extracted features and learn higher-level representations. Table 1 shows the parameters, layer, and output shape information of the LSTM model. In deep learning models, the term “output shape” refers to the ability to efficiently process datasets, which often have variable dimensions. When specifying the dimensions of the output shape, the term ‘None’ implies that it is flexible depending on the dimensions of the input data.

Table 1 Layers and Parameters of LSTM Model

|  |  |  |
| --- | --- | --- |
| Layer (type) | Output Shape | Parameters |
| lstm (LSTM) | (None, 40 256) | 264,192 |
| dropout (Droput) | (None ,40 ,256) | 0 |
| lstm\_1 (LSTM) | (None ,40, 128) | 197, 120 |
| dropout\_1(Dropout) | (None, 40, 128) | 0 |
| lstm\_2 (LSTM) | (None, 64) | 49, 408 |
| dropout\_2(Dropout) | (None, 64) | 0 |
| dense (Dense) | (None, 128) | 8, 320 |
| dropout\_3(Dropout) | (None, 128) | 0 |
| dense\_1 (Dense) | (None, 64) | 8, 256 |
| dropout\_4(Dropout) | (None, 64) | 0 |
| dense\_2 (Dense) | (None, 7) | 455 |

The above table show a summary implementation of the deep learning model. The first column shows the layers and layer types, the second column shows output shape which is the number of units or neurons for a particular layer and the third column shows the parameters per layer. The dropout rate is 0.4 is set for the dropout layer. The total instances used for training is 527, 751. The architectural representation of the proposed LSTM model for emotion classification on TESS dataset is illustrated in the below figure-6.

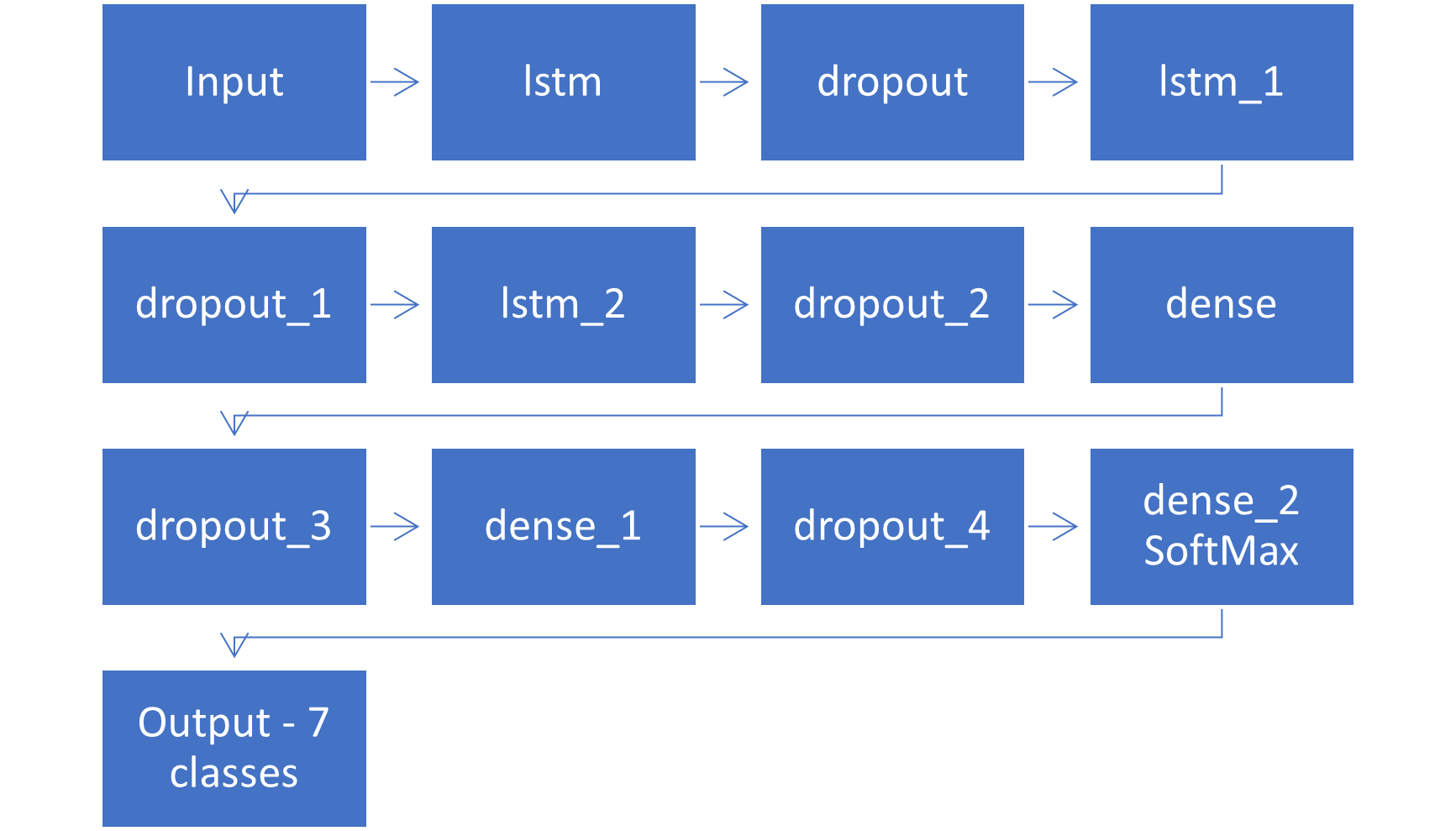


Figure 10 Architecture of Proposed LSTM model

# Experiments

## Experimental Setup

In this experiment, we use the Google Colab as the primary environment for data pre-processing, model training, evaluation and testing. By utilizing Google Colab’s, this experimental setup provides flexible, accessible, and reproducible framework for speech emotion recognition. It’s Integration with deep learning libraries and data processing tools simplifies experimentation and analysis. The ability to share Colab’s notebooks promotes collaboration and knowledge dissemination among researchers. Colab’s cloud-based infrastructure provides access to computational resources, including GPUs and TPUs, enabling efficient execution of deep learning tasks. The entire code is organized into cells, facilitating interactive experimentation and documentation. The Toronto Emotional Speech Set (TESS) dataset serves as the foundation for emotion classification. Within Colab, the dataset is accessed by mounting the Google Drive. Audio files are loaded and preprocessed using libraries like librosa which are readily available within the Colab environment.

A deep learning model with LSTM layers implemented using TensorFlow and Keras, libraries easily accessible in Colab. The model is compiled and trained on the training set using the Adam optimizer and categorical cross-entropy loss function. The use of Early stopping leads to prevent overfitting by monitoring the validation loss. The model is trained using the fit function within Colab, utilizing the CPU of the system for computation.

In this experiment, we compiled and trained the model with Adam optimizer and cross-entropy loss function. We implemented a dropout rate of 0.4 to the LSTM layers in order to avoid overfitting, with the initial learning rate set at 0.0005. We applied L2 regularization with a rate of 0.01 to the dense layers for more robustness and generalizable capabilities over unseen data. The model was mainly trained for 48 epochs and with a batch size of 48. The model uses Early stopping which monitors the validation loss to prevent overfitting and by setting restore\_best\_weights as True with a patience value of 5. The model is then evaluated with the validation set and obtained a mean accuracy of 96.25% by training the model.

* Learning rate: 0.0005
* Batch size: 48
* Training epochs: 48
* Dropout rate: 0.4

## Performance Evalution Metrics

The performance of the model was evaluated using a variety of metrics to ensure a comprehensive assessment. These measurements shed light on various facets of the model’s strengths and weaknesses. The metrics include Accuracy, Precision, Recall, F1-Score and Confusion Matrix.

* Accuracy: This metric represents the overall correctness of the model’s predictions. It is calculated as the ratio of correctly classified samples to the number of samples.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

* Precision: It is a performance metric of correctly predicted positive instances out of all predicted as positive.

Precision = TP / (TP + FP)

* Recall: It is also called as sensitivity or true positive rate that represents the number of true positive predictions made by the model divided by the total number of actual positives instances in the dataset.

Recall = TP / (TP + FN)

* F1 Score: The trade-off between sensitivity and recall is displayed using the F1 score. The F1 score is a balanced indicator of both precision and recall, calculated as the harmonic mean of the two.

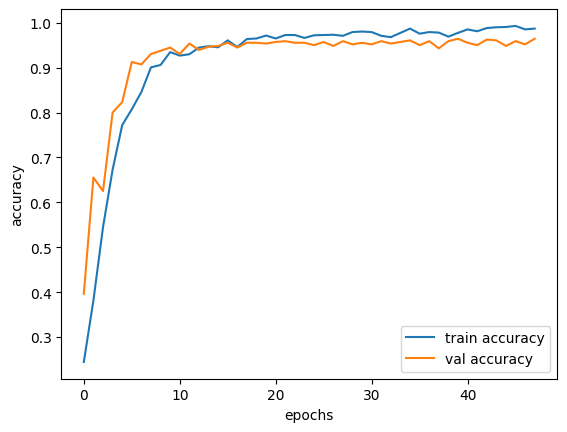
F1-Score = 2 \* (Precision \* Recall) / (Precision + Recall)

The symbols TP, TN, FP, and FN state true positive, true negative, false positive, and false negative values, respectively. These are the metrics we used to evaluate the model performance for the TESS dataset.

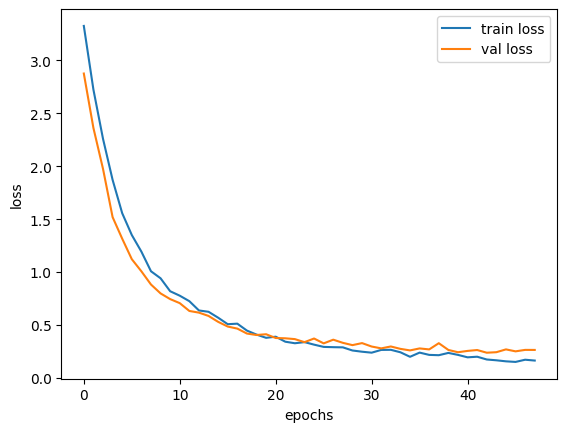
## Results and Discussion

Our proposed LSTM model was performed exceptionally very well on the TESS dataset. It achieved relatively high recognition rate across all the seven emotion labels in the TESS dataset, our model was evaluated on both the validation set and testing sets with the results presented on the table 2 and table 3.

The learning progress and assess the generalizability capability of the model, the training and validation accuracy over epochs are visualized in the plot 1. The model’s learning progress and potential issues like overfitting is also visualized by plotting the training and validation loss over epochs are visualized in the plot 2.



Plot 1 Training accuracy vs validation accuracy



Plot 2 Train loss vs validation loss

By observing these plots, we can understand the model’s performance and accuracy increasing over time, reaching a final validation accuracy of 96.25% the model is learning the underlying patterns of the data and generalizing well to unseen data, with small overfitting which is addressed by using the regularization, hyperparameter tuning. Similarly, both the training and validation loss decreased, indicating that the model is good at generalizing on unseen data.

The below table shows the macro average and weighted average of the validation set for the trained model. The values of the table further highlight the model’s balanced performance, minimizing bias towards any particular class.

For each and every class which have a support value of 80.

Table 2 Performance metrics on validation set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Angry | 0.97 | 0.95 | 0.96 | 80 |
| Disgust | 0.91 | 0.96 | 0.93 | 80 |
| Fear | 0.99 | 1.00 | 0.99 | 80 |
| Happy | 0.97 | 0.94 | 0.96 | 80 |
| Neutral | 1.00 | 1.00 | 1.00 | 80 |
| Pleasant  Surprise (ps) | 0.94 | 0.93 | 0.96 | 80 |
| Sad | 0.96 | 0.96 | 0.96 | 80 |
| Macro average | 0.96 | 0.96 | 0.96 | 560 |
| Weighted average | 0.96 | 0.96 | 0.96 | 560 |

The confusion matrix shown below is the metric obtained for validation set from the trained model it reveals a low rate of misclassification, with majority of the predictions falls along the diagonal, confirming the model’s robustness and reliability for the emotion classification task. The trained model is obtained a mean accuracy of 96.25% on the validation data.

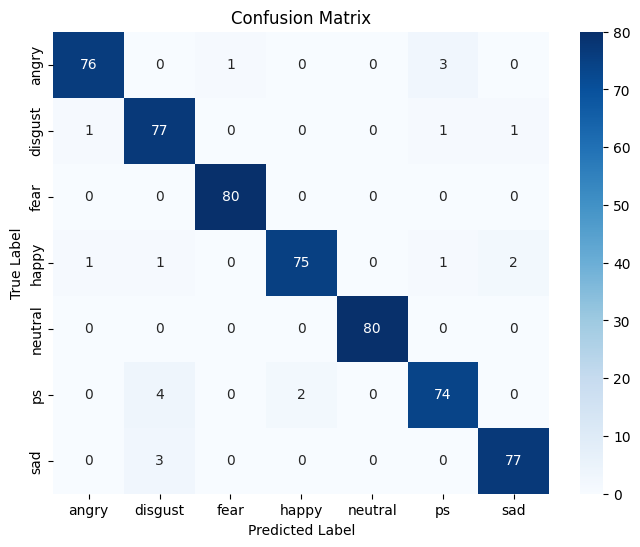


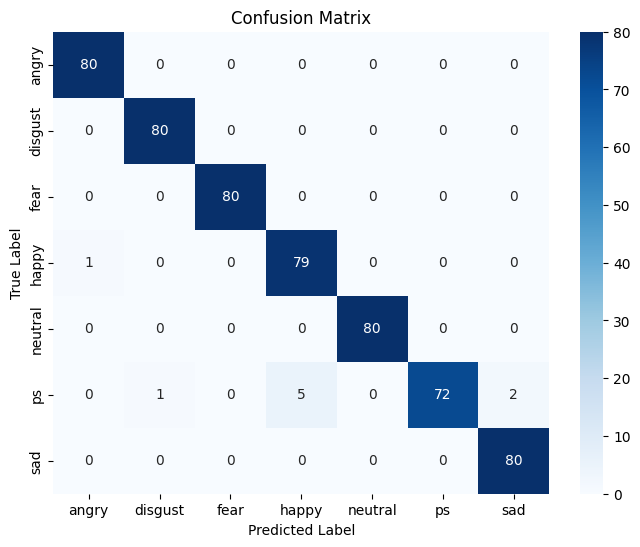
Figure 11 Confusion matrix of validation dataset

The TESS dataset is split into training, validation and testing subsets after evaluating the model on validation set later, we used the testing set for evaluate the trained model, the testing set consists of the 560 instances and stratified that means with no class imbalance which is separated from the TESS dataset. The trained model is loaded for testing and the evaluation metrics are calculated based on the model’s predictions and the ground truth labels of the testing data. The below table shows the macro average and weighted average of the testing set of the trained model. The values of the table further highlight the model’s effective performance, with no bias on any particular class with support value of 80.

Table 3 Performance metrics on testing set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Angry | 0.99 | 1.00 | 0.99 | 80 |
| Disgust | 0.99 | 1.00 | 0.99 | 80 |
| Fear | 1.00 | 1.00 | 1.00 | 80 |
| Happy | 0.94 | 0.99 | 0.96 | 80 |
| Neutral | 1.00 | 1.00 | 1.00 | 80 |
| Pleasant Surprise (ps) | 1.00 | 0.90 | 0.95 | 80 |
| Sad | 0.98 | 1.00 | 0.99 | 80 |
| Macro average | 0.98 | 0.98 | 0.98 | 560 |
| Weighted average | 0.98 | 0.98 | 0.98 | 560 |

The confusion matrix show below is the metric obtained for training set from the trained model it shows a low rate of misclassification better than training confusion matrix, with majority of the predictions are in the diagonal, confirming the model is great for emotion classification task for TESS dataset.



The pleasant sad emotion class have 8 misclassification and other emotion labels are classified with one misclassification. This shows a robust effective model performance on the testing dataset with an overall accuracy of 98.39%.

By these metrics our proposed model’s which is compared to other existing models of hybrid CNN+LSTM model shows a significant improvement of accuracy. In the majority of approaches employed in SER systems, our model is trained using the TESS dataset. Our model works fine with large datasets but show less effectiveness on small datasets like RAVDESS and EMO-DB.

Table 4 Comparison of proposed model performance with other models on TESS

|  |  |  |  |
| --- | --- | --- | --- |
| Reference | Features | Model | Accuracy |
| Md Imran Hossain et al [11] | MFCC | CNN-LSTM | 96.79% |
| Muralidharan et al [8] | Vocal Tract  Prosodic  Non-Linear | CNN+LSTM | 72.66% |
| Beenaa Salian et al [10] | Mel Spectrograms | CNN+LSTM | 89.26% |
| Passricha et al [9] | MFCC | CNN + Bi-LSTM | 86.43% |
| J. Parry et al [12] | Mel filter bank coefficients | CNN  LSTM  CNN+LSTM | 53.35% |
| Neha Prerna Tigga et al [4] | MFCC | CNN  LSTM  CNN+LSTM | 89.33%  89.61%  93.80% |
| Faith Sengul et al [5] | MFCC | LSTM | 90% |
| Proposed  Model | MFCC | 3-layer stacked  LSTM | 98.39% |

Table 4 presents the outcomes of the comparison between proficient SER systems using hybrid CNN+LSTM model, including our research work. However, in the execution of capturing the complex temporal dependencies in time series data, our system demonstrated superior performance compared to the chosen models on TESS dataset. Neha Prerna Tigga et al [4] introduced a novel approach to emotion recognition in human speech employing the hybrid CNN+LSTM model which yielded an accuracy of 93.8%.

Passricha et al [9] proposed a method for emotion recognition in human speech using a CNN with two layer of Bi-LSTM and achieved a testing accuracy of 86.43%.

Our model achieved the highest testing accuracy of 98.39% among the models evaluated, indicating its effectiveness in recognizing speech emotions on TESS dataset.

# Conclusion

In this study, we developed a deep learning model for speech emotion recognition using three-layer stacked LSTM architecture. Through extensive data preprocessing, feature extraction and model optimization, our model outperformed the benchmark of hybrid CNN+LSTM model. Our model achieved a high accuracy of 98.39% on testing dataset, demonstrating its effectiveness in classifying emotions from speech from TESS dataset.

Furthermore, by visualizing the performance via confusion matrices and the classification report, we got detailed insights into specific strengths and weaknesses of the different emotion classes. These results are well-suited for learning temporal dependencies in audio data, making them a promising approach for speech emotion recognition.

While our proposed model has demonstrated promising results on TESS dataset, it’s performance on smaller datasets are not good, this highlights an area of investigation. This limitation suggests potential challenges in generalizing to datasets with limited data diversity and size. Therefore, our future work will focus on Data Augmentation techniques, Transfer learning, Hyperparameter Optimization techniques. By addressing these aspects in future research, we aim to enhance the LSTM’s model’s performance on smaller datasets and bridge the gap between laboratory setting and real-world applications. This would facilitate the development of more robust and reliable speech emotion recognition systems for diverse domains.

# Acknowledgment

Special thanks are given to our lab colleagues for their assistance with data processing and experimental design in this study. Furthermore, we are grateful to the open-source community contributors for creating and disseminating core models and technologies that served as crucial references for our research. Lastly, we would like to express our gratitude to the TESS dataset developers for contributing high-quality emotional data, which allowed emotion recognition research to be verified on actual data.

# References

|  |  |
| --- | --- |
| [1] | Xiaoran Yang, Shuhan Yu and Wenxi Xu, “Improvement and Implementation of a Speech Emotion Recognition Model Based On Dual-Layer LSTM,” in *IEEE*, 2024. |
| [2] | F. Harby, M. Alohali, A. Thaljaoui and A. Talaat, “Exploring Sequential Feature Selection in Deep Bi-LSTM Models for Speech Emotion Recognition,” *Computers, Materials & Continua,* vol. 78, no. 2, p. 31, 2024. |
| [3] | R. Leelavathi, S. Aruna Deepthi and V. Aruna, “Speech Emotion Recognition using LSTM,” *International Research Journal of Engineering and Technology,* vol. 9, no. 1, p. 9, 2022. |
| S[4] | Neha Prerna Tigga and Shruti Garg, “Speech Emotion Recognition for multiclass classification using Hybrid CNN-LSTM,” *International Journal of Microsystems and IOT,* vol. 1, no. 1, p. 10, 2023. |
| [5] | F. Şengül and S. Akkaya, “A Modified MFCC-based Deep Learning Method for Emotion Classification from Speech,” *International Advanced Researchers and Engineering Journal,* vol. 8, no. 1, p. 10, 2024. |
| [6] | R. Basu, J. Chakraborty and M. Aftabuddin, “Emotion recognition from speech using convolutional neural network and recurrent neural network architecture,” in *2nd Internationa Conference of Communication and Electronics Systems (ICCES)*, 2017. |
| [7] | K. Sankara Pandiammal, S. Karishma, K. Harine Sakthe, V. Manimaran, S. Kalaiselvi and V. Anitha, “Emtion Recognition from Speech - an LSTM apporach with the TESS Dataset,” in *5th International Conference on Innovative Trends in Information Technology (ICITIIT)*, Kottayam, India, 2024. |
| [8] | Muralidharan and Aditya Srinevas, “Emotion Detection for a Bot using CNN and LSTM,” 2024. |
| [9] | Passricha, Vishal, Aggarwal and Rajesh Kumar, “A Hybrid of Deep CNN and Bidirectional LSTM for Automatic Speech Recognition,” *Journal of Intelligent Systems,* vol. 29, no. 1, 2020. |
| [10] | Beena Salian, Omkar Narvade, Rujuta Tambewagh and Smita Bharne, “Speech Emotion Recognition using Time-Distrubuted CNN and LSTM,” in *ITM Web of Conferences*, 2021. |
| \[11] | M. Imran Hossain, M. Mojahidul Islam, T. Nahrin, M. Rashed and M. Atiqur Rahman, “Improving Speech Emotion Recognition and Classification Accuracy Using Hybrid CNN-LSTM-KNN,” *International Journal of Research Publication and Reviews,* vol. 5, no. 8, p. 10, 2024. |
| [12] | J. Parry, D. Palaz, G. Clarke, P. Lecomte, R. Mead, M. Berger and G. Hofer, “Analysis of Deep Learning Architecture for Cross-Corpus Speech Emotion Recognition,” in *Interspeech*. |
| [13] | J. Zhao, X. Mao and L. Chen, “Speech emotion recognition using deep 1D & 2D CNN LSTM networks,” *Biomedical Signal Processing and Control,* vol. 47, p. 8, 2019. |