# ENHANCING SPEECH EMOTION RECOGNITION USING LONG SHORT MEMORY NETWORK

A thesis submitted in partial fulfillment of the
requirement for the award of the degree of

**Bachelor of Technology**
**in**
**Electronics & Communication Engineering**

by

**PEDDI AJAY KUMAR (Y21EC138)**
**YERICHARLA SAYOMI (Y21EC196)**
**RAMAVATH MALLIKHARJUNA NAIK (Y21EC146)**

Under the guidance of
**Dr K.Sravanthi** M.Tech., Ph.D
**Associate Professor in ECE**


**Department of Electronics & Communication Engineering**
**R.V.R. & J.C. COLLEGE OF ENGINEERING**
(Autonomous)
Approved by AICTE :: Affiliated to Acharya Nagarjuna University
Chowdavaram, Guntur - 522019, Andhra Pradesh, India

**2025**

# Department of Electronics & Communication Engineering


## CERTIFICATE

This is to certify that the thesis report entitled **" ENHANCING SPEECH EMOTION RECOGNITION USING LONG SHORT TERM MEMORY NETWORK "** is being submitted by **" PEDDI AJAY KUMAR (Y21EC138), YERICHARLA SAYOMI (Y21EC196), RAMAVATH MALLIKHARJUNA NAIK(Y21EC146) "** in partial fulfillment for the award of the Degree of **Bachelor of Technology** in **Electronics & Communication Engineering** to R.V.R & J.C College of Engineering (Autonomous) is a record of bonafide work carried out by them under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma.

Date:

**Signature of Guide**

**Dr K.Sravanthi** M.Tech., Ph.D

**Associate Professor in ECE**

**Signature of HOD**

**Dr.T.Ranga Babu** M.Tech., Ph.D

**Professor & Head**

# 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) dataset, 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.

# Acknowledgements

We are very thankful to Dr. Kolla Srinivas, Principal of R.V.R. J.C. College of Engineering, Guntur, for providing a supportive environment. We express our sincere thanks to Dr. T. Ranga Babu, Head of the Department of Electronics Communication Engineering, for his encouragement and support in carrying out this project successfully. We are also glad to express our special thanks to our project guide, Dr. K. Sravanthi, Associate Professor in ECE, who inspired us to select this topic. We are grateful for her valuable guidance, encouragement, and consistent support throughout the preparation and execution of this project. The successful completion of any task would be incomplete without proper suggestions, guidance, and a conducive environment. The combination of these three factors acted as the backbone of our project, "Enhancing Speech Emotion Recognition using Long Short Term Memory Network". Finally, we extend our heartfelt thanks to the faculty and supporting staff of the Department of Electronics & Communication Engineering for their cooperation and assistance during the course of this project.

**PEDDI AJAY KUMAR (Y21EC138)**

**YERICHARLA SAYOMI (Y21EC196)**

**RAMAVATH MALLIKHARJUNA NAIK(Y21EC146)**

# Table of contents

# List of Tables

# List of Figures

# 1

# 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.

## 1.1 Problem Statement

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.

## 1.2 Proposed Method

The proposed method for speech emotion recognition involves several key steps, beginning with data acquisition and preprocessing. The TESS dataset, containing audio files categorized by emotion, is loaded by iterating through the directory structure to collect file paths and extract corresponding labels, ensuring a balanced dataset across different emotions. Initial data exploration includes creating a pandas Data-Frame to manage file information and

visualizing the distribution of emotion labels. For feature extraction, the Mel-Frequency Cepstral Coefficients (MFCCs) are computed for each audio file, a standard practice in audio analysis due to their effectiveness in representing the spectral characteristics of sound relevant to human hearing. These MFCCs are then normalized and scaled using StandardScaler to ensure that features are on a similar scale, which is important for the performance of many machine learning models. The scaled MFCC data is then prepared for input into a recurrent neural network by reshaping it into a sequence format suitable for LSTM layers, specifically with dimensions representing samples, time steps, and features. The emotion labels are simultaneously one-hot encoded to align with the output requirements of the chosen model's categorical cross-entropy loss function. The core of the method is an LSTM-based deep learning model designed for sequential data processing.

This model architecture includes multiple LSTM layers to capture temporal dependencies in the MFCC sequences, interspersed with Dropout layers to mitigate overfitting by randomly dropping units during training. Dense layers with ReLU activation and L2 regularization follow the LSTM layers to perform further processing and classification, culminating in a final Dense layer with a softmax activation to output probability distributions over the emotion classes. The model is compiled using the Adam optimizer and trained using the one-hot encoded labels, with an Early Stopping callback implemented to monitor validation loss and halt training when the model's performance on unseen data begins to degrade, thus preventing overfitting and ensuring the selection of the best performing model weights. The training process involves feeding the prepared training data to the model in batches and iterating for a specified number of epochs or until early stopping is triggered, with performance monitored on a separate validation set.

Finally, the trained model's performance is comprehensively evaluated on a held-out test set using standard metrics such as accuracy, precision, recall, and F1-score, and visualized through a confusion matrix and plots of accuracy and loss over epochs to provide a clear understanding of the model's effectiveness in classifying different emotions and its learning dynamics during training.

## 1.3 Objectives

The primary objective of this project is to develop and evaluate a robust speech emotion recognition system capable of accurately classifying human emotions from audio data using deep learning techniques. A key goal is to leverage the temporal characteristics inherent in speech signals by employing a Long Short-Term Memory (LSTM) network, a type of recurrent neural network well-suited for processing sequential data like audio features. This involves several specific objectives: Firstly, to effectively load and preprocess the TESS dataset, ensuring that audio files and their corresponding emotion labels are correctly associated and organized for subsequent analysis and model training. This includes handling the file system navigation and extracting label information embedded within the filenames. Secondly, a significant objective is to perform appropriate feature extraction from the raw audio signals.

This project specifically aims to extract Mel-Frequency Cepstral Coefficients (MFCCs), which are widely recognized as effective features for capturing the phonetic and emotional content of speech while reducing dimensionality. The process involves applying techniques like windowing, Fourier transforms, and filter banks to derive these coefficients, and then further processing them through normalization and scaling to prepare them for the neural network input, ensuring that the model is not unduly influenced by the magnitude of the feature values. A third objective is to design and implement a deep learning model architecture that can effectively learn from the extracted MFCC features to distinguish between different emotional states. The choice of an LSTM model is driven by the need to capture the temporal dependencies and patterns in speech that are indicative of emotion. The architecture is designed with multiple LSTM layers and regularization techniques, such as Dropout and L2 regularization, with the specific aim of building a model that is not only accurate but also generalizes well to unseen data, thereby mitigating the risk of overfitting to the training set. A fourth objective is to train this LSTM model efficiently and effectively. This involves splitting the dataset into appropriate training, validation, and testing sets to facilitate supervised learning and provide unbiased evaluation. The training process aims to

minimize the categorical cross-entropy loss function using an optimization algorithm like Adam, adjusting the model's parameters based on the training data while monitoring performance on the validation set to guide the training process and enable early stopping when the model's performance on unseen data plateaus or degrades.

Finally, a crucial objective is to rigorously evaluate the performance of the trained model using a separate test set that was not used during training or validation. This evaluation aims to quantify the model's accuracy in classifying different emotions and to gain a deeper understanding of its strengths and weaknesses. This involves calculating various classification metrics, such as accuracy, precision, recall, and F1-score, and visualizing the results using a confusion matrix to show the distribution of correct and incorrect predictions across all emotion classes. Through these objectives, the project aims to demonstrate the feasibility and effectiveness of using LSTM networks with MFCC features for speech emotion recognition and to provide a clear analysis of the model's performance on the TESS dataset.

# 2

# LITERATURE SURVEY

Building on an existing speech recognition model, LSTM, Xiaoran Yang et al. [1] added a layer to increase efficiency and accuracy in emotion classification in the RAVDESS dataset. By capturing long-term dependencies within audio sequences through a dual-layer LSTM network,the model can recognize and classify complex emotional patterns more accurately. This resulted in an overall average recognition rate of 87.33% and an improvement of 2 in accuracy compared to a 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. In this paper,the Sequential Forward Selection(SFS) and Sequential Backward Selection(SBS) techniques were employed for multiple audio cues features selection. Finally, the feature sets composed of the hybrid feature extraction methods are fed into the deep bidirectional long-short-term memory (Bi-LSTM) network to discern emotions. The models have an accuracy rate of 90.2%, 93%, and 32% on RAVDESS, EMO-DB, IEMOCAP respectively.

On the RAVDESS dataset, R. Leelavathi et al. [8] 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 [7] 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. [3] developed two distinct CNN and LSTM models. The performances of two deep learning models, namely CNN and LSTM, are compared. The LSTM model achieved an accuracy of 90% with 14 classifications, while the CNN model outperformed with a 99.5% accuracy rate, achieving complete success in 10 out of 14 classes, with near complete success in the

remaining 4 classes. The proposed CNN model aims to classify the gender and emotion of human voices in the TESS data set. The model architecture embodies three convolutional layers, two max-pooling layers, two fully connected layers, and a dropout layer.

A convolutional neural network with recurrent neural network (CNN+LSTM) model employing acceleration and velocity features is used by R. Basu et al. [12] to create an emotion recognition system with an accuracy rate of 80% on the EMO-DB dataset.

Achieving an overall average of 53.35% on the TESS dataset, J Parry et al. [13] 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. [4] presented an LSTM model that provides an efficient method for voice emotion recognition and addresses resource restrictions. Md Imran Hossain et al. [5] 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. [6] 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 technique for emotion recognition in human speech and obtained an 86.43% testing accuracy.

The audio samples are compiled from the TESS, RAVDESS, and SAVEE datasets and are further enhanced by adding noise. Beena Salian et al. [9] used a 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. [11] proposed constructing 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.

# 3

# METHODOLOGY

## 3.1 Block Diagram

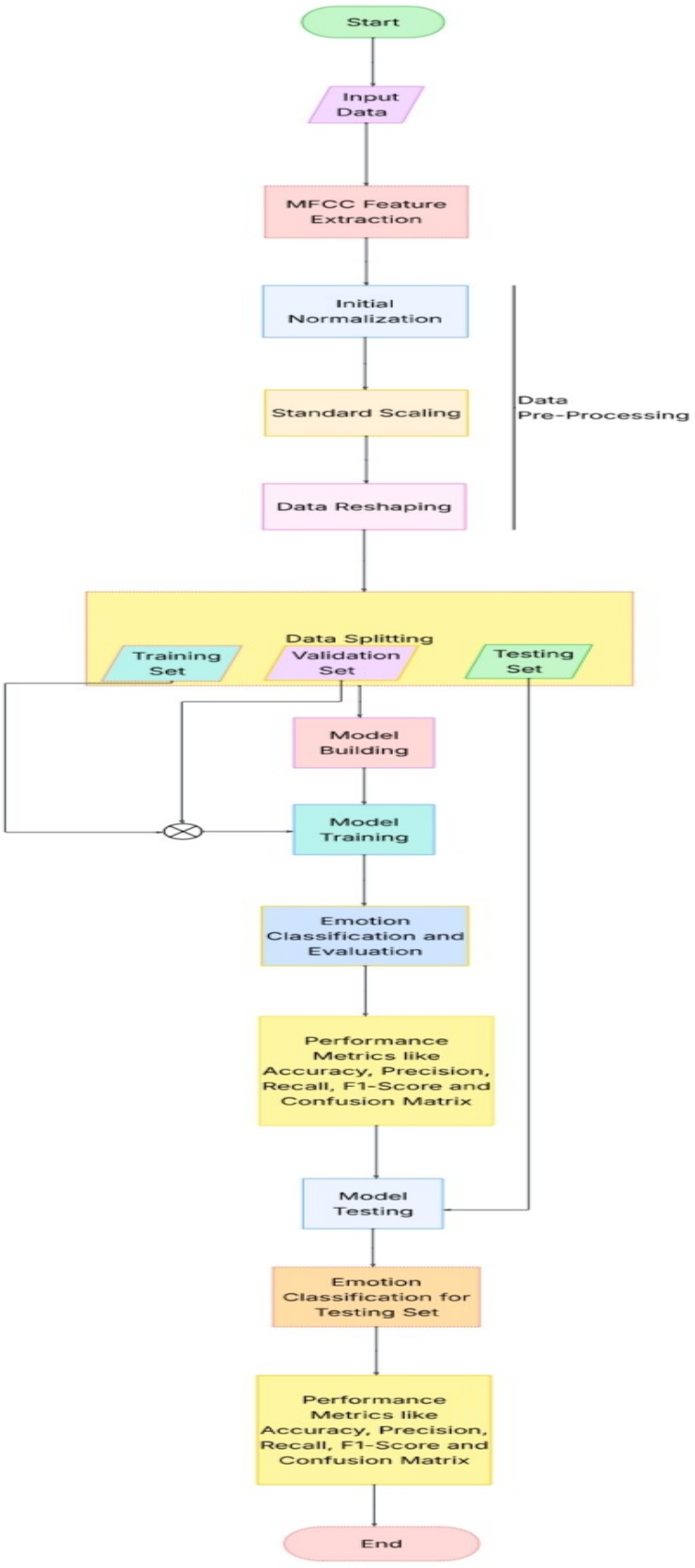



Figure 3.1: Flow Chart

## 3.2 Data Acquisition

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 organised 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.

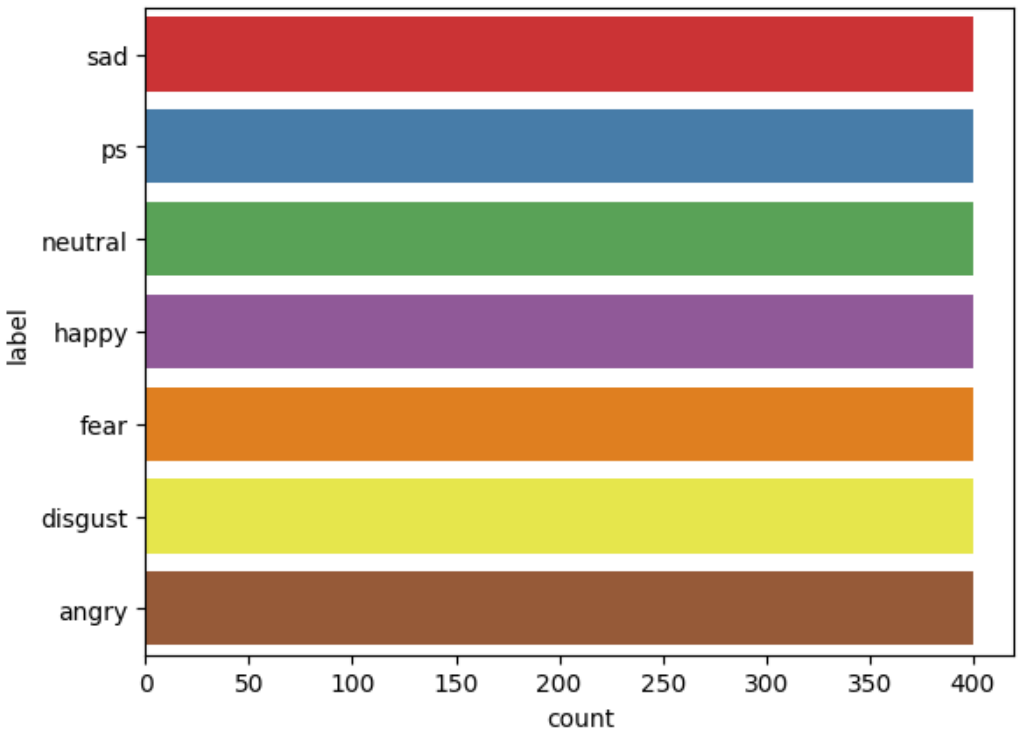



Figure 3.2: Distribution of TESS Dataset

## 3.3 Data Loading and Pre-processing

The initial phase of the research involved meticulous data preparation, commencing with the importation of crucial libraries. These libraries encompassed librosa for audio analysis, pandas for data manipulation, numpy for numerical computations, and matplotlib for visualization. Subsequently, audio files representing diverse emotions were loaded from a designated directory, their file paths and corresponding labels meticulously stored within

a Pandas DataFrame named 'df'. This structured format facilitated efficient data handling and analysis. To gain preliminary insights into the dataset's composition, visualization techniques were employed. Specifically, seaborn and matplotlib were utilized to generate a countplot, visually depicting the distribution of emotions across the dataset. This provided valuable information regarding the prevalence of each emotional category. Further enhancing understanding, specialized functions named 'waveplot' and 'spectogram' were defined. These functions enabled the visualization of audio waveforms and spectrograms, respectively, offering a deeper comprehension of the underlying acoustic characteristics associated with each emotion. This initial data exploration and preprocessing laid a solid foundation for subsequent feature extraction and model development stages.

## 3.4 MFCC Feature Extraction

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:

- 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.
- 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.
- 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.
- 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.

$$mel(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right)$$

- 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.
- 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.

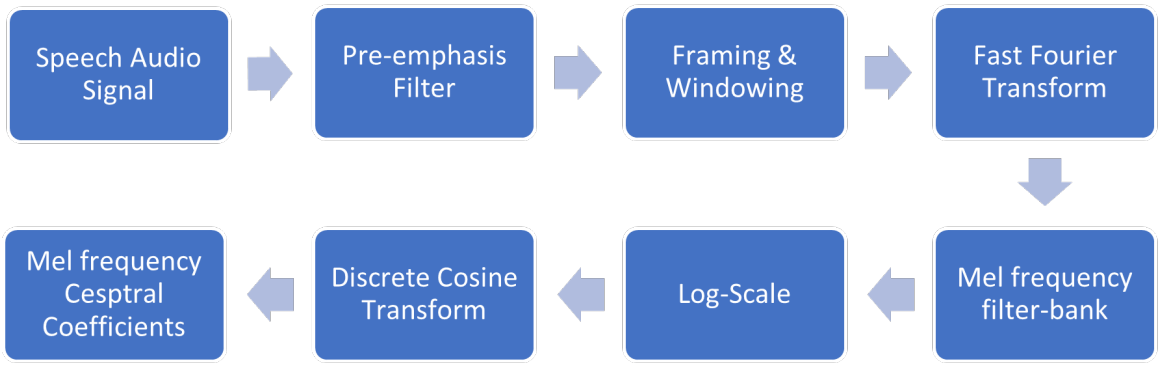



Figure 3.3: Illustration of MFCC Feature Extraction

The code extracts Mel-Frequency Cepstral Coefficients (MFCCs), representing the spectral envelope of audio signals crucial for human perception. Using librosa, audio files are loaded and processed to calculate 40 MFCCs per frame. These frame-level MFCCs are then averaged to produce a single feature vector for each audio file, capturing overall spectral characteristics. To ensure optimal performance of machine learning algorithms, feature scaling and normalization are applied. librosa.util.normalize normalizes each MFCC frame to zero mean and unit variance, reducing the influence of varying recording conditions. Additionally, StandardScaler from sklearn is employed to scale the entire feature set, further enhancing model training by ensuring all features contribute equally. This meticulous pre-

processing prepares the MFCC features for effective use in subsequent emotion recognition tasks.

To leverage the sequential nature of audio data, the extracted MFCC features are meticulously prepared for use with a Long Short-Term Memory (LSTM) network. This involves reshaping the feature matrix into a three-dimensional structure with dimensions. This format enables the LSTM to effectively process the temporal dependencies within the audio data. Furthermore, the categorical emotion labels are transformed into a numerical representation suitable for model training using one-hot encoding. OneHotEncoder from sklearn is employed to convert each emotion label into a binary vector where only one element is 'hot' (1), indicating the corresponding emotion. This transformation ensures that the model can interpret and learn from the categorical labels effectively. These crucial preprocessing steps ensure the data is optimally formatted for LSTM input and facilitate efficient model training for emotion recognition.

## 3.5 Model Building and Training

The core of this project revolves around constructing and training a deep learning model capable of classifying emotions from speech audio. Following the essential preprocessing steps, including the extraction and scaling of Mel-Frequency Cepstral Coefficients (MFCCs), the data was meticulously prepared for the neural network. The scaled MFCCs, initially representing the spectral characteristics of the audio, were reshaped to introduce a temporal dimension. This transformation was vital for utilizing a recurrent neural network (RNN), specifically an LSTM, which excels at processing sequential data like audio. The reshaping converted the data into a 3D format, allowing the model to analyze the evolution of features over time within each audio sample, thereby capturing temporal dependencies crucial for emotion recognition.

The model architecture was defined using TensorFlow's Keras API, a user-friendly framework for building deep learning models. Given the sequential nature of the input data, an LSTM layer formed the foundational component. LSTMs are particularly effective for time-series data as they can maintain internal memory and process variable-length

sequences, making them well-suited for audio analysis. Following the LSTM layer, several Dropout layers were incorporated. Dropout is a regularization technique that randomly sets a fraction of input units to zero at each training step. This helps prevent overfitting by forcing the network to learn more robust features that are not reliant on any single input or hidden unit. The model also included multiple Dense layers with ReLU activation functions. These fully connected layers perform non-linear transformations on the data, enabling the model to learn complex patterns and relationships between the MFCC features and the emotion labels. L2 kernel regularization was applied to the Dense layers. L2 regularization adds a penalty proportional to the square of the weights to the loss function, discouraging large weights and promoting a simpler model, which can also help in preventing overfitting. The final layer was a Dense layer with a softmax activation function. Softmax is a standard choice for the output layer in multi-class classification problems; it converts the raw output scores into a probability distribution over the possible classes, where the sum of probabilities for all classes equals one. The number of units in this output layer matched the total number of distinct emotion categories in the dataset, allowing the model to predict the likelihood of each emotion.

For efficient and effective training, the model was compiled with the Adam optimizer and the categorical crossentropy loss function. Adam is an adaptive learning rate optimization algorithm that is widely used and often performs well. Categorical crossentropy is the standard loss function for multi-class classification when the labels are in a one-hot encoded format, as they were in this project. To prevent overfitting and optimize the training process, an Early Stopping callback was implemented. This callback monitored the validation loss and halted training if the loss did not improve for a specified number of epochs, restoring the model weights from the epoch with the lowest validation loss. The data was split into training, validation, and test sets using a stratified approach to ensure that the distribution of emotion classes was consistent across all sets. The training data was used to update the model's parameters through backpropagation, the validation data was used to monitor performance during training and guide the early stopping, and the test data was held out for a final, unbiased evaluation. The model was trained by iterating over the training data

in batches, and its performance was tracked on the validation set with the early stopping callback ensuring that training concluded at the optimal point to avoid overfitting. Finally, the trained model was saved to a file for future use.

## 3.6 Evaluation and Visualization

Following the training phase, a critical step in the machine learning workflow is evaluating the model's performance and visualizing the results to gain insights into its strengths and weaknesses. The provided code meticulously addresses this through a series of evaluation metrics and visualizations. Initially, the trained model is evaluated directly on the validation set using model.evaluate. This provides a quick assessment of the model's performance on data it hasn't explicitly trained on, giving a measure of its generalization ability during the training process itself. The output typically includes the loss and any specified metrics, such as accuracy, calculated on the validation data.

To understand the training progress and identify potential issues like overfitting or underfitting, the code visualizes the training history. It extracts the training and validation accuracy and loss values recorded during each epoch of training. These values are then plotted against the number of epochs. A plot showing training accuracy increasing while validation accuracy plateaus or decreases, or training loss decreasing while validation loss increases, is a strong indicator of overfitting. Conversely, if both training and validation metrics are poor, it might suggest underfitting. These plots provide a clear visual representation of how well the model is learning and generalizing over time, guiding decisions about hyperparameters or model architecture.

Moving beyond simple accuracy, the code delves into more detailed performance analysis using classification reports and confusion matrices. The model makes predictions on the validation set, and these predictions are then compared to the true labels. A classification report, generated using sklearn.metrics.classification.report, provides a comprehensive breakdown of performance for each class. It includes precision (the proportion of correctly predicted positive instances among all instances predicted as positive), recall (the proportion of correctly predicted positive instances among all actual positive instances), F1-score

(the harmonic mean of precision and recall, providing a balance between the two), and support (the number of actual instances for each class). This report is invaluable for identifying classes where the model performs well or poorly.

Complementing the classification report, a confusion matrix is generated and visualized. The confusion matrix, created using sklearn.metrics.confusion.matrix, is a table that summarizes the performance of a classification model. Each row represents the instances in an actual class, while each column represents the instances in a predicted class. The entries in the matrix indicate the number of instances that fall into each category. Visualizing the confusion matrix using seaborn.heatmap makes it easy to see how well the model distinguishes between different classes. The diagonal elements represent correctly classified instances, while off-diagonal elements indicate misclassifications. This visualization is particularly useful for identifying which classes are most often confused with each other. Finally, the model is loaded and evaluated on the separate test set using the same metrics and visualizations. Evaluating on the test set, which the model has never seen during training or validation, provides the most reliable estimate of the model's performance on unseen data and its readiness for deployment. The classification report and confusion matrix on the test set offer a final, unbiased assessment of the model's ability to generalize to new audio samples.

# 4

# IMPLEMENTATION OF LSTM MODEL

## 4.1 LSTM Overview

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.

A traditional RNN has a single hidden state that is passed through time, which can make it difficult for the network to learn long-term dependencies. LSTMs address this problem by introducing a memory cell, which is a container that can hold information for an extended period. LSTM networks are capable of learning long-term dependencies in sequential data, which makes them well-suited for tasks such as language translation, speech recognition.

In this study, after obtaining the features from the TESS, EMO-DB, RAVDESS datasets, the LSTM method is used for emotion classification. 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, EMO-DB, RAVDESS datasets for model training, evaluation and testing. Working with small or flat data can lead to overfitting problems in our model.

An LSTM recurrent unit tries to "remember" all the past knowledge that the network is seen so far and to "forget" irrelevant data. This is done by introducing different activation

function layers called “gates” for different purposes. Each LSTM recurrent unit also maintains a vector called the Internal Cell State which conceptually describes the information that was chosen to be retained by the previous LSTM recurrent unit.

LSTM networks are the most commonly used variation of Recurrent Neural Networks (RNNs). The critical component of the LSTM is the memory cell and the gates (including the forget gate but also the input gate), inner contents of the memory cell are modulated by the input gates and forget gates. Assuming that both of the segue he are closed, the contents of the memory cell will remain unmodified between one time-step and the next gradients gating structure allows information to be retained across many time-steps, and consequently also allows group that to flow across many time-steps [5]. This allows the LSTM model to overcome the vanishing gradient properly occurs with most Recurrent Neural Network models. The Figure 4.1 shows the LSTM Architecture.

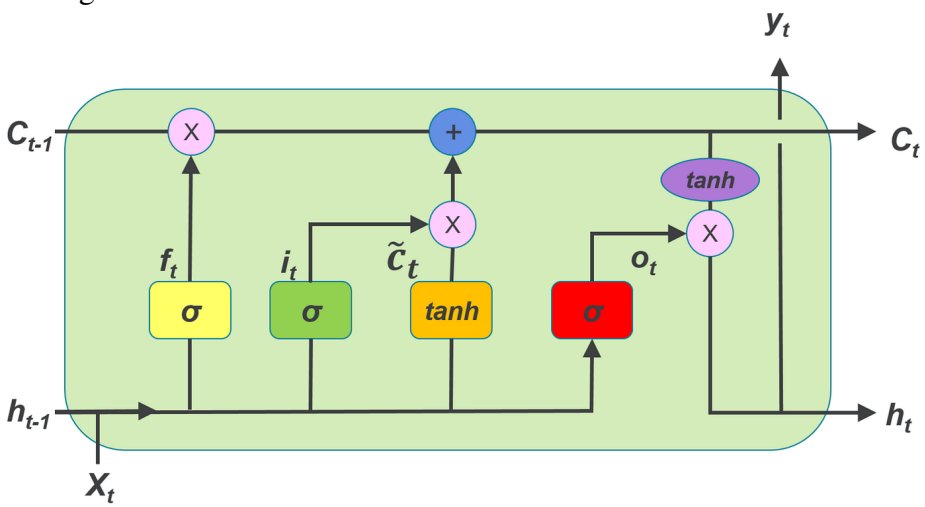



Figure 4.1: LSTM Cell Structure

This architecture is designed to capture complex temporal dependencies an extract complex patterns from the input speech data. LSTM networks are composed of memory cells and various gates that regulate the flow of information within the network.

- **Forget Gate**: This gate decides what information to discard from the cell’s memory. It takes the previous hidden state ($h_{t-1}$) and the current input ($x_t$) as input and outputs a value between 0 and 1 for each element in the cell state ($c_{t-1}$). A value of 0 indicates complete forgetting while 1 means complete retention.

  $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$

- **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 ($\hat{c}_t$) that could be added to the state.

  $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$

- **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.

  $\hat{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$

- **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.

  $c_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t$

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

  $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

- **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.

  $h_t = o_t \cdot \tanh(c_t)$

## 4.2 LSTM Model Architecture

The core of this audio classification task lies in the Long Short-Term Memory (LSTM) neural network architecture, specifically designed to process sequential data like audio features. The implemented model is a Sequential Keras model, meaning layers are stacked linearly. The first layer is the crucial LSTM layer. LSTMs are a type of recurrent neural network capable of learning long-term dependencies in sequences through the use of internal gates (input, forget, and output gates) and a cell state, which allows information to be carried across time steps. In this model, the LSTM layer takes the reshaped MFCC features as input, with an input_shape defined by the number of timesteps (40) and the number of features per timestep. The "return_sequences=False" argument indicates that this layer only outputs the hidden state of the last timestep, which summarizes the information processed throughout the sequence. Following the LSTM layer, Dropout layers are strategically placed. Dropout is a regularization technique that randomly sets a fraction of input units to zero at each training step. This prevents co-adaptation of neurons and helps the model generalize better to unseen data, reducing overfitting. The output of the LSTM layer is then fed into a series of Dense (fully connected) layers. These layers perform non-linear transformations on the features learned by the LSTM. The first two dense layers use the 'relu' (Rectified Linear Unit) activation function, which introduces non-linearity and helps the model learn complex patterns. These dense layers also incorporate kernel_regularizers.l2(0.01), adding L2 regularization to the weights. L2 regularization penalizes large

weights, further helping to prevent overfitting by making the model's decision boundary smoother. Another Dropout layer is included after the dense layers for additional regularization. Finally, the output layer is a Dense layer with 7 units, corresponding to the seven emotion classes in the dataset. It uses the 'softmax' activation function. Softmax converts the raw outputs into probability distributions over the classes, ensuring that the sum of probabilities for all classes equals 1. During compilation, the model uses the Adam optimizer, an adaptive learning rate optimization algorithm known for its efficiency. The loss function is 'categorical_crossentropy', appropriate for multi-class classification with one-hot encoded labels. The model is trained with an EarlyStopping callback, which monitors the validation loss and stops training early if it plateaus, preventing unnecessary computation and potential overfitting. The model's architecture, with its combination of LSTM for sequence processing, dense layers for higher-level feature learning, and regularization techniques like Dropout and L2 regularization, aims to build a robust classifier for audio emotion recognition.

| **Layer (type)** | **Output Shape** | **Parameters** |
|---|---|---|
| lstm (LSTM) | (None, 40, 256) | 264, 192 |
| dropout (Dropout | (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 |

Table 4.1: Three Layer LSTM Model Architecture

The model summary table 4.1 provides a concise overview of the neural network's architecture. The "Layer (type)" column identifies each layer by its name and type, such as LSTM for Long Short-Term Memory layers, Dropout for regularization layers that randomly set a fraction of input units to 0 at each update during training, and Dense for fully connected layers where each neuron is connected to all neurons in the previous layer. The

"Output Shape" column indicates the dimensions of the tensor output by each layer, with (None, ...) signifying a variable batch size. The subsequent numbers define the specific dimensions; for instance, in an LSTM layer, this might represent (batch_size, time_steps, features). The "Parameters" column details the number of trainable parameters such as weights and biases within each layer that the model adjusts during training. Finally, the table summarizes the "Total parameters," "Trainable parameters," and "Non-trainable parameters" for the entire model, differentiating between parameters learned during training and those that remain fixed. This summary is crucial for understanding the model's structure and computational complexity.

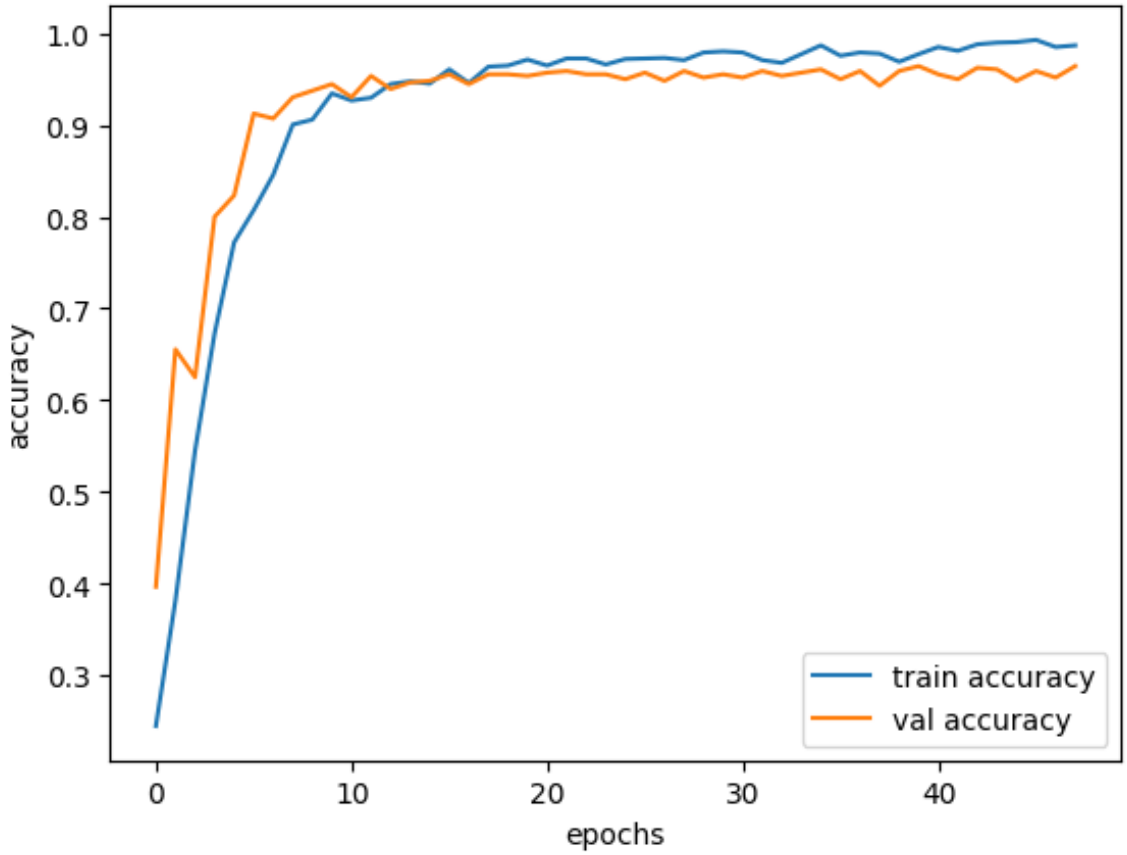



Figure 4.2: Plot of Training Accuracy vs Validation Accuracy

The figure 4.2 accuracy plot visualizes the model's performance during training by showing the accuracy on both the training and validation datasets across epochs. The "train accuracy" line typically shows how well the model is learning to classify the training data, and it generally increases over epochs as the model adjusts its parameters. The "val accuracy" line, representing the accuracy on the unseen validation data, is a crucial indicator of

how well the model generalizes to new examples. Ideally, the validation accuracy should track closely with the training accuracy and also increase, indicating that the model is not just memorizing the training data but is learning generalizable patterns. If the training accuracy continues to rise significantly while the validation accuracy plateaus or decreases, it suggests overfitting.

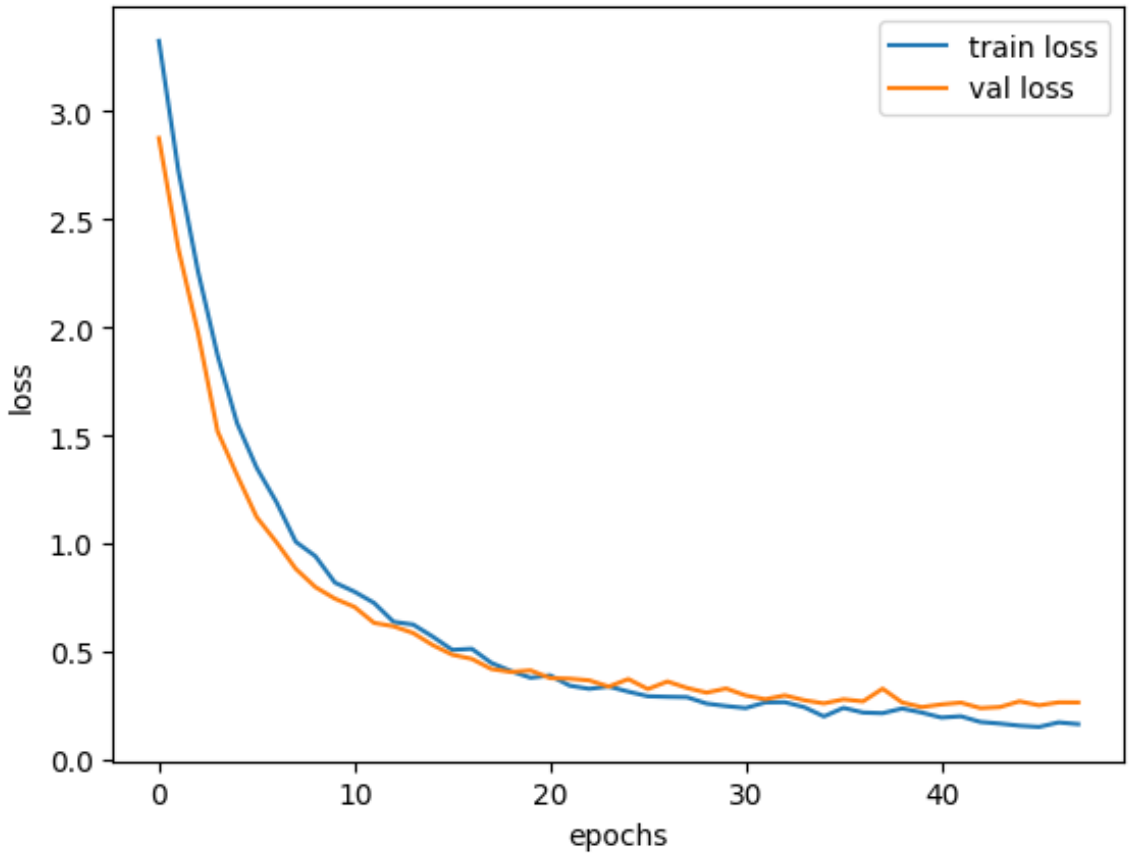



Figure 4.3: Plot of Training Loss vs Validation Loss

Similarly, the figure 4.3 loss plot displays the value of the loss function for both the training and validation datasets over epochs. The loss function quantifies the error between the model's predictions and the true labels, so a decreasing loss indicates that the model is improving. The "train loss" line shows the error on the training data, and the "val loss" line shows the error on the validation data. Like the accuracy plot, the validation loss is essential for detecting overfitting; if the training loss continues to decrease while the validation loss starts to increase, it's a strong sign that the model is overfitting to the training data. These plots together provide valuable insights into the training process, helping to determine if the model is learning effectively, if it is overfitting, and when to stop training.

# 5

# SOFTWARE REQUIREMENTS

Google Colab, in essence, is a free and accessible cloud-based platform that provides a Jupyter Notebook environment specifically tailored for data science, machine learning, and deep learning tasks. It eliminates the need for local setup by offering pre-configured environments with popular libraries like TensorFlow, PyTorch, and scikit-learn readily available. A significant advantage of Colab is its provision of free access to powerful hardware accelerators such as GPUs and TPUs, which are crucial for accelerating the training of computationally intensive deep learning models. This allows researchers, students, and practitioners to experiment with and develop complex models without requiring expensive local hardware. Moreover, Colab facilitates collaboration by allowing users to share their notebooks and work simultaneously on projects, making it an ideal tool for team-based deep learning endeavors. Its seamless integration with Google Drive enables easy storage and retrieval of datasets, models, and notebooks, streamlining the entire deep learning workflow from data ingestion to model deployment.

The applications of deep learning models, empowered by platforms like Google Colab, span a remarkable array of fields, revolutionizing how we interact with technology and solve complex problems. In computer vision, deep learning fuels advancements in image recognition, object detection and image generation, impacting areas like autonomous driving, medical imaging analysis, and security surveillance. Natural language processing has been transformed by deep learning, leading to breakthroughs in machine translation, sentiment analysis, text generation and sophisticated chatbots. In the realm of healthcare, deep learning aids in drug discovery, disease diagnosis, and personalized medicine. Financial institutions leverage deep learning for fraud detection, risk assessment and algorithmic trading. Furthermore, deep learning powers recommendation systems that personalize user experiences on e-commerce platforms and content streaming services. Its ability to learn intricate patterns from vast amounts of data makes deep learning a driving force behind innovation across diverse sectors, with new applications constantly emerging as the field continues to evolve.

The main key aspects of using Google Colab specifically when working with deep learning models like LSTM networks:

- **Free Access to Powerful Hardware Acceleration**: This is a cornerstone of Colab's appeal for deep learning. Training LSTMs, especially on sequential data like text or time series, can be computationally intensive. Colab provides free access to GPUs (like NVIDIA Tesla K80, T4, P100) and even TPUs (Tensor Processing Units), which are specifically designed for accelerating matrix operations crucial for deep learning. This significantly reduces training times compared to using just a CPU on a local machine. You can easily switch between hardware accelerators within the notebook settings.

- **Pre-installed Deep Learning Libraries**: Colab comes with essential deep learning libraries such as TensorFlow and PyTorch, along with other crucial data science libraries like NumPy, Pandas, and scikit-learn, pre-installed and ready to use. This eliminates the often cumbersome process of setting up your environment and installing dependencies, allowing you to jump straight into building and training your LSTM models.

- **Seamless Integration with Google Drive**: Colab's tight integration with Google Drive provides a convenient way to manage your project files. You can easily mount your Google Drive within the Colab notebook, allowing you to access your datasets, save your trained models and store your notebooks directly in the cloud. This simplifies data management and ensures your work is easily accessible from anywhere.

- **Collaborative Environment**: Built on the Jupyter Notebook framework, Colab facilitates collaboration. You can easily share your notebooks with others, allowing for real-time co-editing and sharing of code, results and insights. This is particularly beneficial for team projects involving LSTM model development and experimentation.

- **Web-Based Interface**: Being entirely web-based, Colab requires no local installation. All you need is a web browser and a Google account to start working. This makes it highly accessible, regardless of your operating system or local computing resources.

- **Free Tier Limitations and Colab Pro/Pro+**: While the free tier of Colab is incredibly useful, it does have limitations in terms of GPU/TPU usage time, memory, and background execution. For more demanding LSTM training tasks or longer training sessions, Google offers paid subscriptions like Colab Pro and Pro+ which provide increased resources and longer runtimes.

- **Easy Data Handling**: Colab provides various ways to upload and handle data. You can upload files directly, connect to your Google Drive or even fetch data from URLs. This flexibility is crucial when working with diverse speech emotion datasets for your LSTM network.

- **Interactive Experimentation and Visualization**: The Jupyter Notebook interface allows for interactive experimentation. You can write and execute code cells, inspect variables and visualize results directly within the notebook. This iterative approach is essential for debugging and improving your LSTM model.

- **Sharing and Reproducibility**: Colab notebooks can be easily shared, and the code and environment are generally reproducible, making it easier to share your speech emotion recognition project and allow others to understand and run your work.

- **Community and Resources**: The Colab environment has a large and active community and there are numerous online resources, tutorials, and examples available, specifically for using deep learning libraries like TensorFlow and PyTorch within Colab for tasks like sequence modeling with LSTMs.

Google Colab provides a free, accessible, and powerful platform perfectly suited for developing and training deep learning models, including LSTMs. Its access to hardware acceleration, pre-installed libraries, seamless Google Drive integration, and interactive Jupyter Notebook interface significantly streamlines the workflow for researchers, students, and practitioners working with sequential data and complex LSTM architectures.

# 6

# RESULTS AND ANALYSIS

## 6.1 Experimental Setup

In this experiment, we use the Google Colab as the primary environment for data preprocessing, 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. To achieve the study's objectives, three experiments were conducted to classify the emotions of TESS, EMO-DB, RAVDESS Datasets. Pre-trained models are used for the datasets . First model is three layer stacked LSTM, second model is Bi-LSTM, and third model is single layer LSTM. Model performance was then evaluated using a testing dataset. The experiments were coded in Python, known for its readability and versatility, and executed on the Google Colab platform, which offers a diverse range of datasets for research purposes.

## 6.2 Hyperparameters

A specific set of hyperparameters was selected to optimize model performance. The training was conducted over 48 epochs with early stopping to prevent overfitting, monitoring validation loss and stopping if no improvement was observed after 20 epochs. While the default input size was [40, 1] for most models, the Three layer stacked LSTM model utilized an increased size of to assess performance benefits. The Adam optimizer with a standard learning rate facilitated stable training. Metrics such as categorical cross-entropy loss, accuracy, precision, recall and f1-score were used to evaluate model effectiveness. These hyperparameters were essential in achieving high-quality results. Table 6.1 presents the training hyperparameters and loss functions utilized.

| **Hyperparameter** | **Value** |
|---|---|
| Optimizer | Adam |
| Loss function | Categorical Cross-entropy |
| Metrics | Accuracy, Precision, Recall, F1-Score |
| Number of epochs | 48 |
| Batch size | 48 |

Table 6.1: Training hyperparameters and loss function for training

## 6.3 Performance Evaluation 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, EMO-DB, RAVDESS datasets.

## 6.4 Results & 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 6.2 and table 6.3. 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.

| **Emotion** | **Precision** | **Recall** | **F1-Score** | **Support** |
|---|---|---|---|---|
| Angry | 0.99 | 0.96 | 0.97 | 80 |
| Disgust | 0.98 | 0.99 | 0.98 | 80 |
| Fear | 0.96 | 0.99 | 0.98 | 80 |
| Happy | 0.97 | 0.91 | 0.94 | 80 |
| Neutral | 1.00 | 1.00 | 1.00 | 80 |
| Pleasant Surprise (ps) | 0.92 | 0.97 | 0.95 | 80 |
| Sad | 0.99 | 0.97 | 0.98 | 80 |
| Macro Average | 0.97 | 0.97 | 0.97 | 560 |
| Weighted Average | 0.97 | 0.97 | 0.97 | 560 |

Table 6.2: Performance metrics on Validation Set

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 97.17% on the validation data. The figure 6.1 shows the confusion matrix of validation set from the trained model.

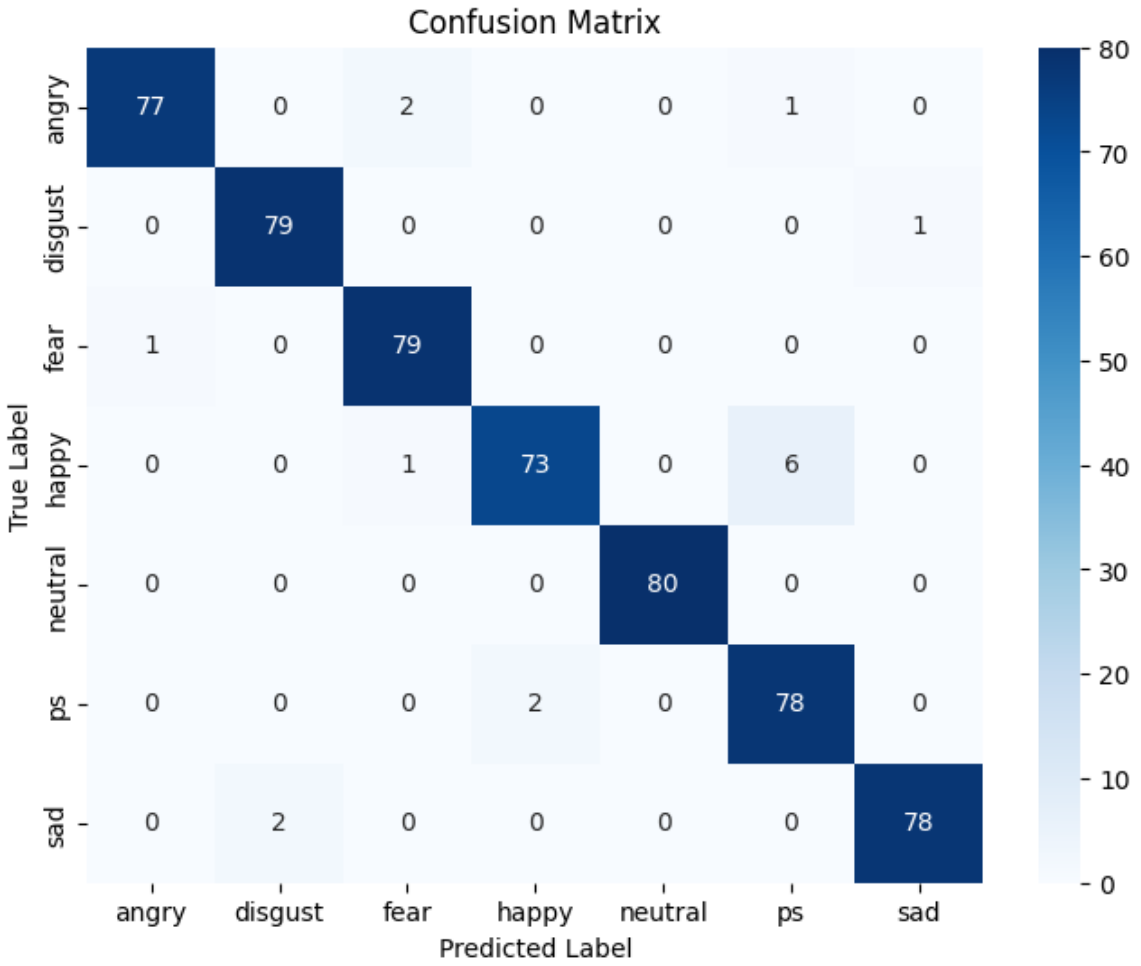



Figure 6.1: Confusion Matrix of Validation Set

The figure 6.1 is the confusion matrix for the validation set provides a detailed look at how well the model performed for each emotion class. It's a table where rows represent the true labels and columns represent the predicted labels. The values in the matrix show the number of instances where a true label was predicted as a certain class. For example, the cell at the intersection of 'true sad' (row) and 'predicted sad' (column) shows how many sad audio files were correctly classified as sad (True Positives for sad). The off-diagonal values indicate misclassifications. For instance, a value in the 'true sad' row and 'predicted happy' column would mean a sad audio file was incorrectly classified as happy (False Negative for sad, False Positive for happy).

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 6.3 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.

| **Emotion** | **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 |

Table 6.3: Performance metrics on Testing Set

The confusion matrix show below is the metric obtained for testing 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. Similar to the validation confusion matrix, the figure 6.2 confusion matrix for the testing set shows the counts of true versus predicted emotion labels for the test data. The diagonal entries represent the number of correctly classified instances for each emotion. For instance, the cell corresponding to 'true happy' and 'predicted happy' shows how many happy audio samples were correctly identified as happy in the test set. High values on the diagonal are desirable and indicate accurate predictions. The pleasant sad emotion class have 8 misclassifications 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%.

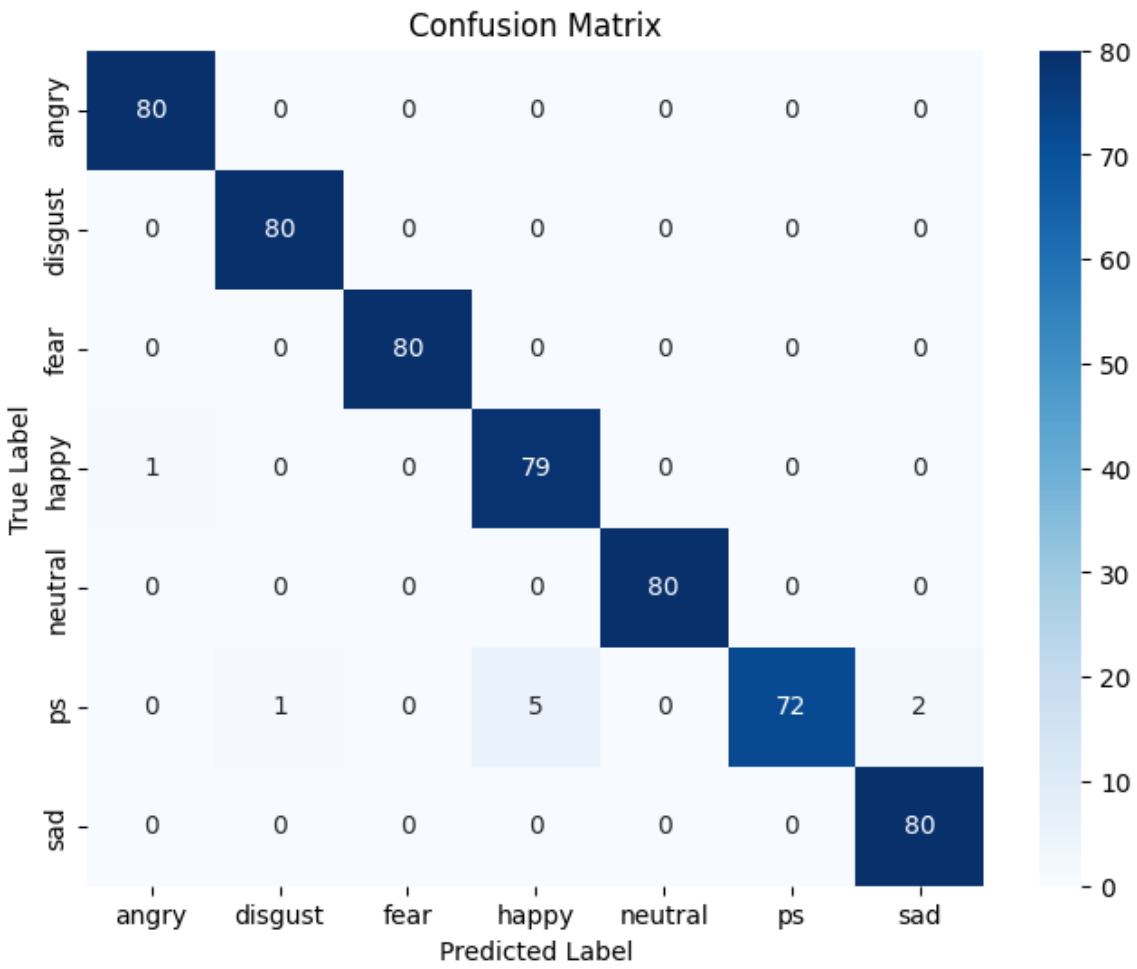



Figure 6.2: Confusion Matrix of Testing Set

## 6.5 Comparison of Proposed Model to Other Models

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.

The Table 6.4 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.

| **References** | **Features** | **Model** | **Accuracy** | **Comments** |
|---|---|---|---|---|
| Muralidharan et al [8] | Vocal Tract, Prosodic, Non-Linear | CNN+LSTM | 72.66% | Contribute to provide better models for emotion detection for a bot. |
| Beenaa Salian et al [10] | Mel Spectograms | CNN+LSTM | 89.23% | The optimizer for this model is Stochastic Gradient Descent (SGD). |
| Passricha et al [9] | MFCC | CNN+Bi-LSTM | 86.43% | Non-linearity with dropout idea shows 5.8% and 10% relative decrease in WER over CNN and DNN systems. |
| J. Parry et al [12] | Mel Filter Bank Coefficients | CNN<br>LSTM<br>CNN+LSTM | 89.33%<br>89.61%<br>93.80% | Performed on Cross Corpus and TESS corpus is out of domain test set. |
| Neha Prerna Tigga et al [4] | MFCC | CNN<br>LSTM<br>CNN+LSTM | 89.33%<br>89.61%<br>93.80% | A CuDNNLSTM layer was applied. |
| Faith Sengul et al [5] | MFCC | LSTM | 90% | Emotion Classification based on Speaker gender. |
| Testcase on EMO-DB | MFCC | Bi-LSTM | 90% | The sparse categorical cross-entropy loss function is used. |
| Testcase on RAVDESS | MFCC | Single Layer LSTM | 54% | The MFCC features are saved in xlsx file. |
| Proposed Model | MFCC | 3-Layer Stacked LSTM | 98.39% | The categorical cross entropy loss function is used. |

Table 6.4: Comparison of Proposed Model Performance with Other Model on TESS

# 7

# 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.

# 8

# References

# Bibliography

[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] 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.

[4] 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.

[5] 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.

[6] Muralidharan and Aditya Srinevas, “Emotion Detection for a Bot using CNN and LSTM ”in 2024.

[7] 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.

[8] 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.

[9] Beena Salian, Omkar Narvade, Rujuta Tambewagh and Smita Bharne, "Speech Emotion Recognition using Time-Distributed CNN and LSTM,"in ITM Web of Conferences, 2021.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

# 9

# Publications

# List of Publications

- “PEDDI AJAY KUMAR (Y21EC138), YERICHARLA SAYOMI (Y21EC196), RAMAVATH MALLIKHARJUNA NAIK(Y21EC146)”, “ENHANCING SPEECH EMOTION RECOGNITION USING LONG SHORT TERM MEMORY NETWORK”, 15th International Conference on Science and Innovative Engineering (ICSIE), OSIET, ISBN: 978-81-983498-5-9, April & 2025.

# 15TH INTERNATIONAL CONFERENCE ON SCIENCE & INNOVATIVE ENGINEERING - 2025

ORGANIZED BY

**PRINCE SHRI VENKATESHWARA PADMAVATHY ENGINEERING COLLEGE**

**PRINCE DR. K. VASUDEVAN COLLEGE OF ENGINEERING & TECHNOLOGY**

IN ASSOCIATION WITH

**MANIPAL UNIVERSITY COLLEGE, MALAYSIA**

**ORGANIZATION OF SCIENCE & INNOVATIVE ENGINEERING TECHNOLOGY**

## Certificate of Presentation

This is to certify that the paper entitled

Enhancing Speech Emotion Recognition using Long Short Term Memory Network

Authored by

Peddi Ajay Kumar,R.V.R. & J.C. College of Engineering, Chowdavaram

has been presented at

"15th International Conference on Science & Innovative Engineering" held on 26th & 27th April 2025 at Prince Dr. K. Vasudevan College Of Engineering & Technology , Chennai, India.

Dr. Antony V.Samrot, M.E., Ph.D.,
Director (Research, Innovation and Postgraduate Studies)
Manipal University College, Malaysia

Dr. A. Krishnamoorthy, M.E., Ph.D
Convener - ICCET

K.Janani, M.Tech.,
CEO, OSIET

# 15TH INTERNATIONAL CONFERENCE ON SCIENCE & INNOVATIVE ENGINEERING - 2025

ORGANIZED BY

**PRINCE SHRI VENKATESHWARA PADMAVATHY ENGINEERING COLLEGE**

**PRINCE DR. K. VASUDEVAN COLLEGE OF ENGINEERING & TECHNOLOGY**

IN ASSOCIATION WITH

**MANIPAL UNIVERSITY COLLEGE, MALAYSIA**

**ORGANIZATION OF SCIENCE & INNOVATIVE ENGINEERING TECHNOLOGY**

## Certificate of Presentation

This is to certify that the paper entitied

Enhancing Speech Emotion Recognition using Long Short Term Memory Network

Authored by

Yericharla Sayomi,R.V.R. & J.C. College of Engineering, Chowdavaram

has been presented at

"15th International Conference on Science & Innovative Engineering" held on 26th & 27th April 2025 at Prince Dr. K. Vasudevan College Of Engineering & Technology , Chennai, India.

Dr. Antony V.Samrot, M.E., Ph.D.,
Director (Research, Innovation and Postgraduate Studies)
Manipal University College, Malaysia

Dr. A. Krishnamoorthy, M.E., Ph.D
Convener - ICCET

K.Janani, M.Tech.,
CEO, OSIET

# 15TH INTERNATIONAL CONFERENCE ON SCIENCE & INNOVATIVE ENGINEERING - 2025

ORGANIZED BY

**PRINCE SHRI VENKATESHWARA PADMAVATHY ENGINEERING COLLEGE**

**PRINCE DR. K. VASUDEVAN COLLEGE OF ENGINEERING & TECHNOLOGY**

IN ASSOCIATION WITH

**MANIPAL UNIVERSITY COLLEGE, MALAYSIA**

**ORGANIZATION OF SCIENCE & INNOVATIVE ENGINEERING TECHNOLOGY**

## Certificate of Presentation

This is to certify that the paper entitied

Enhancing Speech Emotion Recognition using Long Short Term Memory Network

Authored by

Ramavath Mallikharjuna Naik,R.V.R. & J.C. College of Engineering, Chowdavaram

has been presented at

"15th International Conference on Science & Innovative Engineering" held on 26th & 27th April 2025 at Prince Dr. K. Vasudevan College Of Engineering & Technology , Chennai, India.

Dr. Antony V.Samrot, M.E., Ph.D.,
Director (Research, Innovation and Postgraduate Studies)
Manipal University College, Malaysia

Dr. A. Krishnamoorthy, M.E., Ph.D
Convener - ICCET

K.Janani, M.Tech.,
CEO, OSIET

# ICSIE 2025

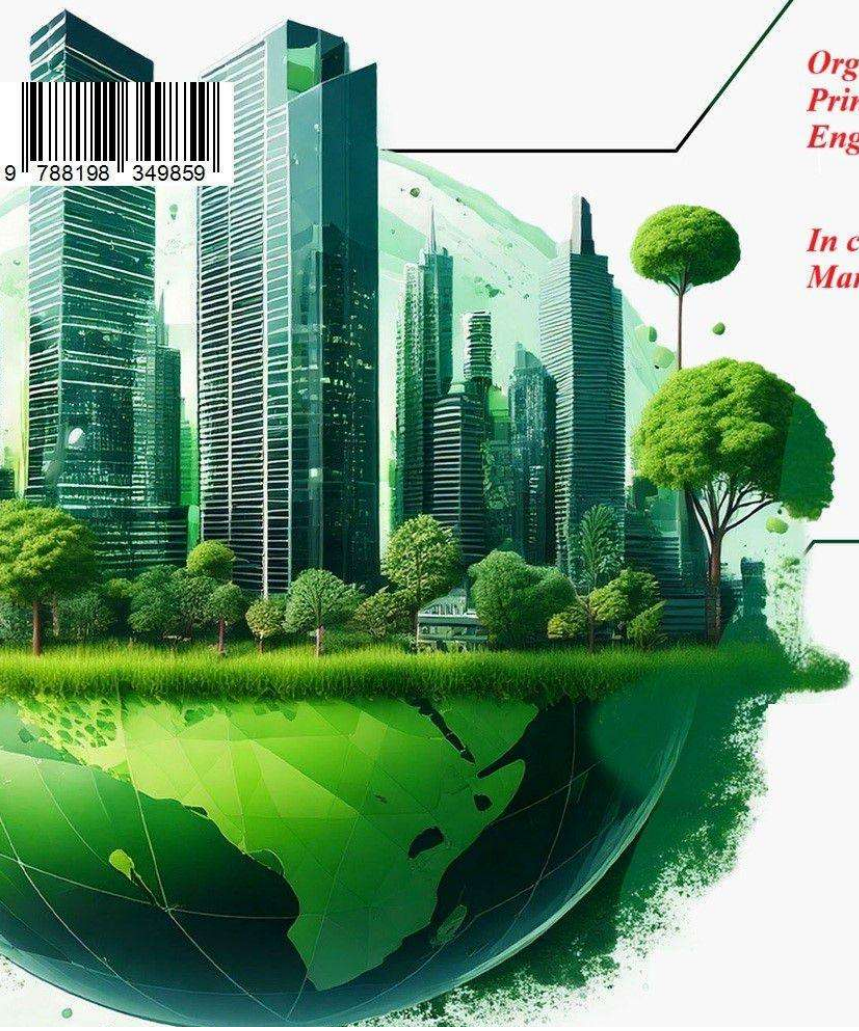
*Proceedings of*

**15TH INTERNATIONAL CONFERENCE ON SCIENCE AND INNOVATIVE ENGINEERING**

**15 ICSIE 2025**

**April 26th and 27th 2025**

**ISBN 978-81-983498-5-9**



*Organised by*
*Prince Dr.K.Vasudevan college of Engineering and Technology, India.*

*In collaboration with*
*Manipal University College, Malaysia.*

| | | |
|---|---|---|
| 346 | ICSIE251453 | A DIGITAL RECOMMENDATION SYSTEM FOR PERSONALIZED LEARNING TO ENHANCE ONLINE EDUCATION |
| 347 | ICSIE251633 | BATTERY AGEING PREDICTION IN ELECTRIC VEHICLES USING HYBRID AI MODEL |
| 348 | ICSIE251635 | AN IMPROVED DOUBLE FUZZY BASED PHYSICAL UNCLONABLE FUNCTION FOR SECURED KEY ENCRYPTION AND AUTHENTICATION IN IOT APPLICATIONS: DFPUS_IOT |
| 349 | ICSIE251662 | ENHANCING SPEECH EMOTION RECOGNITION USING LONG SHORT TERM MEMORY NETWORK |
| 350 | ICSIE251636 | SMART CURVE CRASH AVOIDANCE USING 2CAP IN VANETS |
| 351 | ICSIE251498 | TRAVEL EASE : PERSONALISED TRAVEL PLANNING SYSTEM USING AI |
| 352 | ICSIE251691 | AI-DRIVEN SOLUTION FOR DETECTING AND PREVENTING PADDY LEAF DISEASES IN AGRICULTURE |
| 353 | ICSIE251845 | DEVELOPMENT OF INFANT SLEEPING BAG WITH INTEGRATED POLYIMIDE LIG HEATER PAD AND DIAPER WETNESS DETECTION USING LIG-INTERDIGITATED ELECTRODES |
| 354 | ICSIE251840 | EMPOWERING AGRICULTURE PRODUCTS WITH QR CODE VERIFICATION AND SENTIMENT ANALYSIS |
| 355 | ICSIE250482 | AN APPLICATION TO ENHANCE AND RESTORE OLD OR DAMAGED PHOTOGRAPHS |
| 356 | ICSIE251590 | DESIGN AND SIMULATION OF EXHAUST MUFFLER FOR KTM 390 SINGLE CYLINDER ENGINE |
| 357 | ICSIE251728 | DATA BREACH MONITORING SYSTEM |
| 358 | ICSIE251962 | ADVANCED BITCOIN PREDICTION USING SOCIAL MEDIA INFLUENCER: REAL TIME APPLICATION USING ISTM, FLASK AND BOOSTING ALGORITHM |
| 359 | ICSIE251575 | RED TIDE ALGAL BLOOM IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS |
| 360 | ICSIE251451 | ANTI FRAUD MODEL FOR INTERNET LOAN DETECTION USING CONVOLUTIONAL NETWORKS |
| 361 | ICSIE251713 | AUTOMATED AADHAAR AND SMART CARD VERIFICATION SYSTEM USING OCR, QR CODE SCANNING AND FACE RECOGNITION FOR GOVERNMENT LOAN WAIVERS |
| 362 | ICSIE251518 | AMBULANCE TRACKING AND RESPONDS SYSTEM |
| 363 | ICSIE251614 | DEVELOPMENT OF SOLAR STILL WITH FRESENAL LENS AND CONDENSOR |
| 364 | ICSIE251418 | ADAPTIVE DIGITAL TWIN OF AN INDUCTION MOTOR USING MACHINE LEARNING AND MIXED REALITY |
| 365 | ICSIE251454 | HELPMATE-A WOMEN SAFETY DEVICE USING IOT AND MACHINE LEARNING |
| 366 | ICSIE251782 | ANALYSIS AND COMPARISON OF VARIOUS POWER CONVERTER |
| 367 | ICSIE251710 | INTEGRATING NETWORK ANALYSIS AND ML FOR SUPERIOR FINANCIAL RISK PREDICTION |
| 368 | ICSIE251500 | LEVERAGING TRANSFER LEARNING TO ANALYSE SOFTWARE DEVELOPERS OPINIONS FOR ENHANCED PROJECT INSIGHTS |
| 369 | ICSIE251724 | CLOUD RESOURCE MONITOR FOR MISCONFIGURATION DETECTION |
| 370 | ICSIE251583 | TOWARDS SAFER SOCIAL MEDIA: REAL-TIME EUPHEMISM AND TOXICITY DETECTION IN ONLINE USER INTERACTIONS |
| 371 | ICSIE251715 | CORRUGATION GEOMETRY'S EFFECT ON HEAT TRANSMISSION AND PIPE FLOW CHARACTERISTICS |
| 372 | ICSIE251778 | ALGORITHMIC TRADING BOT |
| 373 | ICSIE251539 | DETECTING SPAM AND FAKE USERS ON SOCIAL MEDIA PLATFORM USING MACHINE LEARNING |
| 374 | ICSIE251773 | REMLINK: A UNIFIED PLATFORM FOR AUTOMATED CLASSIFICATION AND COLLABORATIVE SHARING OF WEB RESOURCES |

# 348. ENHANCING SPEECH EMOTION RECOGNITION USING LONG SHORT TERM MEMORY NETWORK

Peddi Ajay Kumar
Yericharla Sayomi
Ramavath Mallikharjuna Naik
Department of Electronics and Communication Engineering
R.V.R & J.C College of Engineering
Guntur, India

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.

 *ISBN* *978-81-983498-5-9*