Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Speech Emotion Recognition

By:

Marwan Salah Mohamed Mahmoud	[Computer Science]
Alaa Adel Darwish Hussein	[Computer Science]
Abdelrahman Alaa Hamouda Mohamed	[Computer Science]
Abdelrahman Amr Mohammed Ahmad	[Computer Science]
Abdelaziz Gamal Ali Mohamed	[Computer Science]

Under Supervision of:

Dr. Sally Saad,
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

TA. Samar Aly,
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

July 2023

Table of Contents

Acknowledgement

We would like to acknowledge and express our gratitude to Dr. Sally Saad and TA Samar Aly from the Computer Science Department at Ain Shams University. Their guidance and support have been invaluable throughout our work, and we are truly grateful for their expertise and dedication.

Additionally, we extend our thanks to the Computer Science Department and the Faculty of Computer and Information Sciences for providing a conducive academic environment.

Finally, we appreciate the support of our peers and colleagues who have contributed to our growth and learning.

Abstract

Speech Emotion Recognition (SER) is a growing field that focuses on processing and classifying speech signals to detect underlying emotions. The significance of Deep Learning (DL) techniques in this domain cannot be understated. DL models, such as Long Short-Term Memory (LSTM) networks, have proven to be highly effective in capturing the complex patterns and temporal dependencies present in speech data. Recognizing and understanding emotions in speech is vital for enabling effective human-machine interactions. Emotion recognition holds significant potential for various applications, including online marketing, shopping assistance, and Human-Computer interaction (HCI) with voice assistants like Siri. By leveraging DL techniques and utilizing diverse datasets, this research contributes to the development of robust SER systems, further facilitating seamless communication between humans and machines.

In this study, preprocessing techniques including silence removal and scaling were applied to enhance the quality of the input data. The datasets utilized in the developed SER system were Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) and Toronto emotional speech set (TESS), which provide a diverse range of emotional speech samples for training and evaluation.

The performance of the implemented DL models was assessed using testing and training accuracy metrics. In the developed SER system, multiple DL models have been utilized, such as the CUDA-Deep Neural Network-LSTM (CU-DNN-LSTM) model and the Bidirectional LSTM (B-LSTM) model. The CU-DNN-LSTM model achieved a testing accuracy of 91% and a training accuracy of 97%. Similarly, the B-LSTM model achieved a testing accuracy of 92.54% and a training accuracy of 98%. These results demonstrate the effectiveness of the employed DL models in accurately recognizing and classifying emotions in speech signals.

List of tables

List of Figures

List of Abbreviations

Abbreviation	Full Form
APIs	Application Programming Interfaces
ANNs	Artificial Neural Networks
BLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
CU-DNN-LSTM	CUDA-Deep Neural Network-LSTM
Chroma-STFT	Chroma Short-Time Fourier Transform
DL	Deep Learning
DNNs	Deep Neural Networks
Emo-DB	Emotional Database
ETL	Extract, Transform, and Load
EDA	Exploratory Data Analysis
FF	Fundamental Frequency
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HLDs	High-Level Descriptors
HNR	Harmonic to Noise Rate
IDE	Integrated Development Environment
IDEs	Integrated Development Environments
GPUs	Graphics Processing Units
LSTM	Long Short-Term Memory
LLDs	Low-Level Descriptors
MFCC	Mel Frequency Cepstral Coefficients
MFFC	Mil Frequency Fluctuation Coefficient
ML	Machine Learning
MLP	Multilayer Perceptron
NumPy	Numerical Python
OS	Operating System
Pandas	Python Data Analysis Library
PDREC	Persian Drama Radio Emotional Corpus
RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Songs
RMSE	Root Mean Square Energy
RNN	Recurrent Neural Networks
RML	Ryerson Multimedia Laboratory
SAVEE	Surrey Audio-Visual Expressed Emotion
SER	Speech Emotion Recognition
SVM	Support Vector Machine
SHEMO	Sharif Emotional Speech Database
Sklearn	Scikit-Learn
SciPy	Scientific Python
TESS	Toronto emotional speech set
TEO	Teager Energy Operator
TPUs	Tensor Processing Units
UI	User Interface
ZCR	Zero Crossing Rate

Chapter 1

Introduction

Chapter 1

Introduction

Speech is the most natural way of communication between humans, and it contains information about the characteristics of the speaker, such as gender, emotional state, etc. HCI has increased with the development of technology, so SER is one of the most advanced areas for its improvement. There is a gap between the physical world and the digital world so SER aims to give machines the ability to feel like a human and solving the difficulty of human-machine communication to improve human-machine interaction [1].

Recently, researchers have studied various methods such as DL models to increase the efficiency of SER. DL in neural networks has achieved tremendous success in various domains that led to multiple DL architectures emerging as effective models across numerous tasks. These are some of the types of DL models widely used in various fields, including feed-forward architectures such as Deep Neural Networks (DNNs) and CNNs, which have been particularly successful in image and video processing as well as speech recognition. Additionally, recurrent architectures such as Recurrent Neural Networks (RNNs) and LSTM have been highly effective in speech recognition [2].

The features that can be extracted from a voice signal, such as Mel Frequency Cepstral Coefficients (MFCC), pitch features, and auditory speech, have been used in the developed SER system. These features are commonly employed in SER systems to capture relevant information from the speech signal and represent it in a manner suitable for analysis and classification. By leveraging these features, the system can effectively extract valuable characteristics from the voice signal, facilitating accurate recognition and classification of emotions [3]. Researchers work with audio signals by treating them as time-series data or using spectrograms to generate numeric and image forms of the audio. Both feature extraction techniques, including MFCC, pitch features, and auditory speech, as well as time series analysis, require applying a range of transformations to the original data to effectively represent the signal. There are several databases used in SER field such as RAVDESS, Emotional Database (EMO-DB), Surrey Audio-Visual Expressed Emotion (SAVEE) and Persian Drama Radio Emotional Corpus (PDREC) to train and evaluate the used DL model [4]. SER has been used in many fields, such as online marketing.

1.1 Motivation

Humans can easily identify the emotion of a speaker, but the field of emotion recognition through artificial intelligence is an open research area and there is still a need to make Machine Learning (ML) models robust in SER. Moreover, adding emotions to machines has been recognized as a critical factor in making machines appear and act human-like. Verbal sentiment recognition can be used in various audio recordings, job interviews, caller agent calls, video broadcasts, music recommendation, and rating systems. With further analysis, we can better understand people's motivations, whether they are delighted customers or not.

1.2 Problem Definition

The ability to recognize human emotions is a very difficult task and continues to be a subject of ongoing research. Even for humans, it is difficult to capture emotion in ordinary speech, no matter what the meaning. SER is a system that can recognize human emotions and any emotion from audio samples and refers to the process by which the computer analyzes the signal collected from the sensor to obtain the state of the emotion.

1.3 Objective

- Build a robust SER model to improve HCI.
- Extract voice from data source without any corruption.
- Extract important sound wave features such as Fundamental frequency, MFCC, etc.
- Emotion detection based on the selected features of the model.
- Improve efficiency of the developed SER model.
- Using the results in many applications such as: Marketing.
- Develop a real-time SER application for detecting emotions in real-time without any personal references.

1.4 Time Plan

The project is divided into four quarters throughout the year, as shown in Table 1.1 below. In the first quarter, the focus is on conducting surveys and specifying requirements. The second, third, and fourth quarters involve project analysis, design, testing, and documentation. This timeline ensures a systematic approach to the project's development and completion of each module within its designated quarter.

Table 1. 1 Time Plan

	Quarter one	Quarter two	Quarter three	Quarter four
Survey	■			
Requirement Specifications	■	■		
Project Analysis		■		
Project Design		■	■	■
Project Testing				■
Project Documentation		■	■	■

1.5 Document Organization

Chapter Two:
Chapter 2 explores the background of SER, examining previous work in the field and discussing the insights and benefits derived from relevant papers and projects.

Chapter Three:
Chapter 3 focuses on the development of a comprehensive plan for system analysis and design. This chapter outlines the steps involved, including data collection and the identification of non-functional and functional requirements for the system.

Chapter Four:
Chapter 4 delves into the implementation and testing phase of the SER system. It provides a detailed examination of the system's functionality, including the algorithms employed for feature extraction, feature selection, and DL methods.

Chapter Five:
Chapter 5 offers a user guide and highlights best practices for effectively utilizing the SER system. It provides instructions to users on how to maximize the benefits of the application.

Chapter Six:
Chapter 6 presents a summary of the project's key findings and conclusions. It also discusses potential future updates.

Chapter2

Background

Chapter 2

Background

The first paper on this topic was by “Daellert et al. 1996” [5]. However, the idea has been around for much longer, with the first patent dating back to the late 1970s using measurements of the autonomic nervous system for emotion recognition. Emotion detection from speech is a new field of research In HCI systems. SER System could provide users with improved services by being adaptive to their emotions. In virtual worlds, SER could help simulate more realistic avatar interaction. This background chapter provides a comprehensive overview of SER approaches. It covers both traditional techniques and advanced methods employed in the SER field. Additionally, it explores related works and studies conducted in the domain of SER.

2.1 SER Approaches

SER employs two main approaches: text analysis extracts emotional content from text, while signal processing analyzes acoustic properties of speech signals to infer emotional states.

- The first approach is text analysis, where textual data is associated with the speech, such as speech transcripts.
- The second approach is signal processing. This approach involves extracting acoustic features from the voice signal, such as MFCCs, pitch features, and auditory speech. These features provide valuable information for emotion detection and classification. As shown in Figure 2.1 the sound waves capture the mechanical vibrations of an object in the atmosphere, which then are translated into measurable signals that can change over time or space. This approach enables a deeper understanding of the emotional content expressed in the voice signal.

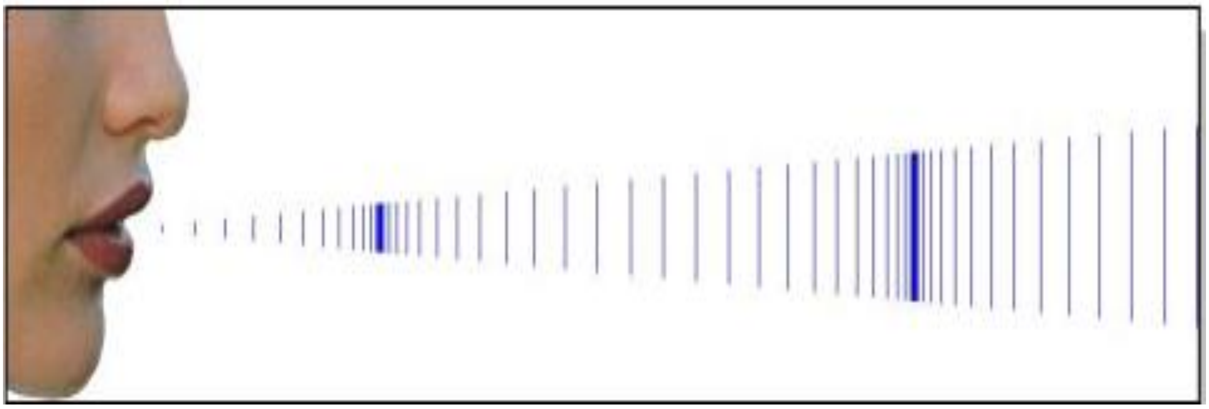


Figure 2. 1 Sound Wave

2.2 Traditional Techniques of SER

In the context of traditional emotion recognition techniques for digitized speech, the system typically comprises three fundamental components: feature extraction, feature selection, and classification.

- Feature extraction: acquiring features such as Low-Level Descriptors and high-Level descriptors.
- Feature selection: select the most important features in the wave.
- Feature classification: using ML models such as Support Vector Machine (SVM).

2.3 Advanced Techniques of SER

Scientists in SER systems have increasingly preferred DL models due to various reasons. These reasons stem from the unique characteristics and capabilities of DL models that make them well-suited for SER tasks.

2.3.1 The Importance of the DL techniques

DL is inspired by the way neurons in the human brain work and uses Artificial Neural Networks (ANNs) to replicate this process. As shown in Figure 2.2 DL models have multiple layers that enable them to learn complex patterns from raw data. They have revolutionized various fields by achieving superior performance in tasks such as SER and natural language processing. DL Advantages such as:

- DL models can recognize complex structures and features without needing to manually extract low-level features from the raw data.
- Classifying the samples after extracting the features.
- Dealing with unlabeled data.

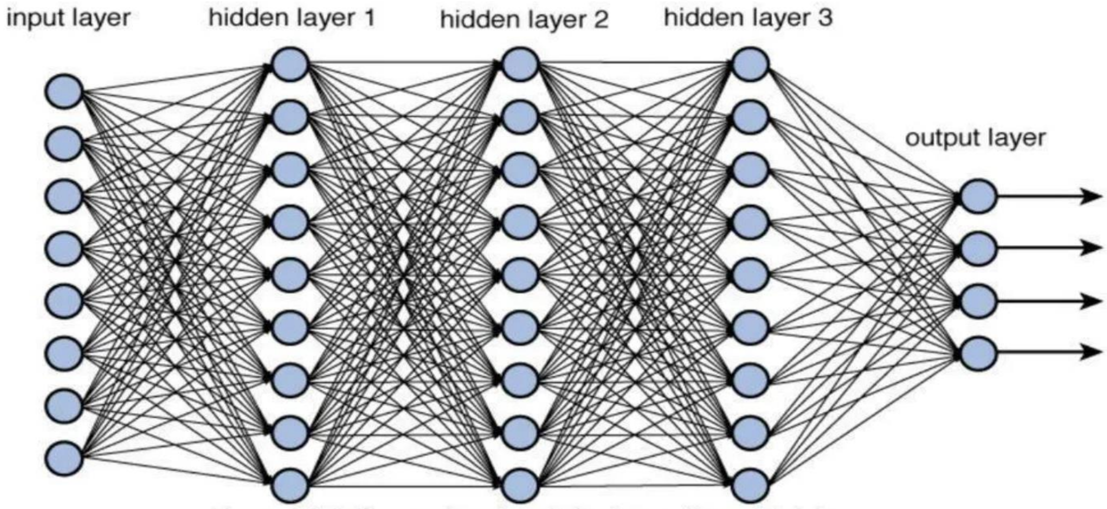


Figure 2. 2 DL Model General Architecture

2.4 Related works

The field of SER has gained significant attention in recent years, fueled by advancements in DL and natural language processing techniques. Researchers have explored various approaches to develop robust SER systems that can accurately identify and classify emotions conveyed through speech. This section provides a comprehensive overview of the related work in SER, highlighting the different methodologies employed, datasets utilized, and performance metrics assessed.

Yazdani, et al [6] conducted a study aiming to achieve a natural HCI by implementing a two-step approach. The first step involved feature extraction and the second step focused on feature classification. To perform these tasks, the researcher utilized the Sharif Emotional Speech Database (SHEMO) dataset and employed a range of DL and ML techniques. In the process, they incorporated signal features from both low- and high-level descriptions. The low-level descriptions encompassed pitch, voicing probability, frame energy, zero crossing rates, and MFCC. Meanwhile, the high-level descriptions included statistical measures such as mean, variance, minimum, maximum, median, quartiles, and higher-order moments. The evaluation of this approach yielded a weighted accuracy of **78.29%**.

Kanani, et al [7] presented a CNN architecture consisting of convolutional, pooling, and fully connected layers, aiming to solve a multi-class classification problem. The researchers employed an unweighted mean parameter to calculate the average for this classification task. The performance evaluation of the models was conducted using a standard confusion matrix, and from their experiments on the RADVESS dataset, the proposed CNN achieved an accuracy of **82.99%**.

Singh, et al [8] employed a process where audio files were played and audio features were extracted. These extracted characteristics were then converted into structured data frames. The researchers compared a loaded model using the prediction function, with a batch size of 32, to determine the expression or emotion present in the audio file. After training and evaluating multiple models, they achieved the highest accuracy of **82%** with a soft argument maximum activation layer, "rmsprop" optimizer, 18 layers, a batch size of 32, and a total of 1000 epochs.

Dong, et al [9] devised a novel approach for emotion classification, utilizing two parallel CNNs to extract spatial features, alongside a transformer encoder network to extract temporal features. By combining the strengths of CNNs in spatial feature representation and the transformer's ability in sequence encoding conversion, the model aimed to accurately classify emotions into eight distinct classes. Evaluation on the hold-out test set of the RAVDESS dataset demonstrated a promising accuracy of **80.46%**, highlighting the effectiveness of the proposed CNN and transformer architecture in emotion classification tasks.

Aouani, et al [10] proposed a two-stage approach for emotion recognition from speech signals. The first stage involved investigating two sets of features. The first set consisted of a 42-dimensional vector of audio features, including 39 coefficients of MFCC, Zero Crossing Rate (ZCR), Harmonic to Noise Rate (HNR), and Teager Energy Operator (TEO). The second set utilized an auto-encoder method to select relevant parameters from the previously extracted features. In the second stage, Support Vector Machines (SVM) were employed as the classifier method. The experiments conducted on the Ryerson Multimedia Laboratory (RML) dataset resulted in an accuracy of **74.07%**. These findings demonstrate the potential of the proposed system in effectively recognizing emotions from speech signals.

E Yu Shchetinin, et al [11] conducted an investigation into the architecture of DNN for the purpose of recognizing human emotions from speech. They utilized CNN and RNNs with LSTM memory cells as DNN models. Furthermore, the authors constructed an ensemble of neural networks based on these models. Through computer experiments, they evaluated the effectiveness of the proposed DNN models and compared them with basic ML algorithms for emotion recognition in human speech. The results revealed an accuracy of **86.2%**, highlighting the promising performance of the proposed DNN architectures in this domain.

Xiangmin Lun, et al [12] introduced a method for speech emotion recognition that involved extracting 64 statistical features from speech signals, such as short-term energy, pitch, frame, formant, and spectrum energy, utilizing a speech emotion database. To identify the most influential feature set, the authors employed mean Impact value and an improved Correlation-based Feature Selection technique. The accuracy of emotion recognition was evaluated using a back propagation Neural network. The proposed method, combining meaniImpact value and Correlation-based Feature Selection, successfully selected the features associated with speech emotion, resulting in reduced recognition errors. The achieved accuracy for the proposed approach was **91.15%**. These results highlight the effectiveness of the proposed feature selection method in improving the accuracy of speech emotion recognition.

Althaf Hussain Basha, et al [13] aimed to enhance the accuracy of speech emotion prediction by employing DL models. The researchers conducted experiments using the Multilayer Perceptron (MLP) and CNN classification models on three benchmark datasets comprising 5700 speech files categorized into seven emotion categories. Through their proposed model, they achieved improved accuracy in speech emotion prediction. The accuracy obtained for their approach was **89.01%**. These findings underscore the effectiveness of DL models in enhancing the accuracy of speech emotion prediction tasks.

Cristina Luna-Jiménez, et al [14] presented an automatic emotion recognizer system comprising a SER component and a facial emotion recognizer. For the SER, the researchers evaluated the performance of a pre-trained xlsr-Wav2Vec2.0 transformer using two transfer-learning techniques: embedding extraction and fine-tuning. The system achieved an accuracy of **81.82%**. These results highlight the effectiveness of the proposed approach in automatically recognizing emotions from speech, demonstrating the potential of pre-trained transformers and transfer learning techniques in improving SER accuracy.

Chapter 3

System Analysis and design

Chapter 3

System Analysis and Design

This chapter provides an overview of the SER system. It outlines the SER system architecture, including its components and interactions. The chapter also discusses the data layer, model layer, and application layer, explaining their functions within the system. Furthermore, it presents the distinct phases involved in the analysis and design process with each phase description.

3.1 SER System Overview

SER is a system that analyzes voice samples to recognize emotions by processing the collected signal from a sensor. In this section, the overview of the SER system will be presented, encompassing the SER system architecture, functional and non-functional requirements, SER system users, use case diagram, class diagram, sequence diagram, and database diagram. The SER system utilizes two datasets, namely RAVDESS and TESS. The SER system takes audio input from the user, preprocesses it, and extracts relevant features to classify the audio into different emotion categories.

3.1.1 SER System Architecture

The following architecture defines the structure of the software system and how it is organized. It also describes the relationships between components, levels of abstraction, and other aspects of the SER system. As shown in Figure 3.1, the SER system architecture consists of three main layers: the data layer, the model layer, and the application layer.

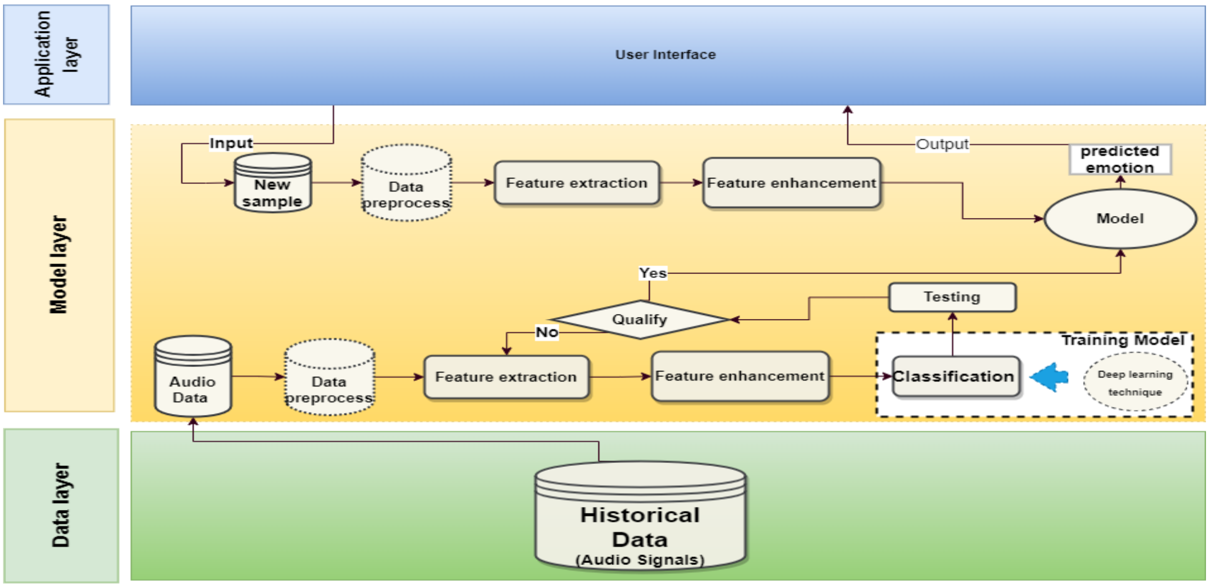


Figure 3. 1 SER System Architecture

3.1.1.1 Data Layer

The data layer has the RAVDESS dataset (audio signals) which will be used to train, validate, and test the model.

3.1.1.2 Model Layer

The model layer contains the DL algorithm, preprocessing, features extraction, load the dataset from the database for training, testing and get the user input voice for predicting his emotion then returns the result to the presentation layer.

3.1.1.3 Application Layer

The application layer handles the main programs of the SER architecture. It includes the code definitions and most basic functions of the developed application and controls the communication between model layer and the presentation layer.

3.1.2 SER Phases Description

The SER system consists of several essential phases. It begins with data storage, where speech datasets are collected and stored. The data then undergoes preprocessing to clean and normalize it. Feature extraction follows, capturing relevant characteristics of emotional speech. Data augmentation techniques are applied to enhance the training process, while feature enhancement techniques refine the extracted features. SER model training utilizes DL algorithms to train emotion recognition models, which can be further enhanced if needed. Finally, the trained models are tested to evaluate their accuracy in detecting emotions in speech signals.

3.1.2.1 Data storage

The data storage in a SER model is crucial for seamless progress in subsequent steps. It ensures easy access and retrieval of the dataset, facilitating preprocessing, feature extraction, model training, and evaluation, leading to efficient development and deployment of the SER system. There exist two datasets to work on in the SER system which are RAVDESS [15] and TESS [16].

- The RAVDESS [15] dataset is a comprehensive speech emotional database featuring audio recordings from 24 professional actors. With 1400 audio files in WAV format, it covers eight different emotions and provides valuable resources for emotion recognition studies.

- The TESS [16] dataset comprises 200 target words spoken by two actresses, capturing all seven emotions. The actresses have normal hearing thresholds and English as their first language, with university education and musical training. The dataset includes a total of 2800 stimuli.

3.1.2.2 Preprocessing

Preprocessing is the most important and initial process that enhances the quality of the DL model. It helps a lot in improving the databases that will be worked on. Preprocessing techniques such as merging the TESS and RAVDESS datasets, splitting the dataset to train and test parts, feature scaling, Exploratory Data Analysis (EDA), data cleaning, dimensionality reduction, noise removal, and data balancing play a crucial role in enhancing the quality of the DL model.

- Merging the TESS and RAVDESS datasets

This merging method involves combining the TESS and RAVDESS datasets, which contain different sets of data, to create a unified dataset. It allows for a broader and more diverse range of data to be used in the DL model, potentially improving its performance and generalization ability.

- Splitting the used dataset

This splitting technique involves dividing the audio dataset into two separate subsets: a training set and a test set. The training set is used to train the DL model, while the test set is used to evaluate its performance. This ensures that the model's performance is assessed on unseen data, providing an estimate of its generalization ability.

- Feature scaling

Feature scaling is the process of transforming audio features to a similar scale. It helps in preventing certain features from dominating the model due to their larger magnitudes, thus ensuring fair and balanced influence of all features during model training.

- EDA

EDA involves analyzing and visualizing the dataset to gain insights, identify patterns, detect outliers, and understand the relationships between variables. It helps in understanding the characteristics of the data and making informed decisions regarding data preprocessing and model design.

Figure 3.2 below shows the distribution of the eight Emotions after merging the two RAVDESS and TESS datasets.

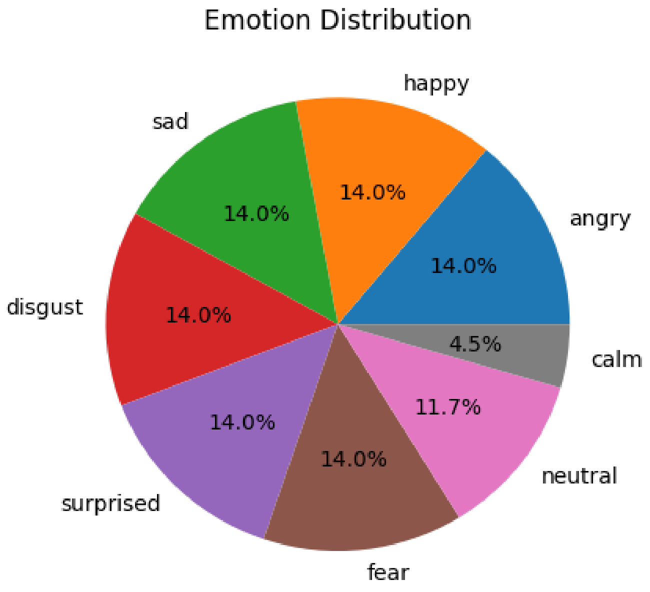


Figure 3. 2 Emotions Distribution

- Signal Noise Removal

Noise removal techniques involve filtering or denoising algorithms to remove unwanted disturbances or artifacts from the audio data. It helps in improving the quality and reliability of the audio dataset, leading to better model performance.

- Audio Data Balancing

Balancing data using class weights in DL is vital for addressing imbalanced datasets. By assigning higher weights to underrepresented classes and lower weights to overrepresented classes during training, this technique helps overcome biases and ensures fair representation. It improves model performance and accuracy across all classes, making it an essential tool in SER tasks.

3.1.2.3 Features Extraction

Modern DL on audio class recognition includes feature extraction as a key component. Feature extraction process in DL involves reducing an initial set of data by identifying key features. In the case of audio data, Low Level Descriptions (LLDs) like MFCC, Chroma, Mel-spectrogram, Contrast, ZCR, Root Mean Square Energy (RMSE), energy and Tonnetz are used to extract audio features. In the second level of processing, High Level Descriptions (HLDs) such as Mean and Standard deviation are applied to obtain a comprehensive representation of the signal, aiming for an overall perfect representation for the signal.

- MFCC

Due to the non-smooth random nature and time variability of speech signals, it is practical to extract feature parameters using MFCC as one of the most important features. Figure 3.3 below depicts the MFCC, a speech emotion feature parameter utilized in automatic speech and speaker recognition.

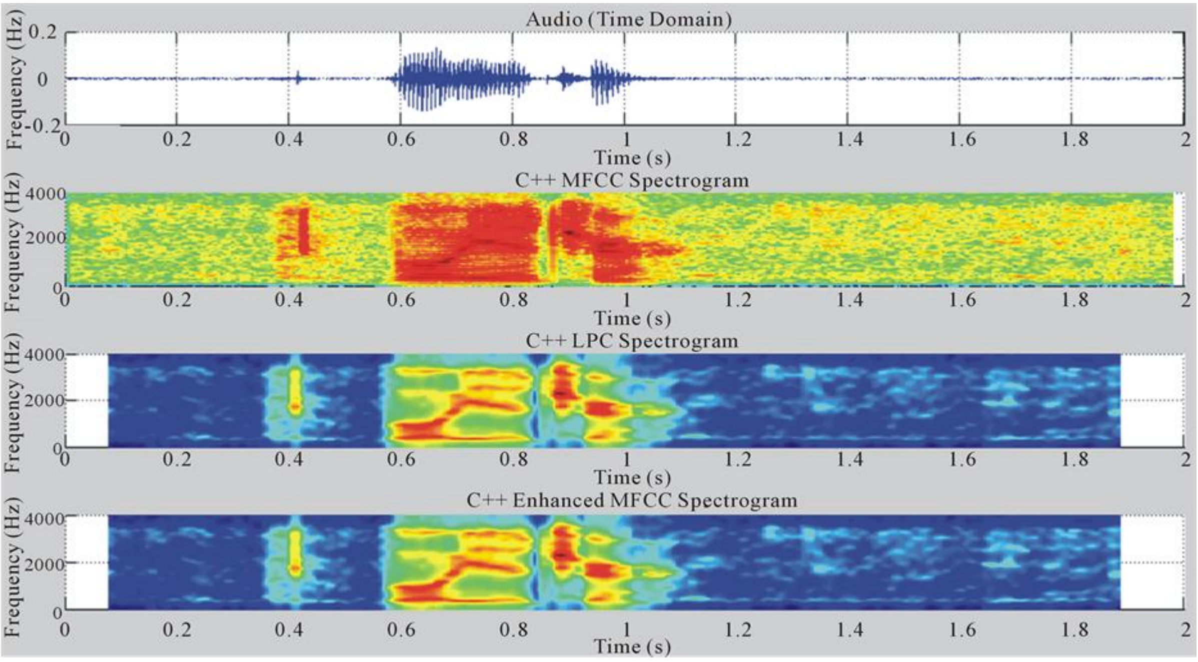


Figure 3. 3 MFCC

The vocal tract can be visualized through the envelope of the time power spectrum of the voice signal, and this envelope is accurately represented by MFCC. MFCC is composed of the Mel frequency cepstral, which captures the short-term power spectrum of any sound. It is derived using the inverse Fourier transform cepstral representation. The Mel frequency cepstral provides a more accurate depiction of sound because its frequency bands, distributed evenly on the Mel scale, closely mimic the response of the human auditory system [17].

The total amount of extracted parameters was:

- 40 MFCC
- 128 Mel spectrogram.
- 12 chromograms.
- Other 6 features (RMS energy, energy, ZCR, spectral centroid, spectral flux, and spectral roll off).

- Mel Spectrogram

Mel spectrogram is a representation of frequencies in the Mel scale. The Mel scale comprises pitches that are equally spaced for the listener. The Mel scale is based on how the human ear works, which better detects differences at lower frequencies than higher frequencies. The Fourier transform can be used to convert frequencies to the Mel scale.

The major three steps for creating Mel Spectrogram are:

- Compute the fast Fourier transform.
- Generate Mel Scale.
- Generate spectrogram.

- RMSE

The RMSE is calculated based on the total number of samples within a frame. It provides an indication of loudness, as higher energy corresponds to louder sounds. Additionally, the RMSE is less affected by outliers. By taking the square root of the mean squared amplitude over a specific time interval, the RMSE is characterized as shown in Equation (3.1):

$$\mathrm{RMS}_t = \sqrt{\frac{1}{K} \sum_{k=t.K}^{(t+1)\cdot(K-1)} s\,(k)^2}. \tag{3.1}$$

- ZCR

The ZCR, defined as the number of zero crossings in a specific region of the signal divided by the number of samples in that region, represents the rate at which the signal crosses the zeroth line. In other words, it quantifies how frequently the signal transitions from positive to negative or vice versa. Mathematically, it can be expressed as shown in Equation (3.2) and (3.3).

$$\mathrm{ZCR} = \frac{1}{N-1} \sum_{n=1}^{N-1} \operatorname{sign}\left(s\,(n)\, s\,(n-1)\right), \tag{3.2}$$

where s = signal, N = length of a signal, and the sign(s(n) s(n-1)) is calculated as

$$\text{sign}\,(s\,(n)\,s\,(n-1)) = \begin{cases} 1, \text{if}\, s\,(n)\,s\,(n-1) \geq 0 \\ 0, \text{if}\, s\,(n)\,s\,(n-1) < 0 \end{cases} \tag{3.3}$$

- Energy

Energy is the overall magnitude of a signal, i.e., how loud it is, is the signal's energy. It is defined as in Equation (3.4)

$$E\,(x) = \sum_{n} |x\,(n)|^2 \,. \tag{3.4}$$

- Mean

The mean statistical function is important in SER as it provides a representative summary of emotional features in speech signals. It allows for efficient HLDs and is less sensitive to outliers, leading to more accurate and robust emotion recognition systems. Mathematically, the mean can be represented by the Equation (3.5).

$$\overline{X} = \frac{\sum X}{N} \tag{3.5}$$

- Standard Deviation

The standard deviation is crucial in SER as it captures the variability of emotional features in speech signals, enabling a more nuanced understanding of emotions. It helps distinguish between speech segments with similar means but different emotional expressions. Mathematically, the standard deviation can be represented by the Equation (3.6).

$$\sigma = \sqrt{\frac{\sum (X - \mu)^2}{n}} \tag{3.6}$$

3.1.2.4 Data Augmentation

In the field of SER, data augmentation is of utmost importance. SER aims to detect and interpret emotions conveyed through speech signals. However, obtaining a large and diverse dataset with a wide range of emotional expressions can be challenging. Data augmentation techniques offer a solution by artificially generating additional training samples. By applying techniques such as pitch shifting, time stretching, noise addition, and vocal tract modification, the dataset can be expanded, allowing the model to learn from a more comprehensive set of emotional variations. This augmentation process enables the model to better generalize and recognize emotions accurately in real-world scenarios, enhancing the overall performance of SER systems.

Some of the augmentation models are:

- Speed perturbation**.**
- Spec augments (time warping, frequency masking, time masking).
- Spec swap (frequency swap, time swap).
- Generative Adversarial Networks.

3.1.2.5 SER Model Training

The SER training process involved the use of six distinct models, namely CNN, MLP, LSTM, CU-DNN-LSTM, BLSTM, and Gated Recurrent Unit-LSTM (GRU-LSTM).

The utilization of LSTM models and their variations, such as BLSTM and stacked LSTMs, has proven to be of great importance in SER systems. These models, as depicted in their respective architectures, can capture long-term dependencies and temporal dynamics in sequential data like speech signals. The inherent structure of LSTM enables the retention of contextual information over extended sequences, facilitating the accurate identification and classification of emotional patterns that unfold over time in speech. By incorporating bidirectional information flow and deeper hierarchical representations, variations of LSTM models further enhance the modeling capability, leading to improved accuracy and robustness in recognizing and categorizing emotions expressed through speech.

As illustrated in Figure 3.4, the LSTM general architecture consists of input nodes, weighted connection, and memory blocks.

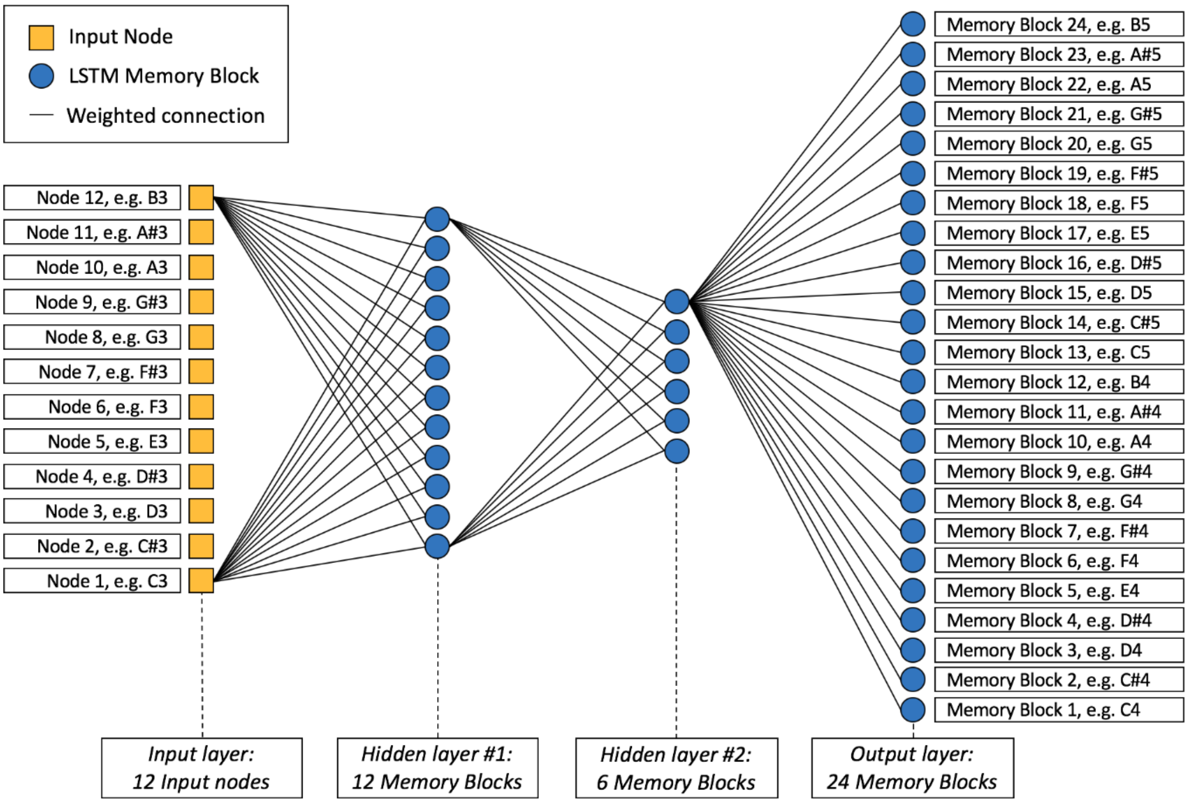


Figure 3. 4 Blocks of the LSTM Architecture

3.1.2.6 SER model enhancement

Enhancing SER model performance refers to a collection of techniques aimed at mitigating overfitting and optimizing the performance of a learning algorithm. Regularization plays a crucial role in this process by reducing the generalization error while maintaining the training error. It encompasses various methods that prevent overfitting and aid optimization. Some commonly used regularization techniques include dropout, drop connection, data augmentation, momentum, and weight decay. Dropout randomly drops out units during training to reduce reliance on specific features as shown below in Figure 3.5. while drop connection performs a similar function on connections between layers as shown below in Figure 3.6.

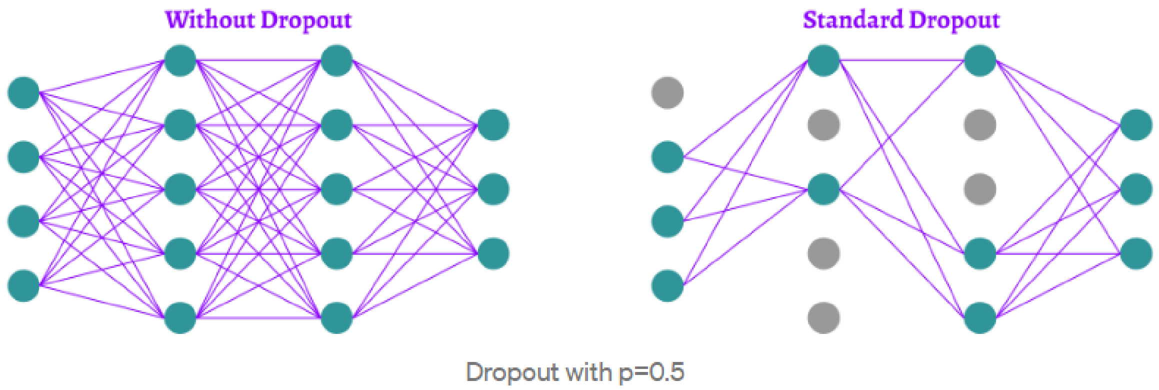


Figure 3. 5 Drop Out

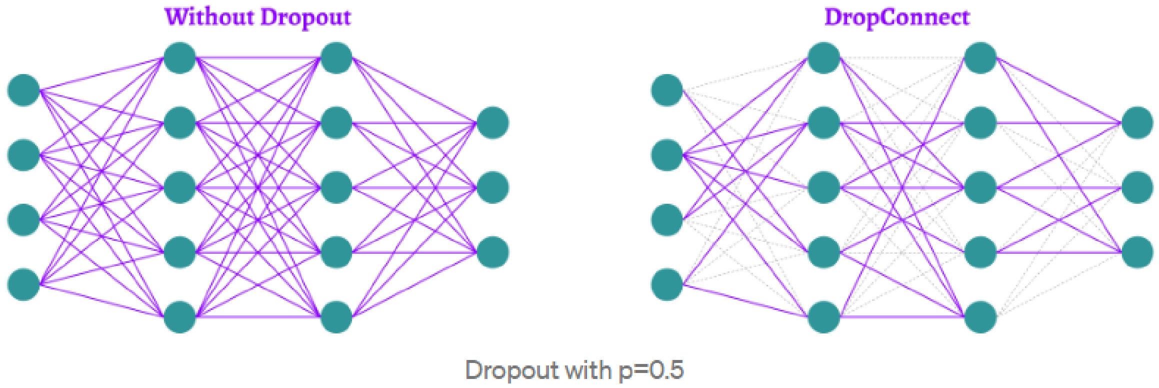


Figure 3. 6 Drop Connection

Data augmentation involves generating additional training samples with variations to enhance the dataset. Momentum helps accelerate convergence by accumulating gradients from previous steps, and weight decay imposes a penalty on large weight values to encourage simpler models. By employing these regularization techniques, SER model enhancement can effectively improve the generalization capabilities and optimization performance of the learning algorithm.

3.1.2.7 Model Testing

SER model testing is a crucial step in evaluating the accuracy and overall performance of a trained model in SER. This process involves assessing various metrics, including the train accuracy, test accuracy, and validation accuracy of the model. Train accuracy measures how well the model performs on the training dataset, indicating how effectively it has learned from the labeled data. Test accuracy evaluates the model's performance on unseen data, providing insights into its ability to generalize and make accurate predictions on new samples. Validation accuracy serves as an intermediate evaluation metric during the training process, guiding the adjustment of model parameters and hyperparameters to optimize its performance. By analyzing these metrics, it will be evident to gain valuable insights into the effectiveness of the model and determine its suitability for real-world applications.

3.1.2.8 Emotion Detection

Emotion detection serves as the final stage where the predicted emotion is identified using a classifier, providing the desired output for the user. After the speech signals undergo feature extraction and classification, the trained classifier analyzes the extracted features and assigns the corresponding emotion label to the input speech. This process allows the system to effectively recognize and categorize the eight emotions conveyed in the speech signals, providing valuable information to the user regarding the detected emotion.

3.2 SER System Users

SER system users can be individuals or organizations who interact with the system to obtain valuable insights and information regarding emotions expressed in speech. SER system users may include researchers, psychologists, speech therapists, HCI designers, sentiment analysis practitioners, and professionals in fields where understanding and analyzing emotions in speech are essential. By leveraging the outputs and functionalities of the speech emotion recognition system, users can gain valuable insights into the emotional states conveyed through speech, contributing to various applications such as affective computing, personalized therapy, sentiment analysis, and HCI.

3.2.1 Intended Users

The intended users of the SER system include companies, doctors, and movie website owners. These entities can benefit from utilizing the system in various ways to enhance their operations and services.

- **Companies** can know a person's current feelings and suggest the right product to him using the SER system**.**
- **Doctors** in psychotherapy sessions can record the patient’s voice, knowing the speech feelings of patients.
- **Movie websites** can use it to recommend movies to users based on their feelings at the time.

3.2.2 User Characteristics

No technical experience is needed for the user to use the SER system.

3.3 SER System Diagrams

In SER, various diagrams play a crucial role in illustrating the system's architecture, interactions, and flow. These diagrams provide a visual representation of the components, relationships, and processes involved in SER. Use case diagram describes a function that can be performed by the system and the user. Class diagrams showcase the structure of the system by depicting the classes, attributes, and methods involved. Sequence diagrams demonstrate the chronological order of interactions between different components during the emotion recognition process. Flowchart diagrams depict the flow of data and control within the system, highlighting the steps and decisions involved.

3.3.1 Use Case Diagram

The use case diagram for SER model describes a function and requirement that a system performs to achieve the user's goal to detect the emotion from the speech as shown below.

In the interaction between the SER model and the user, the user provides speech input through a microphone or by uploading an audio file. This input undergoes preprocessing and feature extraction, followed by analysis and classification by the SER model to determine the predicted emotion(s).

The SER model then generates an output that represents the predicted emotion(s), which is presented to the user for further interpretation and utilization. This interaction enables users to gain insights into the emotional content conveyed through speech as shown below in figure 3.7 below.

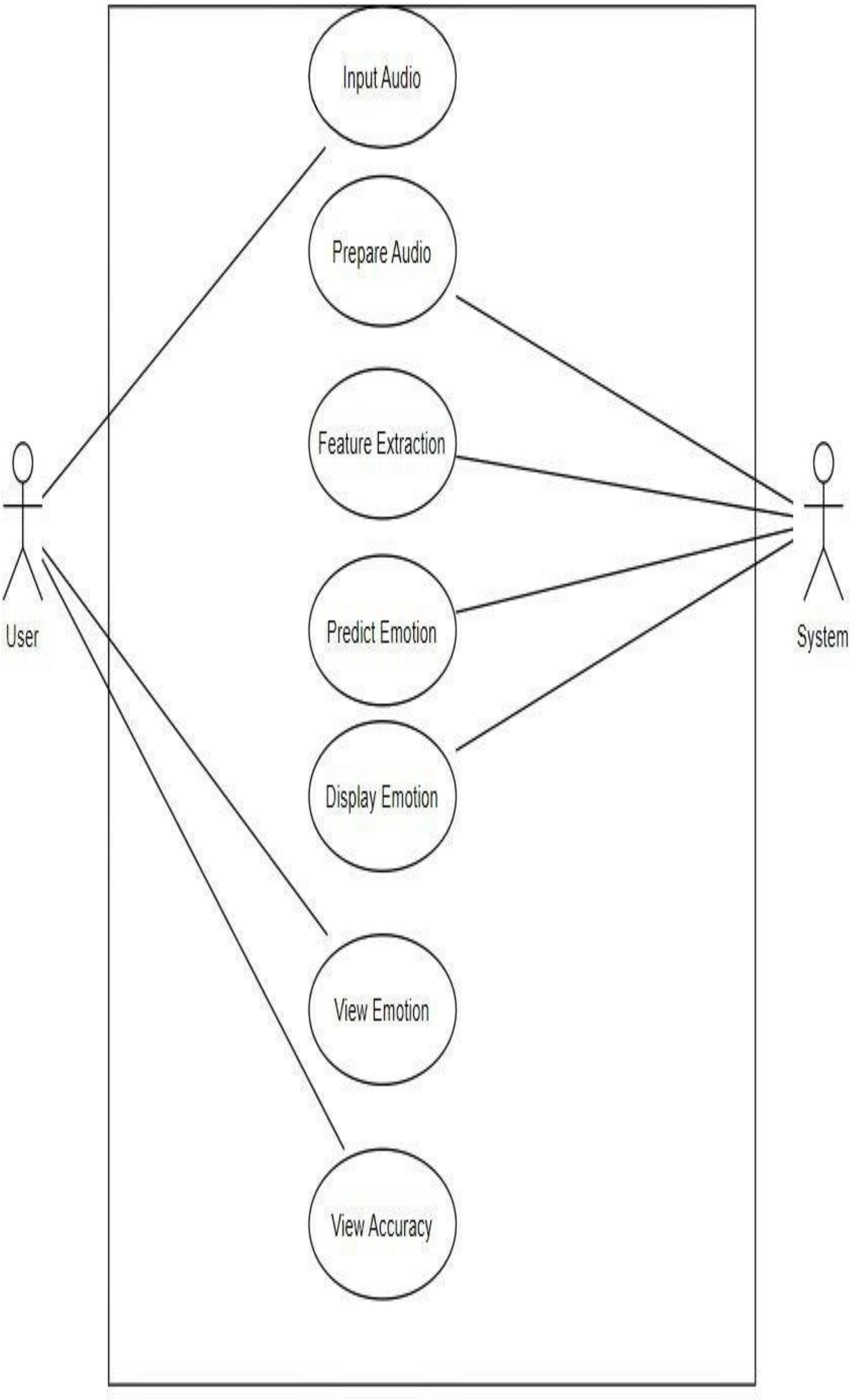


Figure 3. 7 SER Use Case diagram

3.3.1.1 User functions

User functions in SER system encompass various actions and capabilities available to users. These functions involve providing speech input, accessing the system interface, selecting features or settings, and interpreting the predicted emotion outputs.

- The user shall record his voice by pressing the button and then he will record his voice.
- The user shall view his emotion after recording his voice.
- The user shall get the accuracy for the classification process.

3.3.1.2 System functions

A SER system acquires, and preprocesses speech signals, extracts relevant features, selects informative features, classifies emotions using DL model, trains and evaluates the model, and applies it in real-time for accurate emotion recognition in speech.

- The system will save the audio in a .wav file format.
- The system will load the voice from the database in .wav format.
- The system will prepare the audio.
- The system will extract audio features.
- The system will predict emotion.
- The system will display the emotions in real time.
-

3.3.2 Class Diagram

Assuming figure 3.8 below, the SER system is comprised of three classes: system, homepage, and classification. Together, these classes work in synergy to enable the system to perform SER effectively.

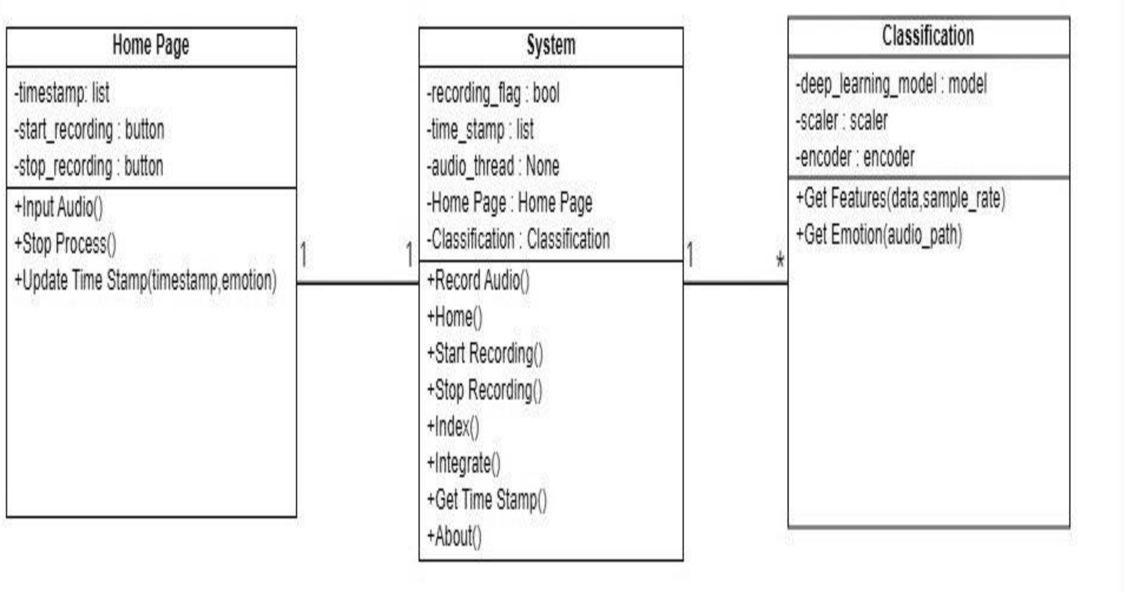


Figure 3. 8 SER Class diagram

3.3.2.1 Home page class

The Homepage class is responsible for managing the user interface and interaction with the system, providing a centralized hub for user input, and displaying relevant information.

- A graphical user interface that the user interacts with to record his voice and view the emotion results.
- Contains timestamp list, start recording button, stop recording button.
- Users can input audio or stop the process.
- The time stamp is updated in real time.

3.3.2.2 SER System class

The System class represents the core functionality and components of the SER system.

- The core class in the system which is used to control the process.
- Contains timestamp list, recording-flag, audio-thread, home page object, classification object.
- Start recording process or stop it.
- Get updated time stamp.
- Controlling the other pages such as: About us, home, integration, and prediction.

3.3.2.3 Classification class

The classification class encompasses the algorithms and techniques used to analyze speech features and determine the emotional state of the speaker.

- Contains DL model, scaler, and encoder.
- Extracts the features from the input audio.
- Predict the emotion state.

3.3.3 Sequence Diagram

The interaction diagram (Sequence diagram), which shows user pipeline for predicting the emotion of the voice as shown in figure 3.9.

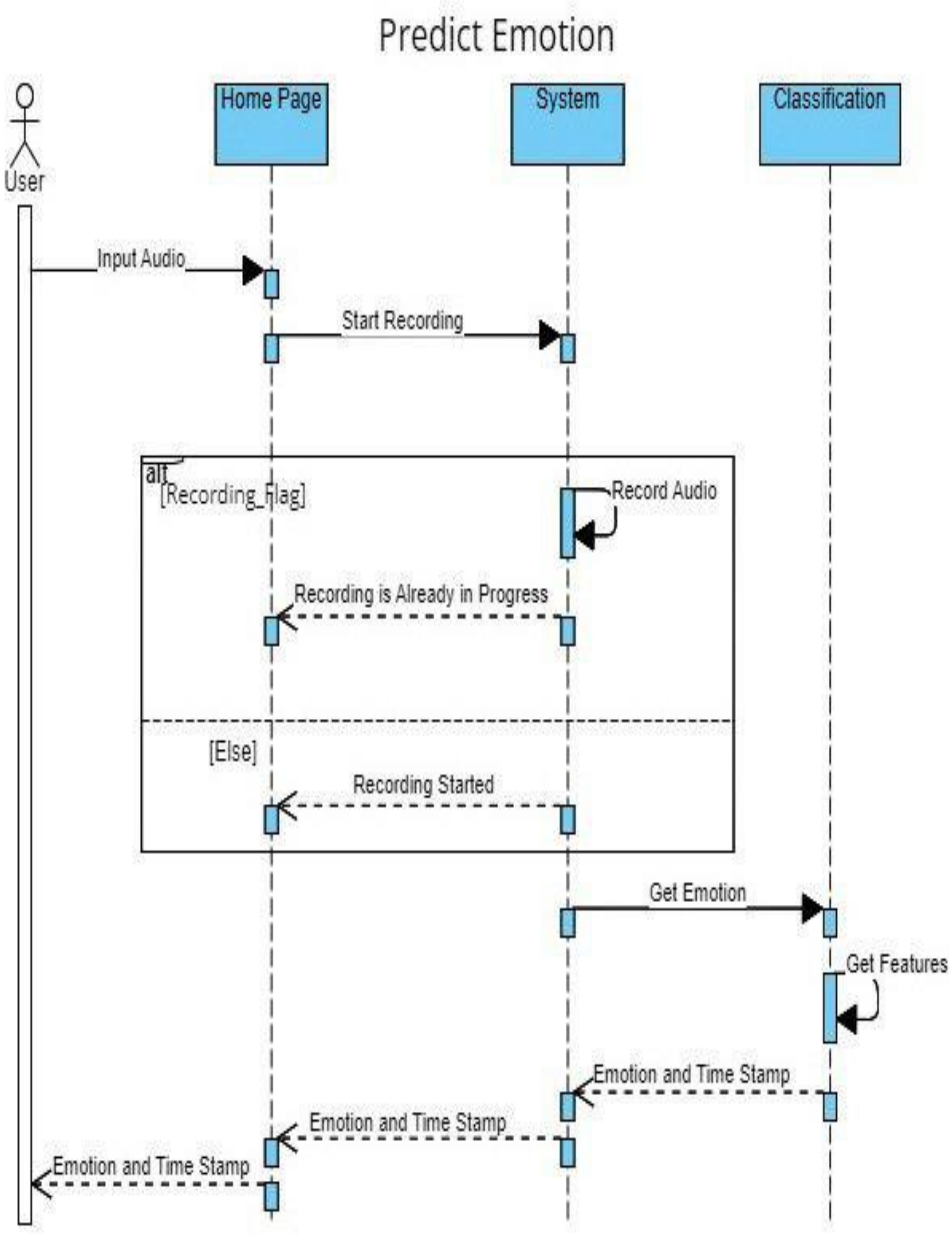


Figure 3. 9 SER Sequence diagram

3.3.4 Flow chart diagram

As shown below in Figure 3.10 the flow chart diagram for a SER system provides a visual representation of the sequential steps involved in the system's operation. It illustrates the flow of data and processes from input to output. The diagram typically includes various components, such as speech signal acquisition, preprocessing, silence removal, feature extraction, emotion classification, and output generation.

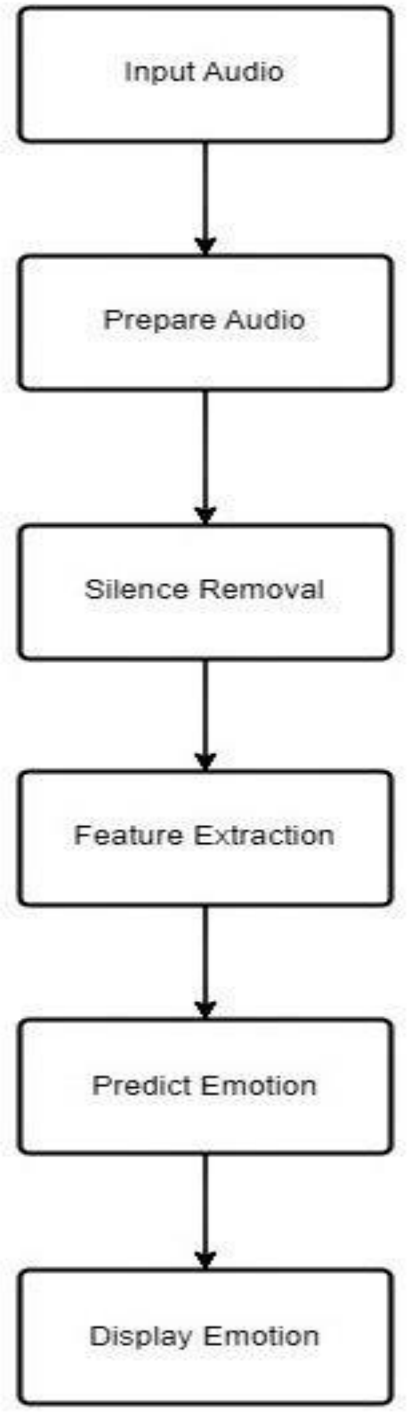


Figure 3. 10 SER Flow chart diagram

3.3.5 Used Dataset

The dataset used in the SER system plays a crucial role in training and evaluating the performance of the system. A high-quality and diverse dataset is essential for building robust SER models that can accurately recognize and classify emotions from speech signals.

3.3.5.1 RAVDESS Dataset

RAVDESS [15] contains 1400 files. The database contains 24 professional actors, vocalizing two lexically matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression. All conditions are available in three modality formats: Audio-only (16bit, 48 kHz .wav), audio-video (720p H.264, AAC 48 kHz, .mp4), and video-only (no sound). Each of the 1440 files has a unique filename. The filename consists of a 7-part numerical identifier (e.g., 03-01-06-01-02-01-12.wav). These identifiers define the stimulus characteristics:

Filename identifiers:

- Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
- Vocal channel (01 = speech, 02 = song).
- Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
- Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.
- Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
- Repetition (01 = 1st repetition, 02 = 2nd repetition).
- Actor (01 to 24. Odd numbered actors are male, even numbered actors are female).

3.3.5.2 TESS Dataset

TESS [16] contains 2800 files. Collection these stimuli were modeled on the Northwestern University Auditory test No. 6 (NU-6; Tillman & Carhart, 1966) [16]. 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 stimuli in total. Two actresses were recruited from the Toronto area. Both actresses speak English as their first language, are university educated, and have musical training. Audiometric testing indicated that both actresses have thresholds within the normal range. TESS contains audio recordings of actors portraying seven cardinal emotions, here are the main attributes found in TESS:

- Actor ID: Each audio recording is associated with a unique identifier.
- Emotion Label: TESS focuses on seven cardinal emotions, including anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral. Each audio file is labeled with the corresponding emotion category.
- Audio Format: TESS audio files are commonly provided in a standard format like Waveform Audio File Format, which is a widely used audio file format that maintains high-quality audio.
- Sampling Rate: The sampling rate refers to the number of audio samples captured per second.

Chapter 4

Implementation and Testing

Chapter 4

Implementation and Testing

The implementation and testing phase of SER system development is crucial for translating theoretical concepts into practical code and evaluating the system's performance. This chapter focuses on the implementation process, highlighting key components and techniques used to create an effective SER system. It also covers testing methodologies employed to assess the system's accuracy and reliability in recognizing emotions from speech signals.

During implementation, the research and theoretical concepts are transformed into executable code. This involves tasks like preprocessing speech signals, extracting features, building models, and integrating modules. The section provides an overview of the software tools, programming languages, and libraries used in the implementation process.

4.1 Software tools & Environments

The development of the SER system requires the utilization of specific software tools and environments to facilitate the implementation and testing processes. This section provides a brief overview of the commonly employed software tools and environments in SER development.

4.1.1 Programming languages

Programming languages are essential for implementing the SER system. They provide the foundation for writing the code that processes speech signals, extracts feature, builds models, and performs classification tasks.

- Python 3.10
- Dart
- JavaScript

4.1.2 Integrated Development Environments (IDE) & Other Environment

In addition to programming languages, the development of a SER system often involves utilizing IDEs and other related environments. These tools provide a comprehensive development environment and aid in the implementation and testing processes.

Here are some notable environments used in SER development.

- IDE: PyCharm

  For building the SER application, it was used as it includes python language with needed libraries, which allow dealing with the DL models.

- Cloud Service

  Google COLAB to run our scripts it was used as it provides free access to Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs), which allows training the DL models with high speed.

4.1.3 Libraries & Packages

In the development of a SER system, leveraging libraries and packages can greatly simplify the implementation process by providing pre-built functionalities and algorithms. These libraries offer a wide range of tools and functions for tasks such as audio processing, feature extraction, DL, and evaluation. Here are some commonly used libraries and packages in SER development.

- **Librosa**: is a python package for music and audio analysis. It provides the building blocks necessary for audio analysis.

- **TensorFlow**: is an end-to-end open-source platform for DL. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in DL and developers easily build and deploy DL-powered applications.

- **Keras**: offers consistent & simple Application Programming Interfaces (APIs), it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

- **Operating System (OS)**: module in Python’s standard library provides functions for interacting with the operating system. This module provides a portable way of using OS-dependent functionality.

- **Pandas**: is a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.

- **Numerical Python (NumPy)**: brings the computational power of languages like C and Fortran to Python, a language much easier to learn and use.
- **Scikit-Learn (Sklearn)**: is also known as Sklearn. It’s a Free and the most useful ML Library for Python. Sklearn library comes loaded with a lot of features such as classification, regression, clustering, and dimensionality reduction algorithms include k-Means, k-Nearest neighbors, SVM, decision trees also support python numerical and scientific libraries NumPy and scientific libraries NumPy and Scientific Python SciPy. It is also used to build ML models.

- **Matplotlib**: is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

4.2 Implementation of SER System Modules

This section focuses on the practical implementation of the key modules within the SER system. It delves into the technical details and steps involved in building and integrating these modules to create a functional SER system.

4.2.1 Extract, Transform, and Load (ETL)

In the implementation phase of a SER system, the dataset needs to undergo an ETL process. ETL refers to the steps involved in extracting the data, transforming it into a suitable format, and loading it into the SER system for further processing. This section provides a brief overview of the ETL process for preparing the dataset in SER implementation.

- Getting Data Function

The function "get_data()" performs the extraction of emotions, filenames, audio, and source from the dataset. It takes no arguments and returns nothing. The purpose of this function is to extract the required information from the dataset for further processing in the SER system. The function processes the dataset to extract the emotions associated with each speech sample, the corresponding filenames, the audio data itself, and the source of the data.

1. Description: Extract Emotions, Filenames, Audio and Source from Dataset.
2. Arguments: None.
3. Return: None.

- Splitting Data into Training and Validation Function

train_test_split(df_all,test_size=0.2,shuffle=True,stratify=df_all["emotion"])

The function "train_test_split ()" is used to split the dataset into a training set and a validation set. Here is a brief description of this function:

1. Description: Split the dataset to 80% train and 20% test.
2. Arguments: data, test size, shuffle, stratify.
3. Return: train, test.

- Splitting Data into Validation and Testing Function

train_test_split(X_TEST,Y_TEST,test_size=0.5,shuffle=True,stratify=Y_TES)

The function "train_test_split()" is used to split the validation dataset into a testing set and a validation set. Here is a brief description of this function:

1. Description: Split the test data to 50% test and 50% valid.
2. Arguments: data, test_size, shuffle, stratify.
3. Return: x_test, x_valid, y_test, y_valid.

4.2.2 Preprocessing

Preprocessing is an essential step in the implementation of a SER system. It involves several techniques to prepare the data for further analysis and modeling. This section focuses on various preprocessing steps, including scaling, encoding, silence removal, EDA, and generating an audio signals Data Frame table.

4.2.2.1 EDA

- Visualize (data, feature) Function.

The "Visualize (data, feature)" function allows for the visualization of audio features within the SER implementation.

1. Description: Visualize all audio features.
2. Arguments: data, feature.
3. Return: None.

- Audio wave form

Audio Waveform: As shown in Figure 4.1 below, the audio waveform provides a visual representation of the amplitude of the audio signal over time. It offers insights into the shape and intensity of the speech or audio signal.

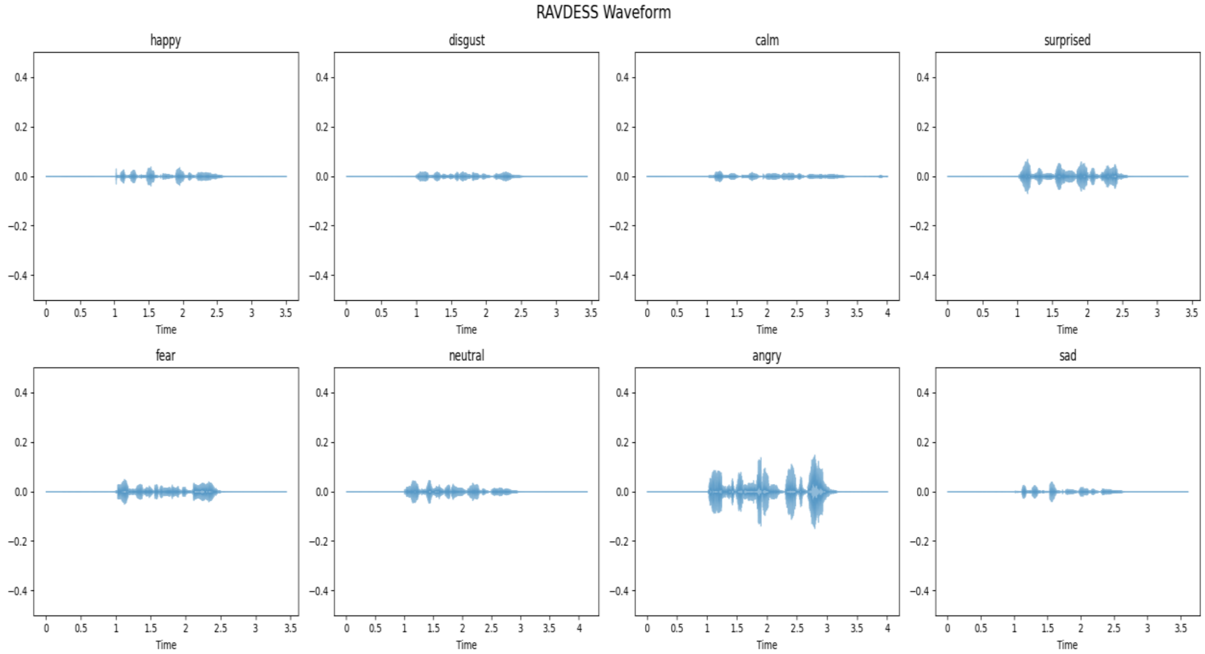


Figure 4. 1 Audio Wave Form

- MFCC

As shown in Figure 4.2 below, the MFCC plot displays the computed MFCC coefficients extracted from the audio signal. MFCC captures the spectral characteristics of the speech signal by extracting relevant frequency bands for human auditory perception.

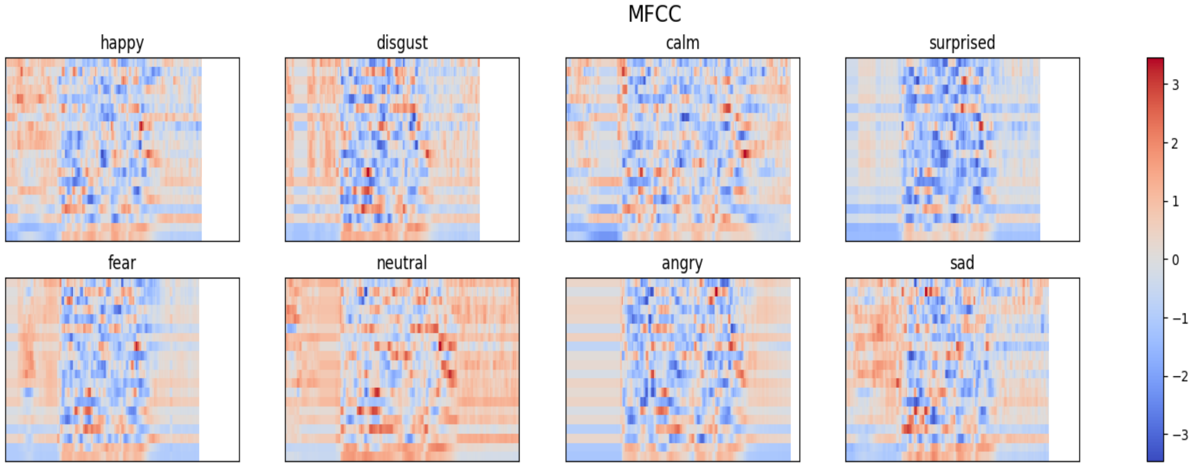


Figure 4. 2 MFCC

- Mel spectrogram

The Mel spectrogram Figure 4.3 below represents the intensity of different frequencies of the audio signal over time. It provides a visual representation of the spectral content of the speech signal, highlighting variations in frequency components and their intensity.

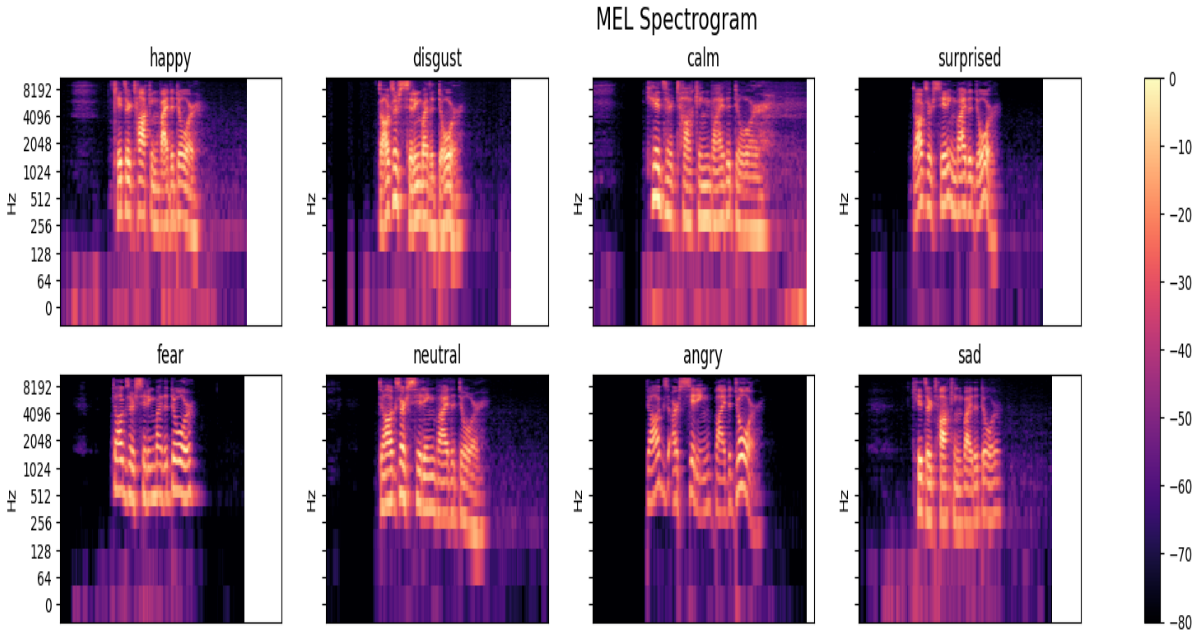


Figure 4. 3 Mel spectrogram

- Chroma- Short-Time Fourier Transform (Chroma-STFT)

As illustrated in Figure 4.4 below, the Chroma-STFT plot depicts the distribution of musical pitch classes over time. It provides insights into the tonal content of the audio signal, indicating the presence of different musical notes.

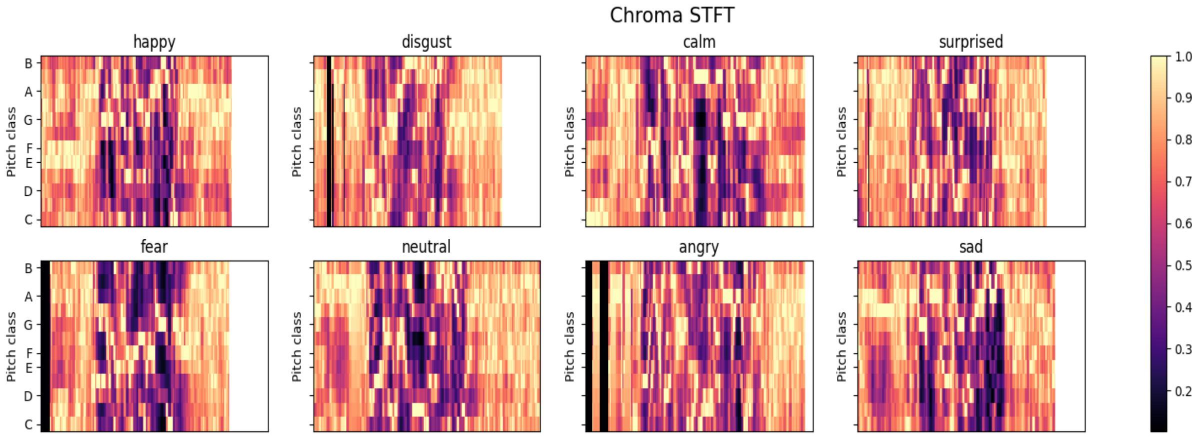


Figure 4. 4 Chroma STFT

- RMSE

As shown in Figure 4.5 below, the RMSE plot displays the RMSE of the audio signal over time. It represents the overall energy or loudness of the signal at different time intervals.

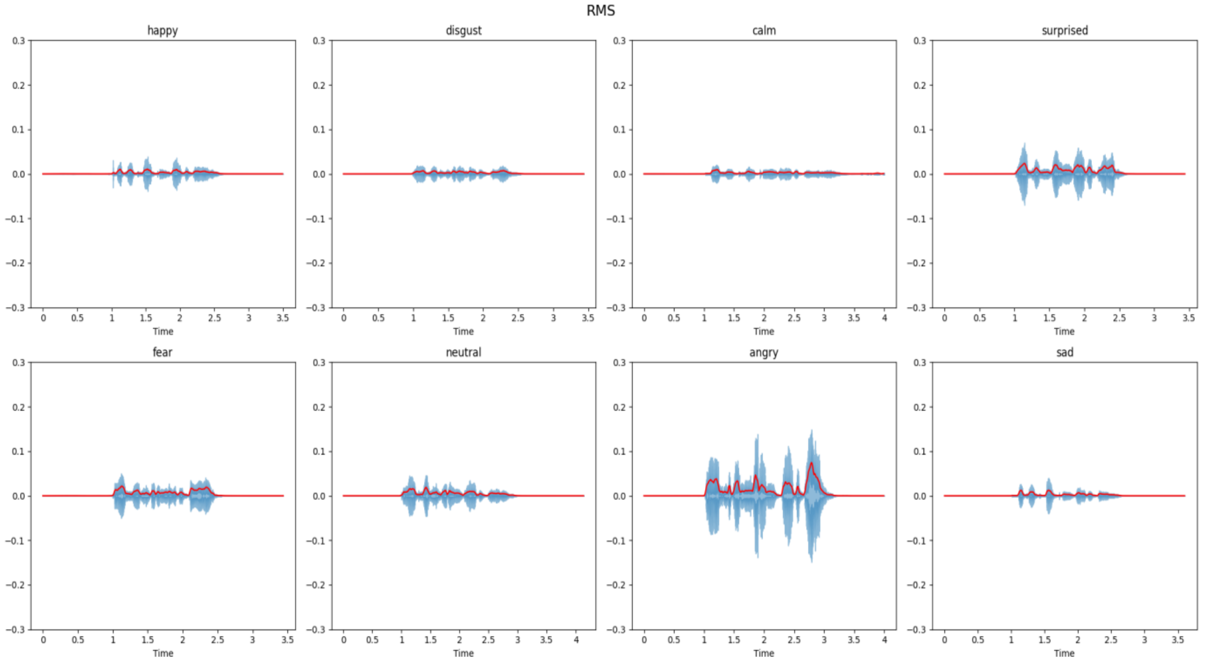


Figure 4. 5 RMSE

- ZCR

Figure 4.6 below represents the rate at which the audio signal crosses the zero axis over time. It provides insights into the variations in the signal's waveform and can indicate changes in speech characteristics or transitions between voiced and unvoiced segments.

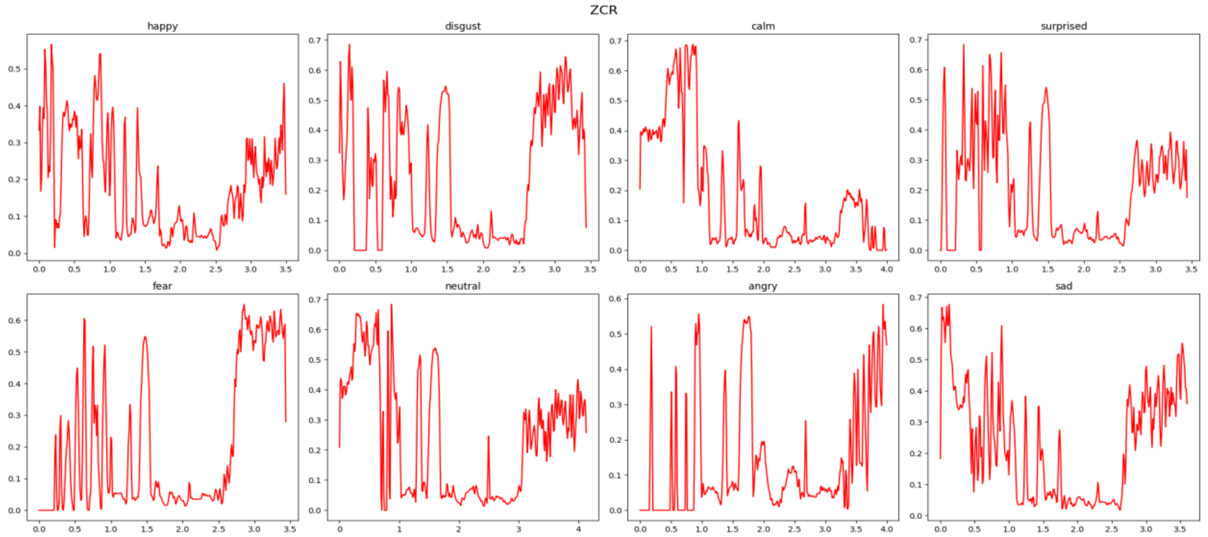


Figure 4. 6 ZCR

4.2.2.2 Silence removal

- Removing silence function

The function "librosa.effects.trim(y=audio_wave) "is important for removing leading and trailing silence from an audio signal, allowing for a more focused analysis of the informative speech segments based on a specified threshold. Here is a brief description of this function:

1. Description: Prepare data by removing silence from an audio signal.
2. Arguments: audio_wave.
3. Return: y_trimmed.

4.2.2.3 Data Generating

- Generating Training Data

The function "generate_train_Data(data1)" is responsible for preparing the training data in the SER system. Here is a brief description of this function:

1. Description: Prepare train data by removing silence, extracting features, and augmenting data.
2. Arguments: data1.
3. Return: RESULT.

- Generating Testing Data Function

The function "generate_test_Data(data1)" is responsible for preparing the training data in the SER system. Here is a brief description of this function:

1. Description: Prepare test data by removing silence, extracting features, and augmenting data.
2. Arguments: data1.
3. Return: RESULT.

4.2.2.4 Scaling

Scaling is an essential step in the implementation of a SER system. It involves normalizing and transforming the feature values to a consistent range, such as 0 to 1.

- Scaling Function

The function "StandardScaler()" is used for scaling features within a specified range, typically from 0 to 1, using a standard scaler.

1. Description: Scaling features in range from 0 to 1 using standard scaler.
2. Arguments: None.
3. Return: scaler.

4.2.2.5 Encoding

Label encoding in a SER system is important for numerical representation, algorithm compatibility, class balancing, evaluation, interpretation, and future compatibility.

- Train Data Encoding Function

In the context of the implementation of a SER system, the function "train_class_one_hot_encoding(y)" is used for encoding the labels in the training data.

1. Description: Encoding labels in train data.
2. Arguments: y.
3. Return: new_y.

- Test Data Encoding Function

the function "test_class_one_hot_encoding(y)" is used for encoding the labels in the testing data by transformation process.

1. Description: Encoding labels in test data.
2. Arguments: y.
3. Return: new_y.

4.2.3 Feature extraction

The feature extraction step plays a crucial role in the SER system by extracting features from the audio dataset. This function encompasses both low-level descriptions, such as MFCC, MEL-Spectrogram, Tonnetz, spectral contrast, chroma STFT, and harmonic features, as well as high-level descriptions that incorporate statistical functions like mean and standard deviation. By capturing both the intricate acoustic details and summarizing them using statistical measures, the extracted features provide a comprehensive representation of the emotional content in the audio signals. These features serve as essential inputs for training and evaluating the SER model, enabling it to effectively analyze and recognize emotions in speech data.

- Feature Extraction Function

The function "Feature_Extraction(X, sample_rate=22050)" extracts various features such as MFCC from the dataset, incorporating a second level of feature extraction that includes mean and standard deviation.

1. Description: extract features from dataset such as: MFCC, MEL-Spectrogram, Tonnetz, spectral contrast, chroma STFT, and harmonic features. Apply the second level of feature extraction which includes (mean and standard deviation).
2. Arguments: X,sample_rate.
3. Return: result.

4.2.4 Data augmentation

Data augmentation in SER is essential for expanding training data diversity. By applying transformations like pitch shifting, time stretching, noise addition, shift, higher speed, and lower speed, it improves model generalization, combats overfitting, and enhances robustness to signal variations. This leads to improved performance in emotion recognition tasks by exposing the model to a wider range of examples.

Original Audio wave

This is the original form of the audio before any augmentation. Figure 4.7 represents the signal of original audio.

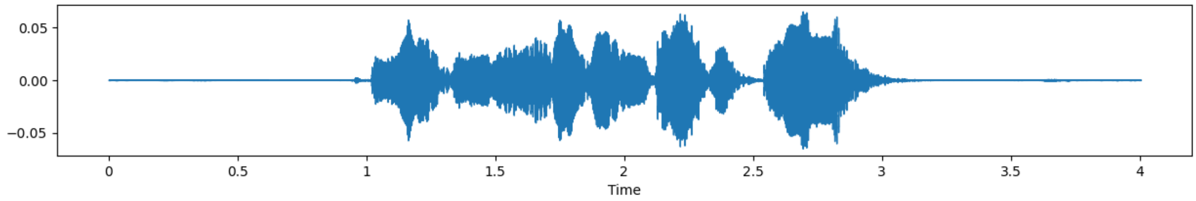


Figure 4. 7 Original Audio

4.2.4.1 Noise

- Noise Function

The "noise(data)" function creates a new form of the audio wave by adding random noise. It combines the original "data" with noise_amp multiplied by a random normal distribution of the same shape as the input data.

1. Description: Add some noise data to the audio wave.
2. Arguments: data.
3. Return:data+noise_amp*np.random.normal(size=data.shape[0]).

As shown in Figure 4.8 below, this is the result of adding noise to the original sound.

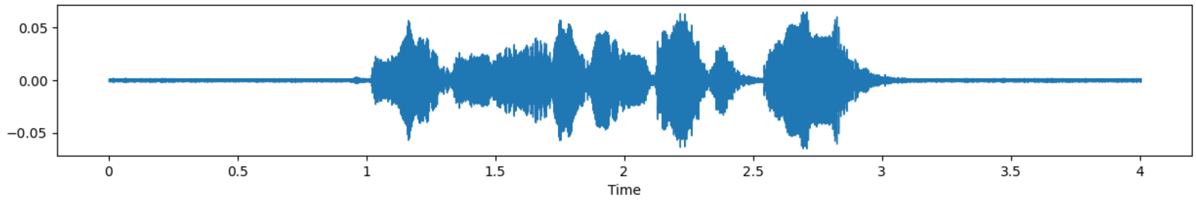


Figure 4. 8 Noise

4.2.4.2 Stretch

- Stretch Function

The "stretch (data, rate=0.8)" function creates a stretched version of the audio waveform by a specified rate. It modifies the original "data" by stretching it, resulting in a waveform with altered timing and duration.

1. Description: Time-stretch an audio series by a fixed rate.
2. Arguments: data, rate.

3. Return: librosa.effects.time_stretch(y=data, rate=0.8)

As shown in Figure 4.9 below, this is the result of stretching the original sound.

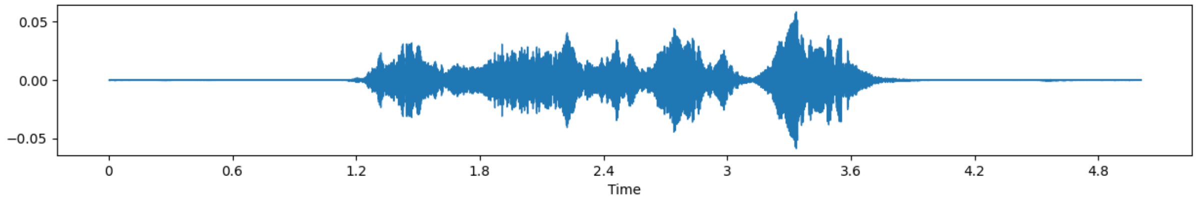


Figure 4. 9 Stretch.

4.2.4.3 Shift

- Shift Function

The "shift(data)" function shifts the audio waveform by a random amount within the given shift range. It creates a new version of the waveform by applying a time shift to the original "data", resulting in a shifted audio signal.

1. Description: Shift the audio wave by fixed shift range.
2. Arguments: data.
3. Return: np.roll(data, shift_range).

As shown in Figure 4.10 below, this is the result of shifting the original sound:

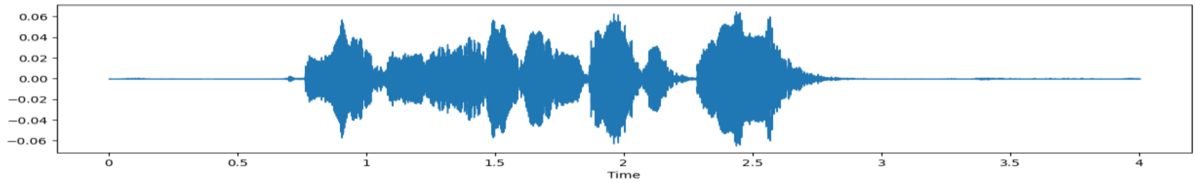


Figure 4. 10 Shift

4.2.4.4 Pitch

- Pitch Function

The "pitch(data, sampling_rate, pitch_factor)" function modifies the pitch of the audio waveform by a specified pitch factor. It creates a new version of the waveform by altering the frequency characteristics of the original "data", resulting in a modified pitch.

1. Description: Shift the pitch of a waveform by n_steps.
2. Arguments: data, sampling_rate, pitch_factor.
3. Return: librosa.effects.pitch_shift(y=data, sr=22050 ,n_steps=pitch_factor).

As shown in Figure 4.11 below, this is the result of pitching the original sound:

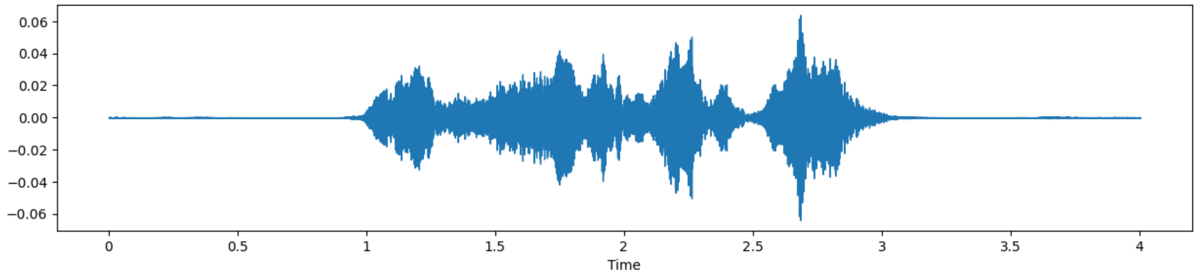


Figure 4. 11 Pitch

4.2.4.5 Higher speed

- higher_speed(data, speed_factor = 1.25)

The "higher_speed(data, speed_factor)" function increases the speed of the audio waveform by a specified factor. It creates a new version of the waveform by adjusting the playback speed of the original "data", resulting in a faster-paced audio signal.

1. Description: Stretch factor by rate> 1, then the signal is sped up.
2. Arguments: data, speed_factor.
3. Return:librosa.effects.time_stretch(y=data, rate=speed_factor).

As shown in the Figure 4.12 below, this is the result of higher speed of the original sound:

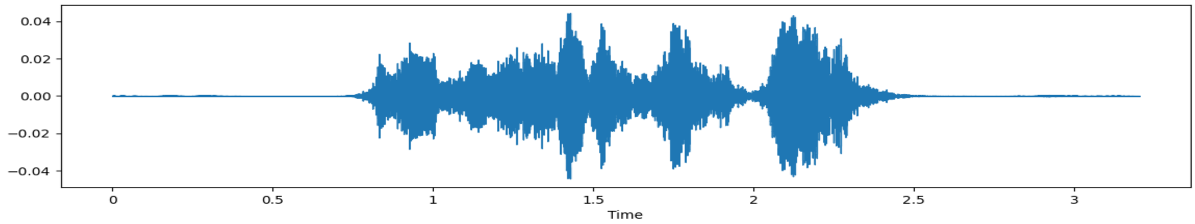


Figure 4. 12 Higher Speed

4.2.4.6 Lower speed

- lower_speed(data, speed_factor = 0.75)

The "lower_speed(data, speed_factor)" function decreases the speed of the audio waveform by a specified factor. It creates a new version of the waveform by adjusting the playback speed of the original "data", resulting in a slower-paced audio signal.

1. Description: Stretch factor by rate < 1, then the signal is slowed down.
2. Arguments: data, speed_factor.
3. Return:librosa.effects.time_stretch(y=data, rate=speed_factor)

As shown in the Figure 4.13 below, this is the result of lower speed of the original sound:

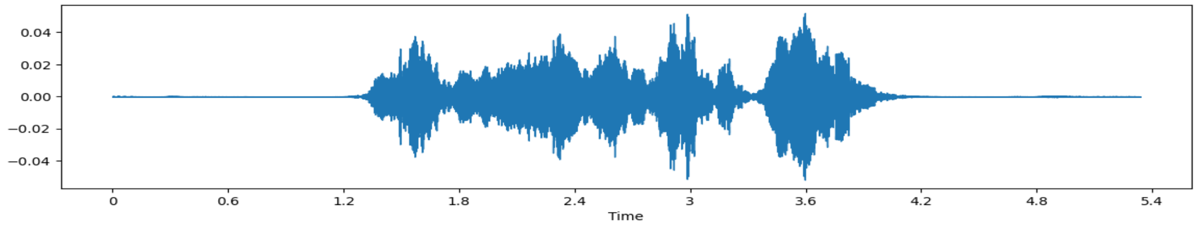


Figure 4. 13 Lower Speed

4.2.5 SER Model Training

In the training section for the SER, the focus is on training a DL model specifically designed for this task. LSTM and CU-DNN-LSTM, identified as the most effective models based on research, are utilized to capture temporal dependencies in speech data. This section highlights the implementation of this architecture and the optimization model parameters for accurate SER model.

4.2.5.1 CU-DNN-LSTM

The CUDLSTM model is a DL model used for training the dataset. It consists of multiple layers, including Cu-DNN-LSTM layers, batch normalization, dropout layers, and dense layers. The model is compiled with a categorical cross-entropy loss function and the Adam optimizer. It is trained on the input data (X_TR) and target labels (Y_TR) for a specified number of epochs and batch size. The model's performance is monitored using validation data (X_VA, Y_VA) and the early stopping callback is applied to prevent overfitting. As shown in Table 4.1 represents CU-DNN-LSTM ’s hyper parameters.

- CU-DNN-LSTM Model Building

  1. Description: Generate CU-DNN-LSTM model.
  2. Arguments: None.
  3. Return: model

- CU-DNN-LSTM Hyper parameters

  1. earlystop=EarlyStopping(monitor='val_accuracy', patience=20, restore_best_weights=True).
  2. Dictionary use of the weights of the classes created to balance the data before.

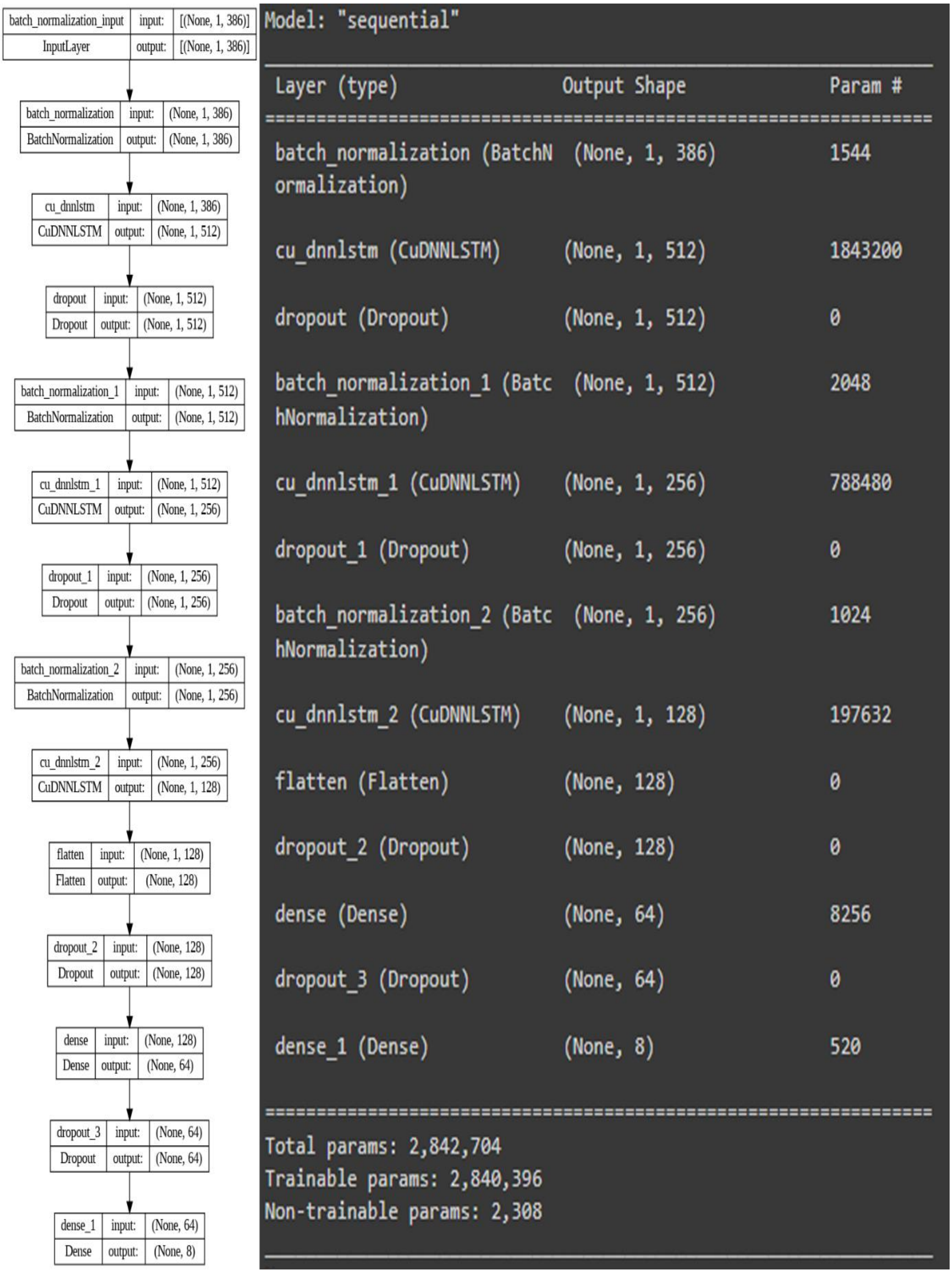
As shown in table 4.1 Cu-DNN-LSTM hyper parameters include: 100 -epochs, 32-batch size, shuffle is True, callbacks (early stop)

Table 4. 1 Cu-DNN-LSTM Hyper parameters

Epochs	Batch size	Shuffle	Callbacks
100	32	True	early stop

- Cu-DNN-LSTM Architecture

As shown in Figure 4.14 below, all the details of the Cu-DNN-LSTM model layers and data shapes:


batch_normalization_input | input: | [(None, 1, 386)]
InputLayer | output: | [(None, 1, 386)]
batch_normalization | input: | (None, 1, 386)
BatchNormalization | output: | (None, 1, 386)
cu_dnnlstm | input: | (None, 1, 386)
CuDNNLSTM | output: | (None, 1, 512)
dropout | input: | (None, 1, 512)
Dropout | output: | (None, 1, 512)
batch_normalization_1 | input: | (None, 1, 512)
BatchNormalization | output: | (None, 1, 512)
cu_dnnlstm_1 | input: | (None, 1, 512)
CuDNNLSTM | output: | (None, 1, 256)
dropout_1 | input: | (None, 1, 256)
Dropout | output: | (None, 1, 256)
batch_normalization_2 | input: | (None, 1, 256)
BatchNormalization | output: | (None, 1, 256)
cu_dnnlstm_2 | input: | (None, 1, 256)
CuDNNLSTM | output: | (None, 1, 128)
flatten | input: | (None, 1, 128)
Flatten | output: | (None, 128)
dropout_2 | input: | (None, 128)
Dropout | output: | (None, 128)
dense | input: | (None, 128)
Dense | output: | (None, 64)
dropout_3 | input: | (None, 64)
Dropout | output: | (None, 64)
dense_1 | input: | (None, 64)
Dense | output: | (None, 8)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 batch_normalization (BatchN  (None, 1, 386)           1544
 ormalization)

 cu_dnnlstm (CuDNNLSTM)      (None, 1, 512)            1843200

 dropout (Dropout)           (None, 1, 512)            0

 batch_normalization_1 (Batc  (None, 1, 512)           2048
 hNormalization)

 cu_dnnlstm_1 (CuDNNLSTM)    (None, 1, 256)            788480

 dropout_1 (Dropout)         (None, 1, 256)            0

 batch_normalization_2 (Batc  (None, 1, 256)           1024
 hNormalization)

 cu_dnnlstm_2 (CuDNNLSTM)    (None, 1, 128)            197632

 flatten (Flatten)           (None, 128)               0

 dropout_2 (Dropout)         (None, 128)               0

 dense (Dense)               (None, 64)                8256

 dropout_3 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 8)                 520

=================================================================
Total params: 2,842,704
Trainable params: 2,840,396
Non-trainable params: 2,308
_________________________________________________________________
```

Figure 4. 14 Cu-DNN-LSTM Architecture

4.2.5.2 BLSTM

The B-LSTM model is a DL architecture used for training the dataset. It comprises multiple layers, including Bidirectional LSTM layers, batch normalization, dropout layers, and dense layers. The model is compiled with a categorical cross-entropy loss function and the Adam optimizer. Training is performed on the input data (X_TR) and target labels (Y_TR) for a specified number of epochs and batch size. The model's performance is monitored using validation data (X_VA, Y_VA), and the early stopping callback is implemented to prevent overfitting. Additionally, class weights are considered during training to address class imbalance. Table 4.2 represents model’s hyper parameters.

- BLSTM Model Building
  1. Description: Generate B_LSTM model.
  2. Arguments: None.
  3. Return: model.

- BLSTM Hyper parameters
  1. earlystop=EarlyStopping (monitor='val_accuracy', patience=20, restore_best_weights=True).
  2. Dictionary use of the weights of the classes created to balance the data before.

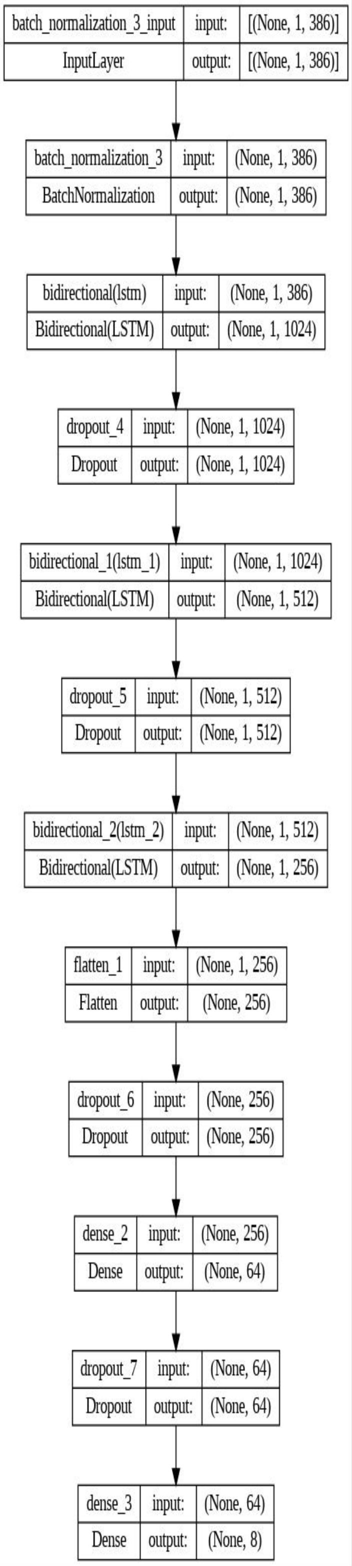
As shown in table 4.2 B-LSTM hyper parameters include: 50-epochs, 64-batch size, shuffle is True, callbacks (early stop)

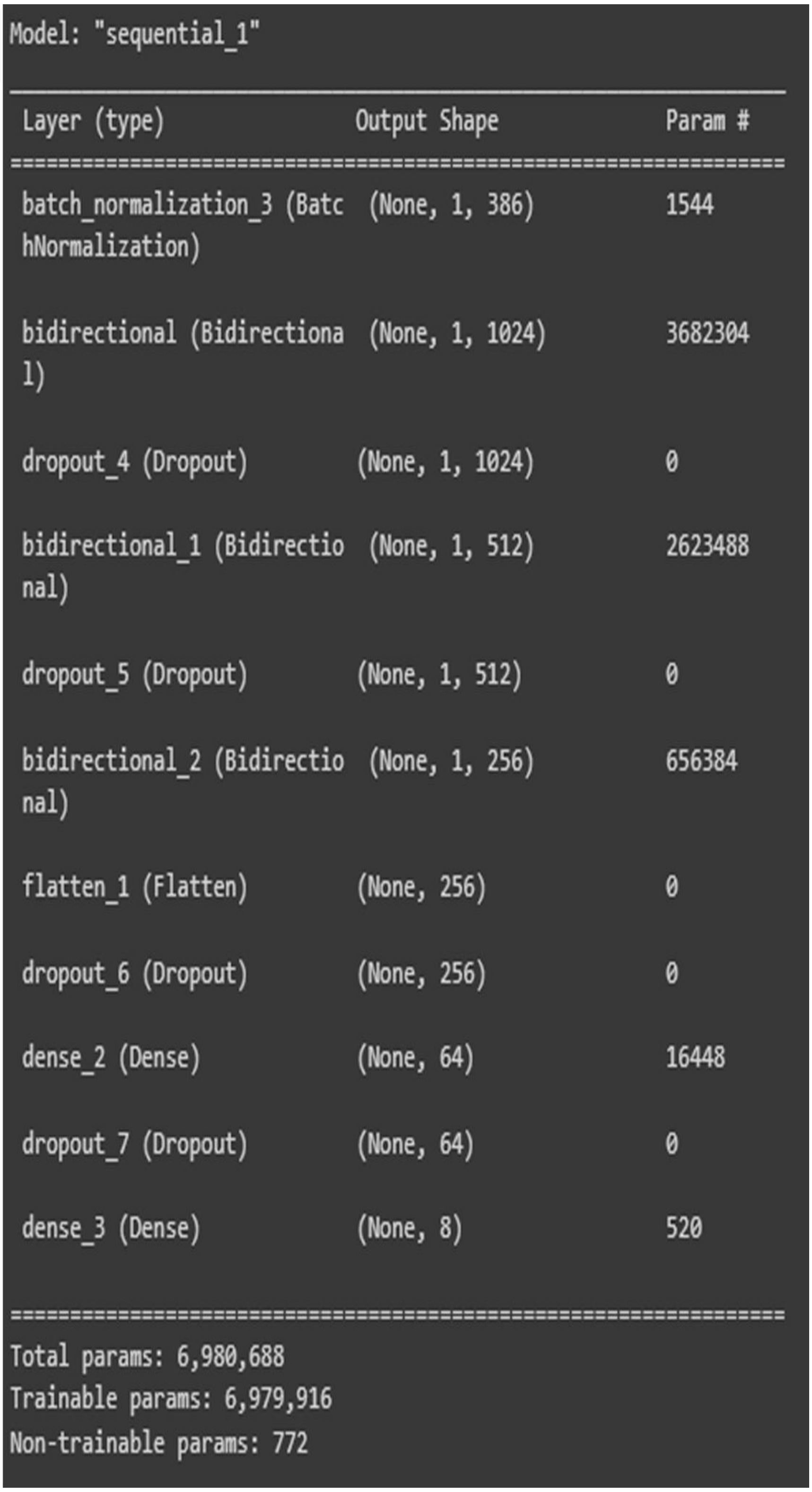
Table 4. 2 B-LSTM Hyper Parameters

Epochs	Batch size	Shuffle	Callbacks
50	64	True	early stop

- B-LSTM -Architecture

As shown in the Figure 4.15 below, all the details of the B-LSTM model layers and data shapes


batch_normalization_3_input | input: | [(None, 1, 386)]
InputLayer | output: | [(None, 1, 386)]
batch_normalization_3 | input: | (None, 1, 386)
BatchNormalization | output: | (None, 1, 386)
bidirectional(lstm) | input: | (None, 1, 386)
Bidirectional(LSTM) | output: | (None, 1, 1024)
dropout_4 | input: | (None, 1, 1024)
Dropout | output: | (None, 1, 1024)
bidirectional_1(lstm_1) | input: | (None, 1, 1024)
Bidirectional(LSTM) | output: | (None, 1, 512)
dropout_5 | input: | (None, 1, 512)
Dropout | output: | (None, 1, 512)
bidirectional_2(lstm_2) | input: | (None, 1, 512)
Bidirectional(LSTM) | output: | (None, 1, 256)
flatten_1 | input: | (None, 1, 256)
Flatten | output: | (None, 256)
dropout_6 | input: | (None, 256)
Dropout | output: | (None, 256)
dense_2 | input: | (None, 256)
Dense | output: | (None, 64)
dropout_7 | input: | (None, 64)
Dropout | output: | (None, 64)
dense_3 | input: | (None, 64)
Dense | output: | (None, 8)



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 batch_normalization_3 (Batc  (None, 1, 386)           1544
 hNormalization)

 bidirectional (Bidirectiona  (None, 1, 1024)          3682304
 l)

 dropout_4 (Dropout)         (None, 1, 1024)           0

 bidirectional_1 (Bidirectio  (None, 1, 512)           2623488
 nal)

 dropout_5 (Dropout)         (None, 1, 512)            0

 bidirectional_2 (Bidirectio  (None, 1, 256)           656384
 nal)

 flatten_1 (Flatten)         (None, 256)               0

 dropout_6 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 64)                16448

 dropout_7 (Dropout)         (None, 64)                0

 dense_3 (Dense)             (None, 8)                 520

=================================================================
Total params: 6,980,688
Trainable params: 6,979,916
Non-trainable params: 772
```

Figure 4. 15 B-LSTM Architecture

4.2.6 SER Model Results

The SER model results showcase a comprehensive evaluation of the system's performance, achieved through a meticulous process of data loading, balancing, and preprocessing. The dataset was carefully split into train, test, and validation sets, maintaining an 80% - 10% - 10% distribution. Various combinations of feature extraction techniques were employed, incorporating both low-level features such as MFCC, ZCR, RMS, Chroma, Mel-spectrogram, Contrast, and Tonnetz, as well as high-level features including mean and standard deviation. Multiple DL models, including MLP, CU-DNN-LSTM, B-LSTM, LSTM, and CNN, were utilized with different hyperparameters, with a rigorous grid search employed to identify the optimal model parameters. Furthermore, various data augmentation techniques, such as noise addition, stretching, shifting, pitch modification, and speed adjustments, were applied to enhance the dataset's diversity. The results of this extensive study have been meticulously recorded and presented in detailed Table 4.3, Table 4.4, Table 4.5, and Table 4.6, providing a comprehensive overview of the performance achieved by the SER models across different datasets, feature sets, DL architectures, and data augmentation strategies.

The following tables present a comprehensive view of the results obtained from the extensive evaluation of the SER models. These tables provide an organized summary of the model performance across different datasets, feature sets, DL architectures, and data augmentation techniques. By examining these tables, one can gain valuable insights into the effectiveness of various models and approaches, enabling a deeper understanding of the performance achieved in SER tasks. Overall, the tables provide insights into the performance of different models, feature combinations, and augmentation techniques for SER. They highlight the strengths and weaknesses of each approach and provide guidance for selecting the most effective configurations for this task.

Table 4. 3 SER Performance Summary - Trial 1

Features	Augmentation	Model	Train Accuracy %	Validation Accuracy %	Test Accuracy %
Mfcc Chroma mel spectrogram	noise stretch shift pitch	CNN LSTM	70 92	79 79	62 50
Mfcc Chroma mel spectrogram	shift pitch speed up	CNN LSTM	65 62	70 67	53 41
Mfcc Chroma mel spectrogram	stretch shift	CNN LSTM	67 92	81 97	53 42
Mfcc Chroma mel spectrogram	Noise Stretch Shift Pitch Speedup speed down	CNN LSTM	67 97	78 98	53 51
Mfcc Chroma mel spectrogram	Stretch Pitch speed down	CNN LSTM	59 96	71 97	53 39
Mfcc Chroma mel spectrogram	NO	LSTM	95	97	41
Chroma mel spectrogram rms	Noise Stretch shift pitch	CNN	99	90	60
Chroma mel spectrogram rms	Shift pitch speed up	CNN	97.5	77.8	46.6
Chroma mel spectrogram rms	Stretch shift	CNN	97.8	83.8	47
Chroma mel spectrogram rms	noise stretch shift pitch speed up. speed down	CNN	99.9	90.8	58.6
Chroma mel spectrogram rms	Stretch pitch speed down	CNN	99	84	46.1

Table 4.3 provides insights about different models exhibit high train accuracy (up to 99.9%) and decent validation accuracy (up to 98%). However, test accuracies vary between 39% and 62%, indicating room for improvement in performance on unseen data. Further investigation, including hyperparameter tuning and exploring different features and augmentations, is needed to optimize the models and enhance test accuracy.

Table 4. 4 SER Performance Summary - Trial 2

Features	Augmentation	Model	Train accuracy	Validation accuracy	Test accuracy
MFCC ZCR	NO	CNN	100	52	58
MFCC ZCR	Noise Stretch shift pitch	CNN	99.87	88.37	65.28
MFCC ZCR	Noise Stretch Shift Pitch	LSTM	73.07	69.4	40.62
MFCC ZCR	Shift Pitch speed up	CNN	100	94.90	63.19
MFCC ZCR	Shift Pitch speed up	LSTM	96.28	85.68	48.96
MFCC ZCR	Stretch Shift	CNN	99.97	94.36	63.54
MFCC ZCR	Stretch Shift	LSTM	86.41	77.33	48.26
MFCC ZCR	Noise Stretch shift pitch speed up. speed down	CNN	99.92	92.99	65.28
MFCC ZCR	Noise Stretch Shift Pitch speed up. speed down	LSTM	92.16	88.96	50.69

MFCC ZCR	Stretch Pitch speed down	CNN	100	95.44	62.15
MFCC ZCR	Stretch Pitch speed down	LSTM	92.70	80.69	48.26
MFCC Chroma Mel-Spectrogram	NO	CNN	68.58	41.99	44.44
MFCC Chroma Mel-Spectrogram	NO	LSTM	27.47	31.17	35.42

Table 4.4 provides insights about different models based on MFCC and ZCR features with various augmentations show varying accuracies. CNN consistently outperforms LSTM. Augmentations like noise, stretch, shift, and pitch generally improve performance. The CNN model with MFCC, ZCR, and noise, stretch, shift, and pitch achieves the highest test accuracy of 65.28%. Further optimization is needed for certain feature-augmentation combinations.

Table 4. 5 SER Performance Summary - Trial 3

Features	Augmentation	Model	Train accuracy	Validation accuracy	Test accuracy
MFCC	NO	CNN	93.8	68.32	60.42
MFCC	Noise Shift pitch	CNN	94	74	69.5
MFCC	Shift pitch	CNN	98.76	69.8	67.36
MFCC	Noise Pitch	CNN	94.79	68.81	65.51
MFCC	Noise Shift	CNN	92.8	67.82	67.13

Table 4.5 provides insights about CNN model trained on MFCC features benefits from augmentations like noise, shift, and pitch. Without augmentation, the model achieves a train accuracy of 93.8%, validation accuracy of 68.32%, and test accuracy of 60.42%. With augmentations, the model improves, reaching a test accuracy of 69.5%. Careful selection and application of augmentations enhance the model's ability to generalize and improve accuracy on unseen data.

Table 4. 6 SER Performance Summary - Trial 4

Model	Data	Features	Augmentation	Train Accuracy	Valid Accuracy	Test Accuracy
MLP	RAVDESS	MFCC, ZCR, RMS, Chroma, Mel-spectrogram	Yes	-	-	47 %
CNN	RAVDESS	MFCC, ZCR, RMS.	Yes	99%	-	71 %
GRU	RAVDESS	LLDs: MFCC, Chroma, Mel-spectrogram, Contrast, Tonnetz. HLDs: Mean, Standard.	No	98 %	-	77 %
GRU	RAVDESS TESS	LLDs: MFCC, Chroma, Mel-spectrogram, Contrast, Tonnetz. HLDs: Mean, Standard.	Yes	96 %	88 %	88 %
LSTM	RAVDESS TESS	LLDs: MFCC, Chroma, Mel-spectrogram, Contrast, Tonnetz. HLDs: Mean, Standard.	Yes	96 %	87 %	89 %
B_LSTM	**RAVDESS** **TESS**	**LLDs: MFCC, Chroma, Mel-spectrogram, Contrast, Tonnetz.** **HLDs: Mean, Standard.**	**yes**	**98.78 %**	**91 %**	**92 %**
Cu-DNN-LSTM	**RAVDESS** **TESS**	**LLDs: MFCC, Chroma, Mel-spectrogram, Contrast, Tonnetz.** **HLDs: Mean, Standard.**	**Yes**	**97.01 %**	**94 %**	**91 %**

Table 4.6 showcases various models' performance in terms of test accuracies. The MLP model achieved a test accuracy of 47%, the CNN model achieved 71% test accuracy, and the GRU model achieved a test

accuracy of 77%. When trained on both RAVDESS and TESS datasets, the GRU, LSTM, Cu-DNN-LSTM, and B_LSTM models achieved higher test accuracies ranging from 88% to 94%.

4.2.7 SER Model Results Discussion

The results presented in the previous Tables 4.3-4.6, Table 4.3 provide a valuable insight into the performance of various feature combinations, augmentation techniques, and DL models for SER. The CNN model achieved high training accuracy but showed signs of overfitting, as its performance on the validation and test sets was not as strong. In contrast, the LSTM model demonstrated consistent performance, indicating good generalization capabilities and resistance to overfitting. Table 4.4 further highlighted the potential issue of memorization without effective generalization in the CNN model when using specific features. Additionally, Table 4.5 emphasized the CNN model's effectiveness in capturing spatial patterns from spectrogram-based features, achieving competitive results with different feature-augmentation combinations.

Lastly, Table 4.6 showcased the superiority of the **Cu-DNN-LSTM** model, consistently outperforming other models across datasets and features, emphasizing its suitability for SER tasks by effectively capturing temporal dependencies within the data. These findings provide valuable guidance for selecting appropriate models, features, and augmentation techniques in SER applications.

- SER performance with other systems

As shown in Figure 4.16 below, the performance of the SER system has been evaluated in comparison to other existing systems, showcasing its higher accuracy in the past four years. This evaluation highlights the superior performance of the SER system in accurately recognizing and classifying emotions from speech signals when compared to other competing systems. The consistent achievement of higher accuracy reinforces the effectiveness and reliability of the SER system as a powerful tool in emotion analysis and understanding.

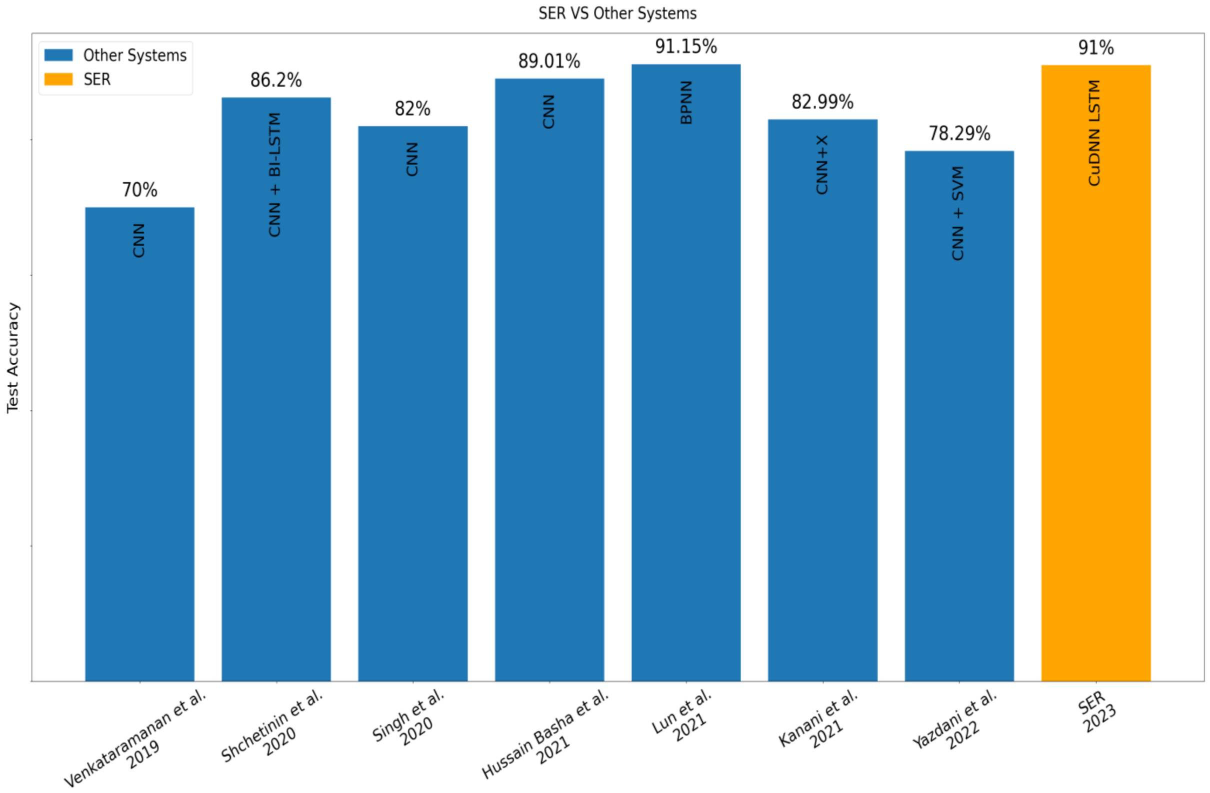


Figure 4. 16 SER Performance with other systems

- Cu-DNN-LSTM confusion matrix

As shown in Figure 4.17 CU-DNN-LSTM confusion matrix represents the performance of a CU-DNN-LSTM model for emotion classification. With an accuracy of 91%, the SER model demonstrates good overall performance. The precision, recall, and F1-scores range from 0.75 to 0.95, indicating the model's ability to correctly predict different emotions. The "sad" class has the highest F1-score, while the "calm" class has the lowest. The weighted average F1-score of 0.91 suggests balanced performance across classes, considering their distribution in the dataset. Overall, the CU-DNN-LSTM model shows promise in accurately classifying emotions, but further analysis is needed to understand misclassifications and potential biases.

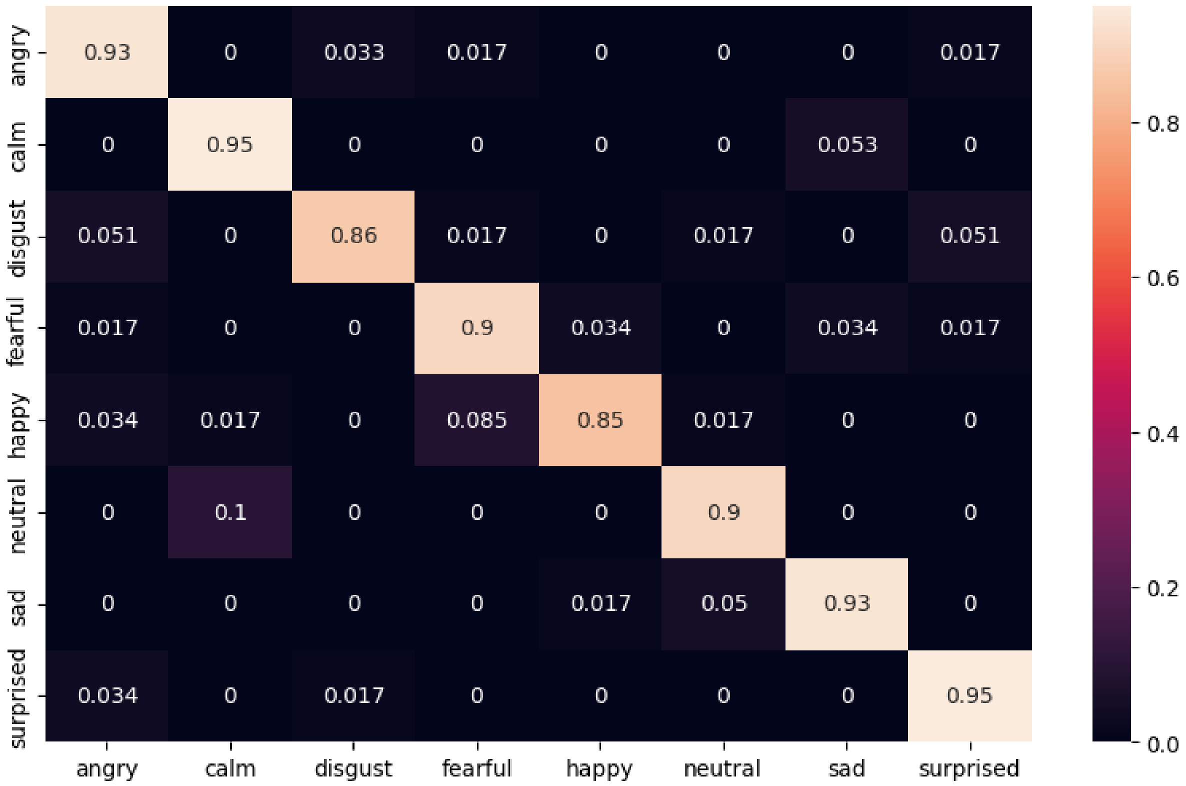


Figure 4. 17 CU-DNN-LSTM Confusion-Matrix

- BLSTM confusion matrix

As shown in Figure 4.18 BLSTM confusion matrix and classification report reveals that the SER system achieved an overall accuracy of 92% for the classification of different emotions. The system showed high precision and recall for emotions like anger, disgust, and surprise, while emotions such as calm had slightly lower precision and recall scores. The weighted average F1-score of 0.92 indicates a balanced performance across all emotion categories. These results demonstrate the effectiveness of the system in accurately identifying emotions, but further analysis is needed to address potential areas for improvement.

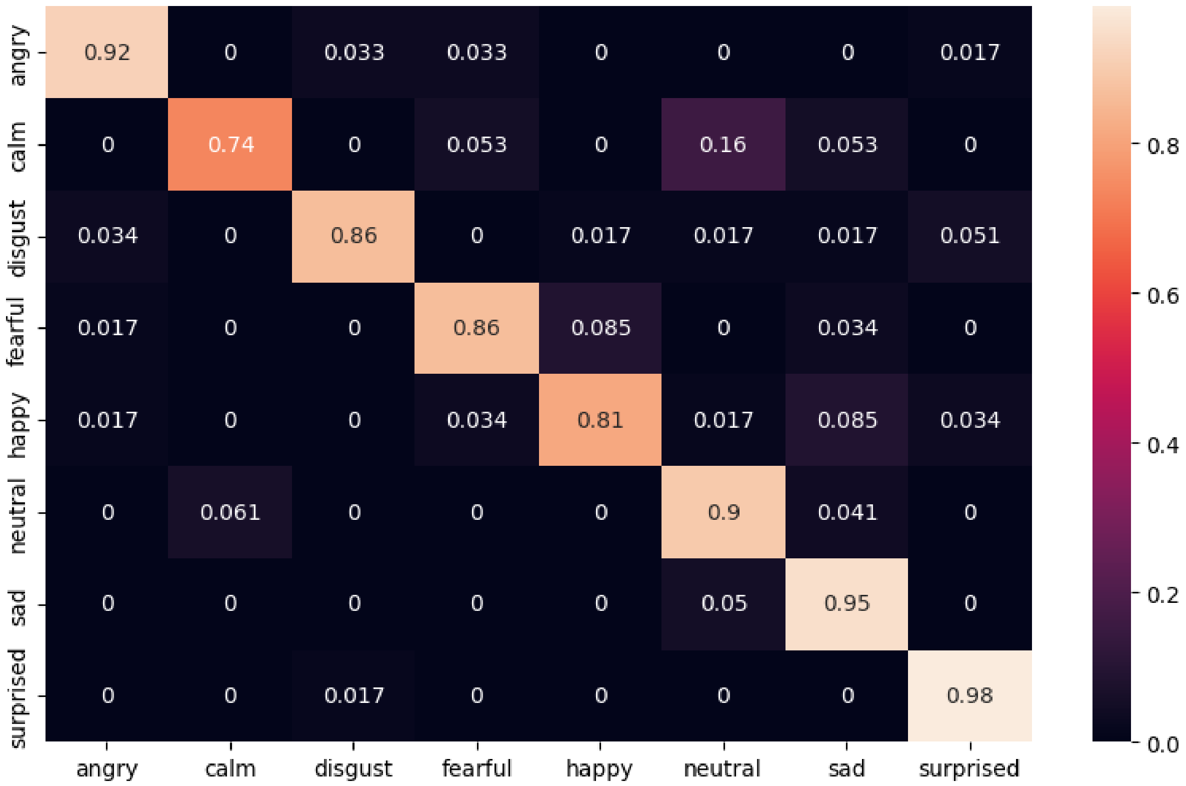


Figure 4. 18 B- LSTM Confusion-Matrix

4.2.8 User Interface (UI)

The UI implementation in SER involves designing and developing a user-friendly interface that allows users to interact with the SER system. The UI includes features such as audio recording capabilities, text input fields, and visual representations of detected emotions. Efforts are made to ensure compatibility with various devices and optimize the UI for different screen sizes. The goal is to enhance the user experience by providing clear instructions, real-time feedback, and visually appealing elements. Overall, the UI implementation in SER aims to create an intuitive and engaging interface that enables users to conveniently input speech data and obtain emotion recognition results.

- Start Recording Function

The "start_recording()" function is responsible for initiating real-time audio recording without any arguments and does not return any value.

1. Description: call record audio for real time recording.
2. Arguments: None.

3. Return: None.

- Record Audio Function

The "record_audio()" function is used to capture the user's voice and update the time stamp list. It doesn't require any arguments and doesn't return any value.

1. Description: Record voice from user and update time stamp list.
2. Arguments: None.
3. Return: None.

Chapter 5

User Manual

Chapter 5

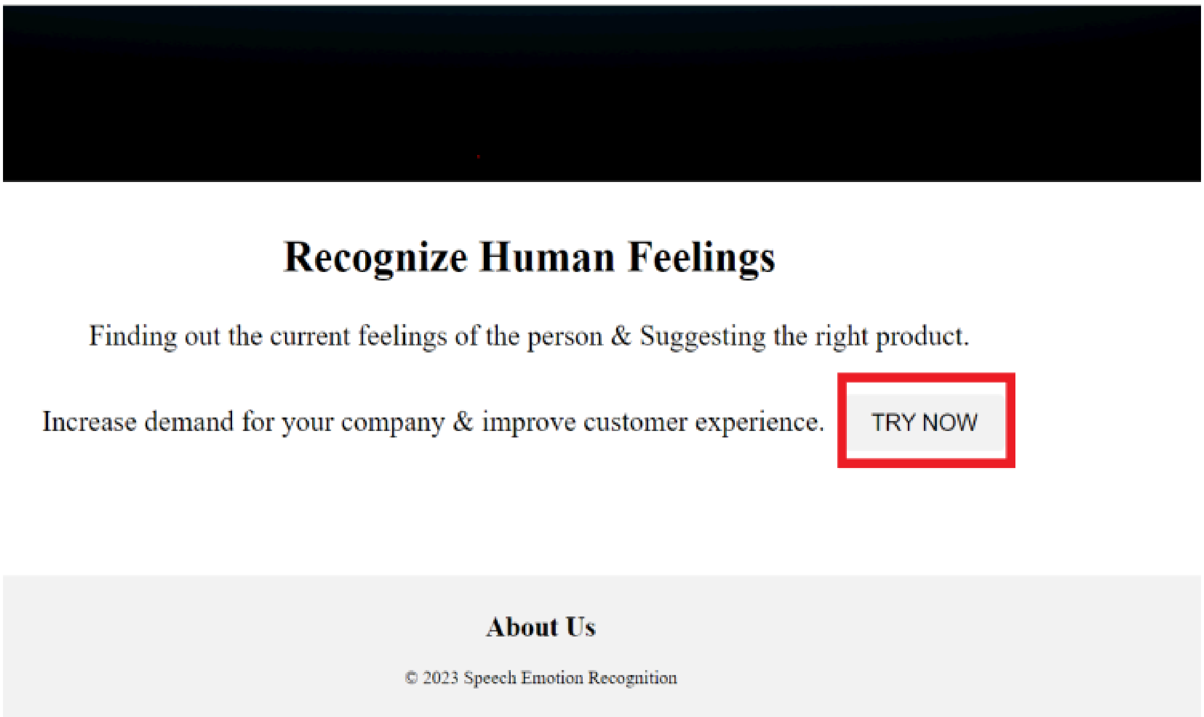
User Manual

The user manual provides guidance on how to effectively utilize the SER system through both the website and mobile app platforms. It offers detailed instructions on the system's functionalities, including features such as audio recording, emotion analysis, and result visualization. Users will gain a comprehensive understanding of how to interact with the SER system, enabling them to leverage its capabilities for accurate and efficient emotion recognition.

5.1 Website Application

This user manual provides an overview of the website app that records users' voices and detects their emotions in real time. The manual covers the following steps:

- Open the website link!
- To start the process, click on the "Try Now" button as shown in Figure 5.1.



Recognize Human Feelings

Finding out the current feelings of the person & Suggesting the right product.

Increase demand for your company & improve customer experience.

About Us

© 2023 Speech Emotion Recognition

Figure 5. 1 Website Home Page

- To complete the registration process as shown in Figure 5.2, you need to provide the following information in the registration form:

  1. Full name.
  2. Email address.
  3. Age.
  4. Select your gender.

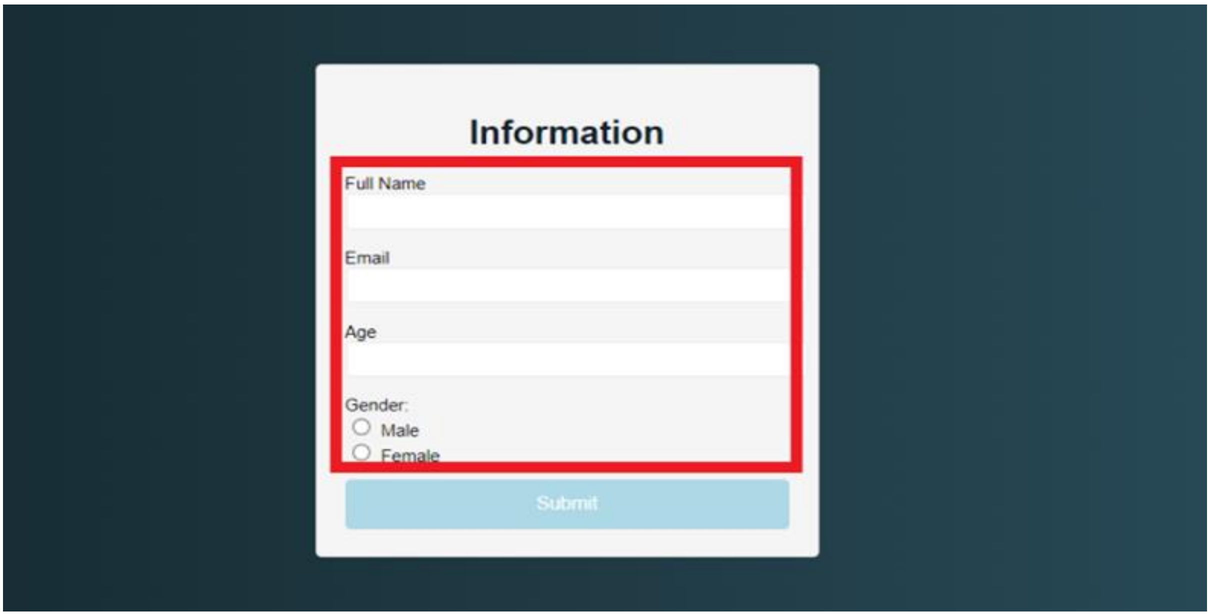
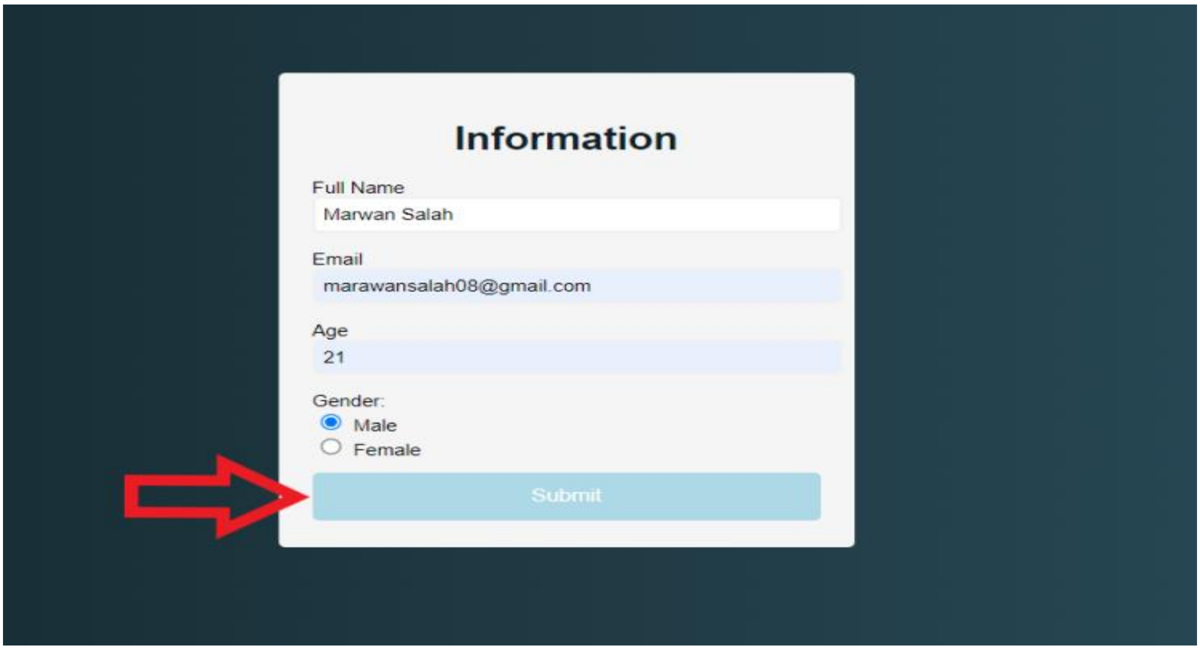


Figure 5. 2 User information entry page

- To complete the registration process as shown in Figure 5.3 below, click on the "Submit" button:



Information

Full Name
Marwan Salah

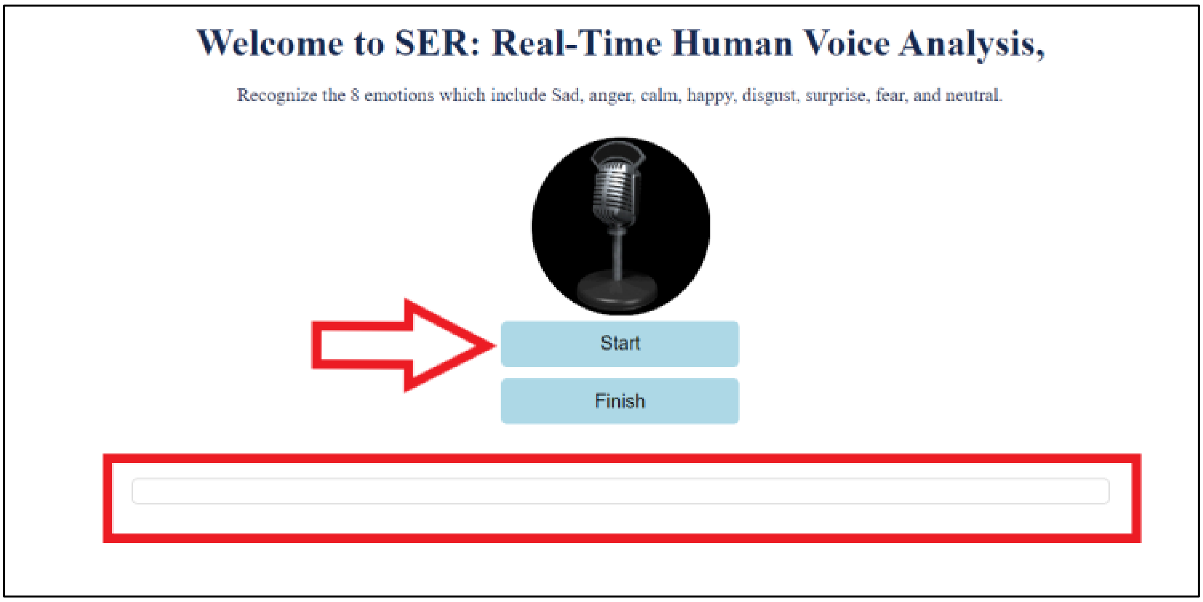
Email
marawansalah08@gmail.com

Age
21

Gender:
● Male
○ Female

Figure 5. 3 User Submit Page

- When you click on the "Start" button for real-time emotion recognition as shown in Figure 5.4, the system should initiate the process of analyzing emotions based on the input it receives. Here's what you will typically see as a result:

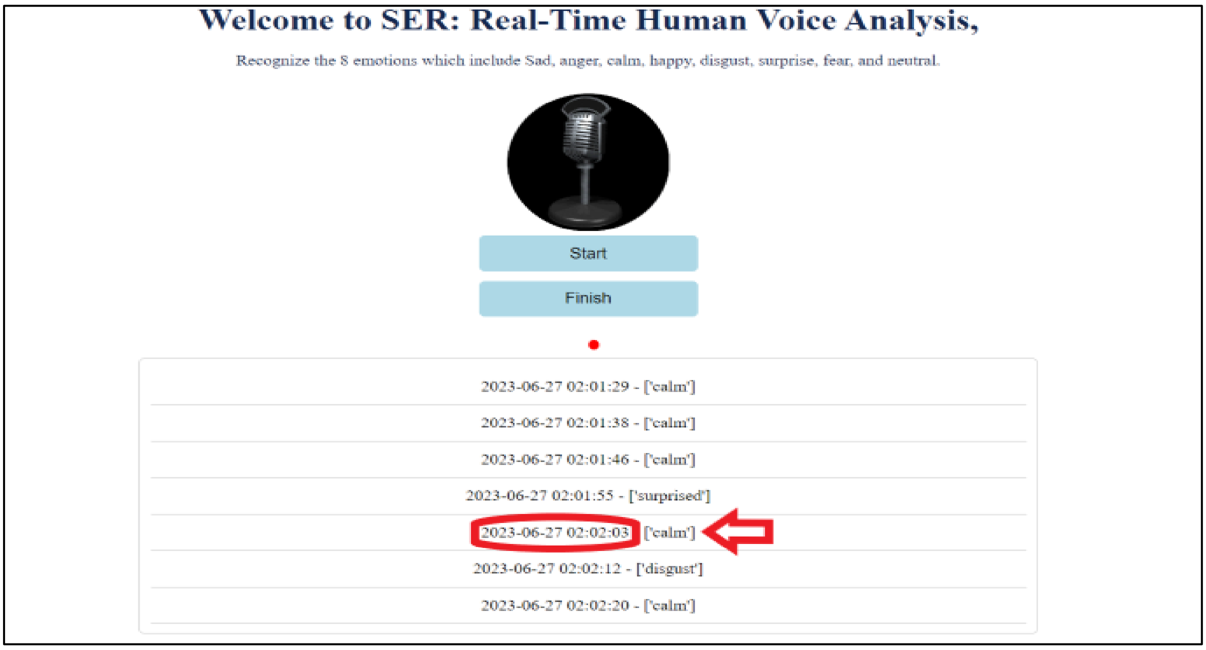


Welcome to SER: Real-Time Human Voice Analysis,

Recognize the 8 emotions which include Sad, anger, calm, happy, disgust, surprise, fear, and neutral.

Figure 5. 4 SER Real-Time Page

- Below, you can find the displayed results which include the current time and the corresponding emotion associated with that time as shown in Figure 5.5:



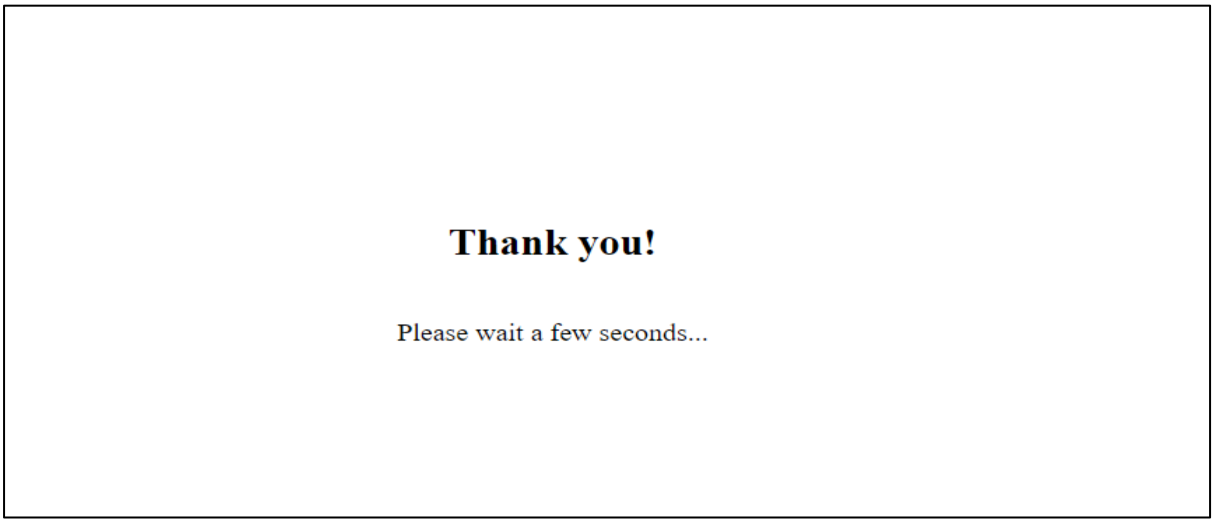
Welcome to SER: Real-Time Human Voice Analysis,

Recognize the 8 emotions which include Sad, anger, calm, happy, disgust, surprise, fear, and neutral.

```
2023-06-27 02:01:29 - ['calm']
2023-06-27 02:01:38 - ['calm']
2023-06-27 02:01:46 - ['calm']
2023-06-27 02:01:55 - ['surprised']
2023-06-27 02:02:03 ['calm']
2023-06-27 02:02:12 - ['disgust']
2023-06-27 02:02:20 - ['calm']
```

Figure 5. 5 SER Real-Time Page Results

- If you choose to click "Finish" button you will be redirected to the home page as the final step as shown in Figure 5.6 below:



Thank you!

Please wait a few seconds...

Figure 5. 6 SER Website End Page

5.2 Mobile Application

This user manual provides an overview of the mobile app that records users' voices and detects their emotions in real time. The manual covers the following steps:

- Install the mobile application.
- Open the app to access the interface as shown in Figure 5.7.

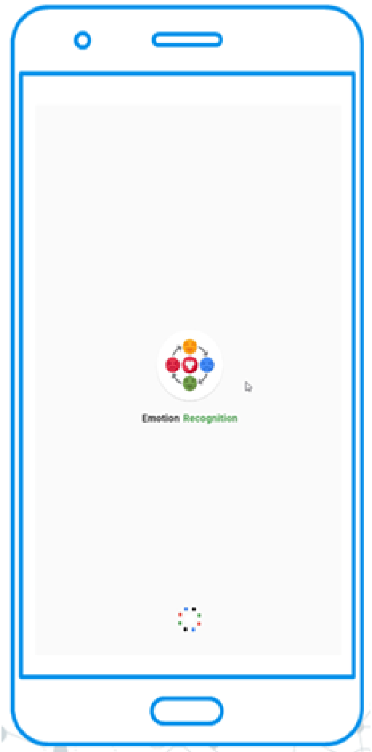
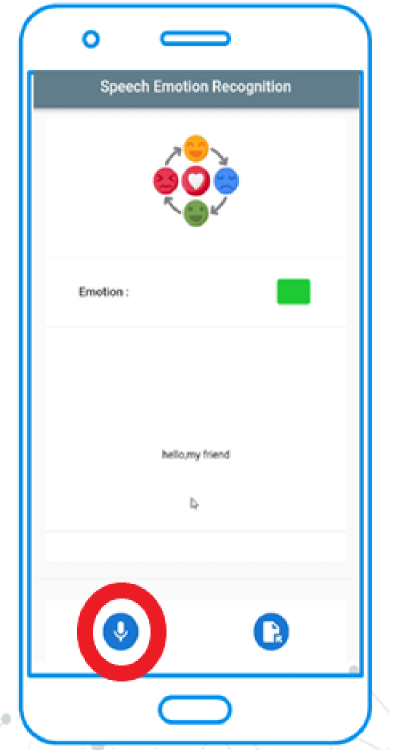


Figure 5. 7 SER Mobile Application Loading Page

- Tap "Start recording" to start recording your speech as shown in Figure 5.8.

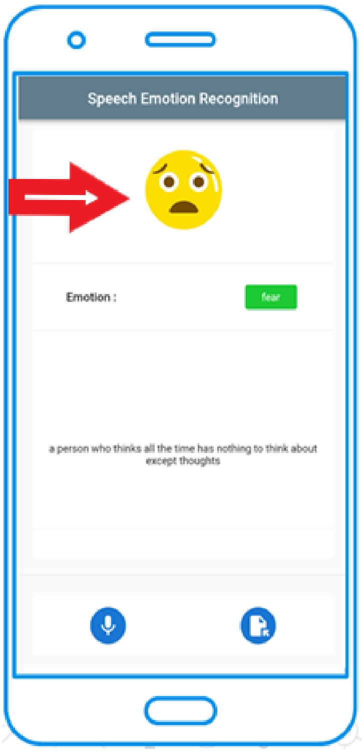


Emotion :

hello,my friend

Figure 5. 8 SER Mobile Application Main Page

- Tap "Stop recording" to stop recording and see the detected emotion as shown in Figure 5.9.

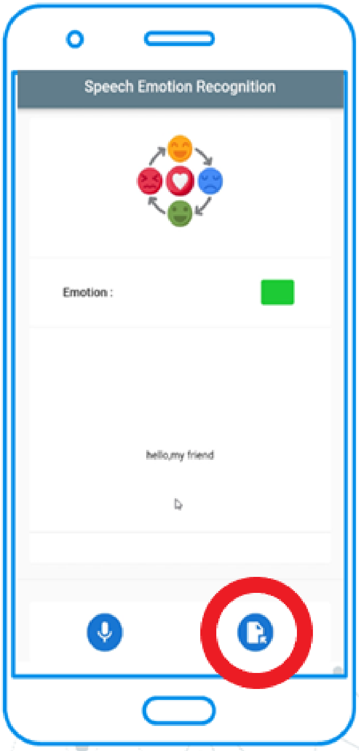


Emotion : fear

a person who thinks all the time has nothing to think about except thoughts

Figure 5. 9 SER Mobile Application Emotion Result Page

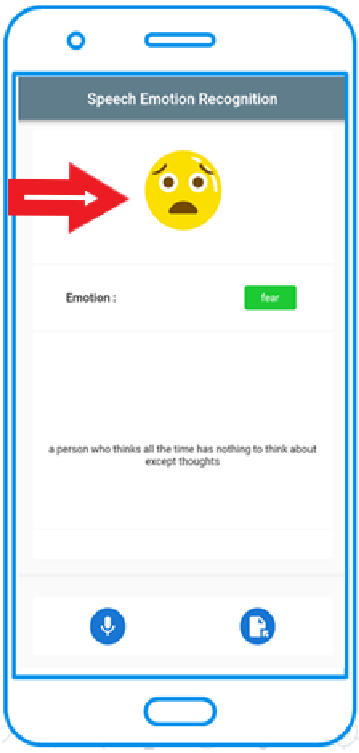
- Tap "Upload File" to upload an audio file and see the detected emotion as shown in Figure 5.10 below.



hello,my friend

Figure 5. 10 SER Mobile Application Loading Voice File Page

- View the detected emotion and corresponding transcript in the app as shown in Figure 5.11.



a person who thinks all the time has nothing to think about except thoughts

Figure 5. 11 SER Mobile Application Emotion Result Page

Chapter 6

Conclusion and Future Work

Chapter 6

Conclusion and Future Work

This section presents the concluding remarks of the SER system and outlines potential avenues for future work.

6.1 Conclusion

SER field is advancing rapidly and has significant implications across various domains. This research emphasizes the progress in developing SER models and their potential applications in areas such as HCI, mental health assessment, and affective computing. The high accuracies achieved, 91% for the CU-DNN-LSTM model and 92% for the BLSTM model, demonstrate their effectiveness in accurately recognizing emotions from speech signals. However, challenges related to dataset diversity, subjective emotion interpretation, and real-time emotion tracking need to be addressed to improve the precision, robustness, and cultural sensitivity of SER models.

This study highlights the successful implementation of SER models in accurately detecting emotions from speech signals. The results demonstrate notable accuracy and performance metrics, findings underscore the potential of SER to enable machines to understand and respond to users' emotional states, facilitating personalized and empathetic interactions.

However, challenges within the field of SER remain, including limited dataset diversity, the subjective nature of emotion interpretation, and the need for real-time emotion tracking. Further exploration and improvements in these areas are essential to enhance the precision, robustness, and cultural sensitivity of SER models. Addressing these challenges will contribute to the advancement of SER, empowering the development of more accurate and reliable models that can better capture and interpret emotions in various contexts.

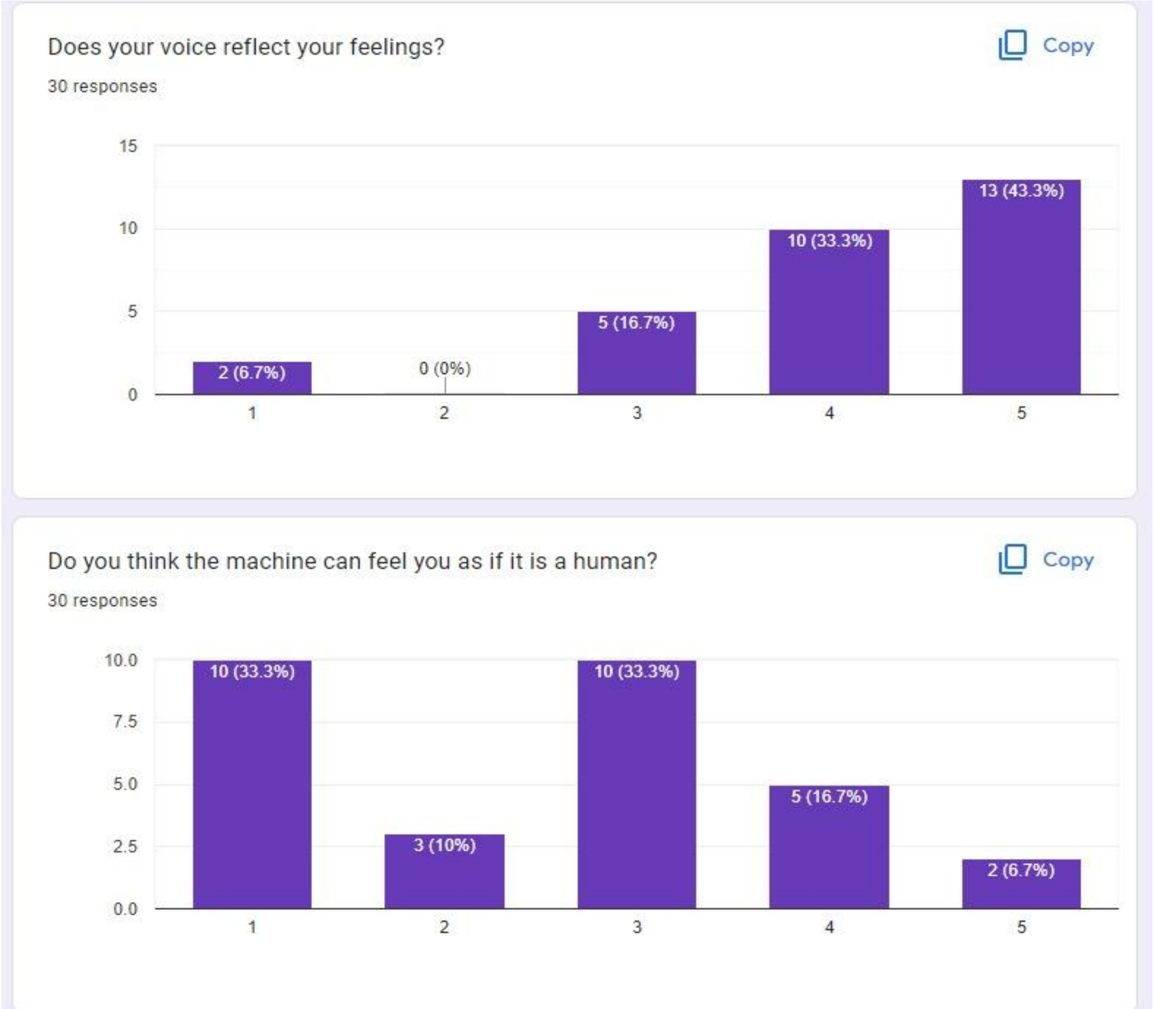
6.2 Future Work

To improve the proposed idea, several avenues can be explored in future work. Firstly, the inclusion of a more extensive and diverse dataset can offer richer and more representative audio samples, leading to improved performance and generalization. Additionally, incorporating text data alongside phonetic information can provide valuable contextual cues for more accurate emotion recognition. Exploring and implementing state-of-the-art DL models specifically designed for audio analysis can further enhance the system's capabilities and accuracy. Moreover, novel preprocessing techniques can be investigated to enhance the quality and relevance of the input data, such as denoising, normalization, or feature augmentation. The exploration of various feature extraction techniques and incorporating newly discovered features can also contribute to the system's robustness and discriminative power. Finally, expanding the application of the system to real-life scenarios and diverse user populations can provide valuable insights and validate its effectiveness in practical settings.

Appendix

- Survey

The Google Form survey conducted as part of the SER Graduation Project yielded valuable insights and feedback from the participants. The survey results provided a comprehensive understanding of users' perceptions, experiences, and satisfaction with the implemented system. The feedback collected will contribute to further improvements and refinements of the project, ensuring its alignment with user needs and expectations. The survey results serve as a valuable resource for evaluating the system's effectiveness and gathering user perspectives for future enhancements and iterations as shown in Figure 6.1 - Figure 6.2 - Figure 6.3 – Figure 6.4.



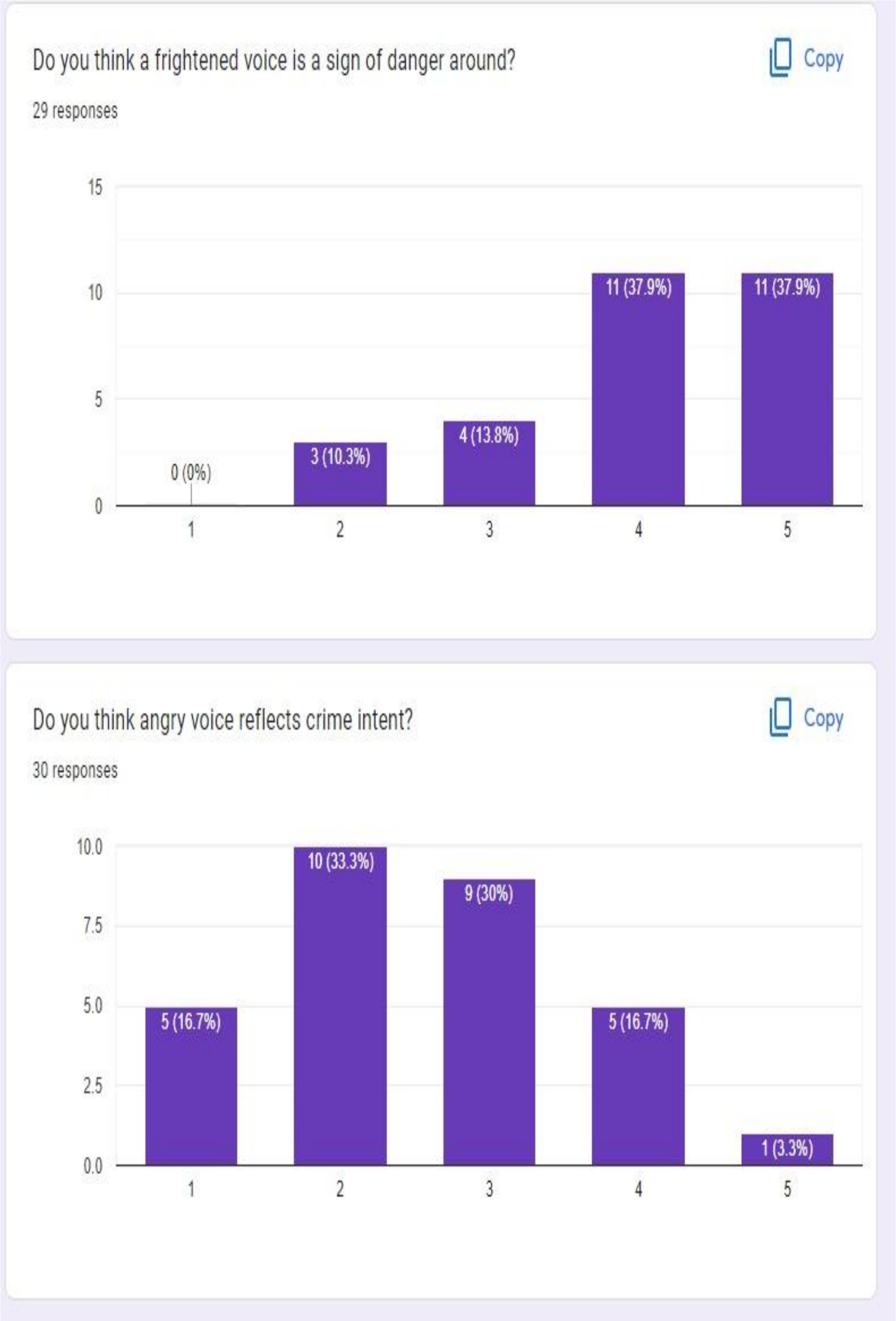
Does your voice reflect your feelings?

30 responses

Do you think the machine can feel you as if it is a human?

30 responses

Figure 6. 1 Survey.1



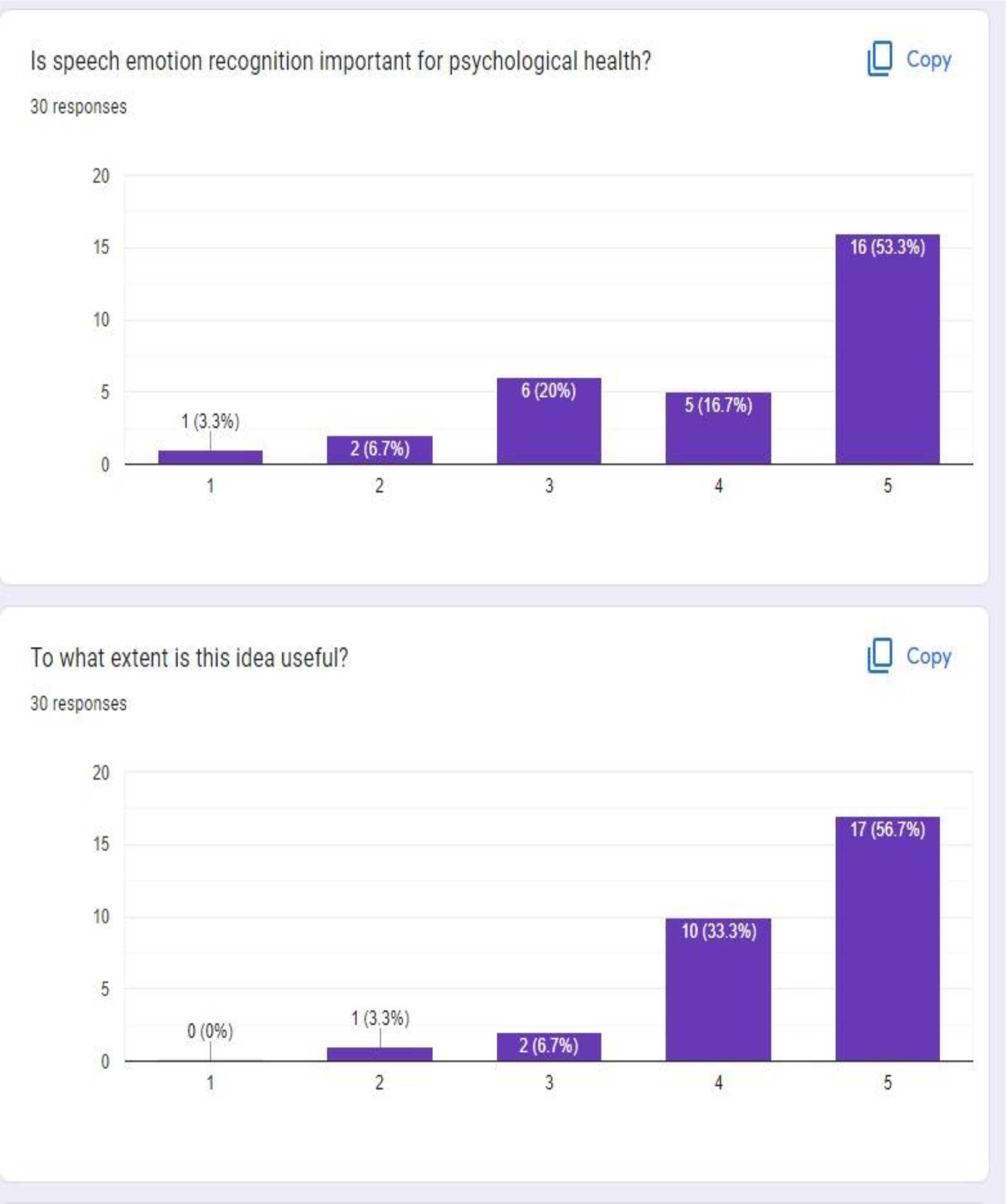
Do you think a frightened voice is a sign of danger around?

29 responses

Do you think angry voice reflects crime intent?

30 responses

Figure 6. 2 Survey.2



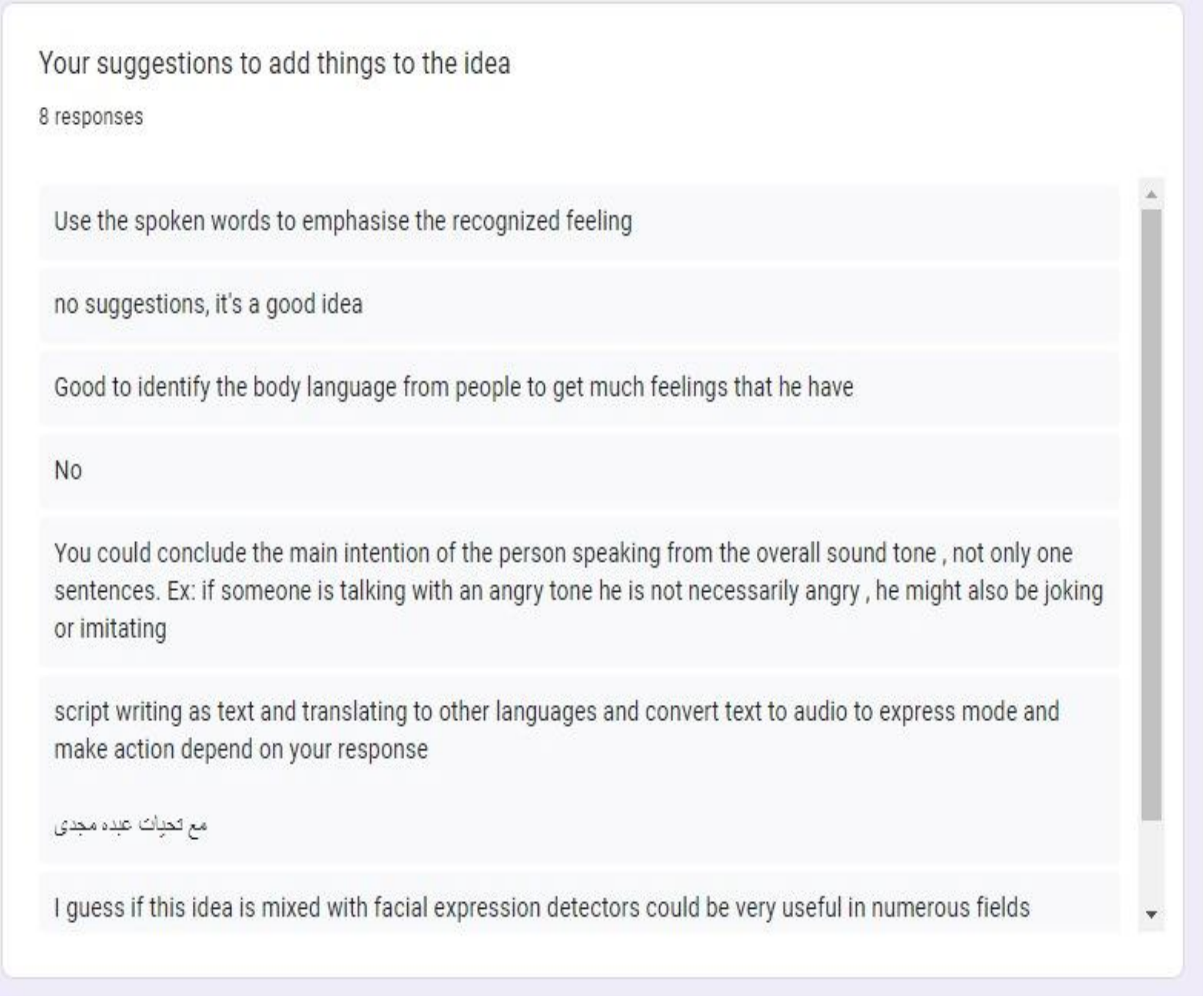
Is speech emotion recognition important for psychological health?

30 responses

To what extent is this idea useful?

30 responses

Figure 6. 3 Survey.3



Your suggestions to add things to the idea

8 responses

Use the spoken words to emphasise the recognized feeling

no suggestions, it's a good idea

Good to identify the body language from people to get much feelings that he have

No

You could conclude the main intention of the person speaking from the overall sound tone , not only one sentences. Ex: if someone is talking with an angry tone he is not necessarily angry , he might also be joking or imitating

script writing as text and translating to other languages and convert text to audio to express mode and make action depend on your response

مع تحيات عبده مجدى

I guess if this idea is mixed with facial expression detectors could be very useful in numerous fields

Figure 6. 4 Survey.4

References

[1] A. Chiurco *et al.*, "Real-time Detection of Worker's Emotions for Advanced Human-Robot Interaction during Collaborative Tasks in Smart Factories," *Procedia Comput. Sci.*, vol. 200, pp. 1875–1884, Jan. 2022, doi: 10.1016/J.PROCS.2022.01.388.

[2] H. M. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for Speech Emotion Recognition," *Neural Networks*, vol. 92, pp. 60–68, Aug. 2017, doi: 10.1016/J.NEUNET.2017.02.013.

[3] E. Rejaibi, A. Komaty, F. Meriaudeau, S. Agrebi, and A. Othmani, "MFCC-based Recurrent Neural Network for automatic clinical depression recognition and assessment from speech," *Biomed. Signal Process. Control*, vol. 71, p. 103107, Jan. 2022, doi: 10.1016/J.BSPC.2021.103107.

[4] M. Rayhan Ahmed, S. Islam, A. K. M. Muzahidul Islam, and S. Shatabda, "An ensemble 1D-CNN-LSTM-GRU model with data augmentation for speech emotion recognition," *Expert Syst. Appl.*, vol. 218, p. 119633, May 2023, doi: 10.1016/J.ESWA.2023.119633.

[5] B. W. Schuller, "Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends," *Commun. ACM*, vol. 61, no. 5, pp. 90–99, May 2018, doi: 10.1145/3129340.

[6] "Emotion Recognition In Persian Speech Using Deep Neural Networks | Papers With Code." https://paperswithcode.com/paper/emotion-recognition-in-persian-speech-using (accessed Nov. 18, 2022).

[7] C. S. Kanani, K. S. Gill, S. Behera, A. Choubey, R. K. Gupta, and R. Misra, "Shallow over Deep Neural Networks: A Empirical Analysis for Human Emotion Classification Using Audio Data," pp. 134–146, 2021, doi: 10.1007/978-3-030-76736-5_13.

[8] "(PDF) Speech Emotion Recognition Using CNN." https://www.researchgate.net/publication/342231090_Speech_Emotion_Recognition_Using_CNN (accessed Nov. 18, 2022).

[9] X. Wu *et al.*, "Speech Emotion Recognition algorithm based on deep learning algorithm fusion of temporal and spatial features," *J. Phys. Conf. Ser.*, vol. 1861, no. 1, p. 012064, Mar. 2021, doi: 10.1088/1742-6596/1861/1/012064.

[10] H. Aouani and Y. Ben Ayed, "Speech Emotion Recognition with deep learning," *Procedia Comput. Sci.*, vol. 176, pp. 251–260, Jan. 2020, doi: 10.1016/J.PROCS.2020.08.027.

[11] X. Wu, W.-L. Zheng, and Z. Li, "Recognition of emotions in human speech with deep learning models You may also like Investigating EEG-based functional connectivity patterns for multimodal emotion recognition," *J. Phys. Conf. Ser. Pap. • OPEN ACCESS*, doi: 10.1088/1742-6596/1703/1/012036.

[12] M. M. Hussein *et al.*, “Human speech emotion recognition via feature selection and analyzing You may also like Image Pattern Recognition Algorithm Based on Improved Genetic Algorithm Qing Kuang-An Improved Artificial Neural Network Design for Face Recognition utilizing Harmony S,” *J. Phys. Conf. Ser.*, vol. 1748, p. 42008, 2021, doi: 10.1088/1742-6596/1748/4/042008.

[13] Y. S. Lalitha, A. H. B. Sk, and M. V. A. Nag, “Neural Network Modelling of Speech Emotion Detection,” *E3S Web Conf.*, vol. 309, p. 01139, 2021, doi: 10.1051/E3SCONF/202130901139.

[14] “A proposal for Multimodal Emotion Recognition using aural transformers and Action Units on RAVDESS dataset | Papers With Code.” https://paperswithcode.com/paper/a-proposal-for-multimodal-emotion-recognition?fbclid=IwAR3OUa_hdqUulx6K5kieFVT7JImcpPn4ifTCCNG4lgT_CAE97ak0DDsrSns (accessed Nov. 18, 2022).

[15] S. R. Livingstone and F. A. Russo, “The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS),” Apr. 2018, doi: 10.5281/ZENODO.1188976.

[16] “Toronto emotional speech set (TESS) | TSpace Repository.” https://tspace.library.utoronto.ca/handle/1807/24487 (accessed Jun. 25, 2023).

[17] A. A. Alnuaim *et al.*, “Human-Computer Interaction with Detection of Speaker Emotions Using Convolution Neural Networks,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/7463091.