SPEECH EMOTION RECOGNITION

By

DHEEPAN G

(Registration Number: **22352018**)

Project report submitted in partial fulfilment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF ENGINEERING & TECHNOLOGY

PONDICHERRYUNIVERSITY

MAY 2024

<u>BONAFIDE CERTIFICATE</u>

This is to certify that this project work entitled "**SPEECH EMOTION RECOGNITION**" is a bonafide record of work done by **Mr. DHEEPAN G** (Reg. Number **22352018**) in the partial fulfilment for the Degree of **Master of Computer Applications** in Computer Science in the **Department of Computer Science, School of Engineering and Technology** of Pondicherry University.

This work has not been submitted elsewhere for the award of any other degree to the best of our knowledge.

INTERNAL GUIDE	HEAD OF THE DEPARTMENT
DR.S.L.JAYALAKSHMI	**Dr. S. K. V. JAYAKUMAR**
Assistant Professor	Professor/HOD
Department of Computer Science	Department of Computer Science
School of Engineering & Technology	School of Engineering & Technology
Pondicherry University	Pondicherry University
Pondicherry – 605 014	Pondicherry – 605 014

Submitted for the Viva-Voce Examination held on:

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

The completion of this work is undoubtedly ensured by the support of lot of people and it would be impossible without people who supported and believed in me.

I am very happy to show my immense gratitude and grateful acknowledgement to my beloved guide, **Dr.S.L.JAYALAKSHMI**, Assistant Professor, Department of Computer Science, School of Engineering and Technology, Pondicherry University, Pondicherry, for her excellent guidance, constant encouragement and centering efforts, which helped me to complete my project work.

I express my sincere thanks to **Dr. S.K.V. JAYAKUMAR**, Professor and Head, Department of Computer Science, School of Engineering and Technology, Pondicherry University, Pondicherry, for his kind advises and constant encouragement during the course of this project work.

My gratitude also extends to all the **faculty members** and **non-teaching staffs** of the Department of Computer Science, for all the assistance they rendered so willingly, to help me in completing my project report.

Finally, I am extremely thankful to my parents and friends for giving me the moral support in doing all things.

(DHEEPAN G)

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The Speech Emotion Recognition (SER) project aims to develop an intelligent system capable of recognizing human emotions from speech signals. Emotion recognition from speech plays a crucial role in various applications such as human-computer interaction, sentiment analysis and psychological research.

In this project, we leverage machine learning techniques and signal processing methods to analyse speech signals and extract features that capture the emotional content of the speech. The project involves several key steps, including data collection, preprocessing, feature extraction, model training and evaluation.

Utilizing techniques such as Mel-Frequency Cepstral Coefficients (MFCCs), low-pass filtering, and employing machine learning models like Support Vector Machines (SVM), alongside deep learning architectures such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, facilitates emotion classification. The system's objective is to accurately classify emotions such as happiness, sadness, anger, and neutral states from speech signals.

Moreover, the inclusion of a user-friendly interface enhances accessibility and usability, enabling seamless interaction with the system. Through experimentation and rigorous evaluation, the efficacy of the proposed approach in recognizing emotions from speech is demonstrated.

The SER project holds immense potential to contribute to various domains, including human-computer interaction, mental health assessment, and affective computing, thereby augmenting our comprehension and interaction with human emotions.

1. INTRODUCTION

1.1 ABOUT THE PROJECT

The Speech Emotion Recognition (SER) project endeavors to develop a sophisticated system capable of accurately detecting and categorizing emotions conveyed through human speech captured in audio recordings. With applications spanning diverse domains such as human-computer interaction, customer service and healthcare, emotion recognition holds significant promise. By integrating machine learning algorithms and advanced signal processing techniques, this project meticulously extracts relevant features from speech signals. These features serve as inputs to train robust models, enabling precise classification of emotions with a high degree of accuracy. With the addition of a user-friendly interface, the system becomes accessible to a wider audience, facilitating seamless interaction and utilization across various contexts. Ultimately, by accurately discerning emotional nuances within spoken language, the system aims to provide invaluable insights, enhance user experiences, and foster improved communication dynamics.

1.2 PROJECT PLAN

Our project aims to develop an advanced system for emotion recognition in human speech using machine learning techniques. Similar to human perception of emotions through voice, we are training computers to achieve this capability. Initially, we collect speech recordings and apply noise reduction techniques to ensure clear data. Subsequently, we employ sophisticated algorithms to analyse various aspects of the recordings, such as speech rate and tone, to extract emotional cues. Leveraging this information, our system learns to classify different emotions, including happiness, sadness and neutral. We utilize a range of techniques to train the system effectively. Following training, rigorous testing is conducted to evaluate the system's ability to accurately identify emotions in new recordings. Ultimately, our objective is to enhance the computer's proficiency in understanding emotions in speech, which holds significant potential for applications such as improving the empathetic capabilities of virtual assistants and assisting therapists in better understanding their clients.

1.3 ABOUT THE ORGANISATION

Pondicherry University, also known as PU, is a central research university located in Kalapet, Pondicherry in Union Territory of Puducherry, India. It was established by an Act of Parliament in 1985 by the Department of Higher Education, Ministry of Education, Government of India.

The Vice President of India is the Chancellor along with the Lieutenant Governor of Puducherry acting as the Chief Rector and the President of India is the Visitor of the university. The university is a collegiate university with its jurisdiction spread over the Union Territory of Puducherry located in TamilNadu (Pondicherry and Karaikal), Kerala (Mahe) and Andhra Pradesh (Yanam) and Union Territory of Andaman and Nicobar Islands. The vast jurisdiction over three Union Territories namely gives the university a national character.

2. PROBLEM DEFINITION & FEASIBILITY ANALYSIS

2.1 PROBLEM DEFINITION

The problem we aim to address is the accurate recognition of emotions conveyed through human speech. Emotions play a crucial role in communication and being able to detect them accurately can enhance various applications, including virtual assistants, customer service systems and mental health monitoring tools. However, recognizing emotions solely based on audio data poses significant challenges due to the complexity and variability of human speech.

2.2 EXISTING SYSTEM

Currently, emotion recognition systems primarily rely on manual analysis or basic rule-based approaches, which often lack accuracy and scalability. These systems struggle to capture subtle nuances in speech that convey different emotions. Additionally, they may require extensive human intervention for training and customization, limiting their practicality and efficiency.

2.3 PROPOSED SYSTEM

Our system automates speech emotion recognition using Python and machine learning techniques. It begins by preprocessing audio data to remove noise and extract features like MFCCs. These features are then scaled and used to train SVM, LSTM and CNN classifiers for emotion classification. Evaluation metrics and confusion matrices are generated to assess classifier performance. The system aims to provide accurate emotion detection in real-time, with potential applications in human-computer interaction and mental health monitoring.

MFCC

MFCC is a feature extraction technique widely used in speech and audio processing. MFCCs are used to represent the spectral characteristics of sound in a way that is well-suited for various machine learning tasks, such as speech recognition and music analysis.

In simpler terms, MFCCs are a set of coefficients that capture the shape of the power spectrum of a sound signal.

In our speech emotion recognition system, we employ three different classifiers: Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN). Each classifier utilizes different techniques for emotion classification and has its own strengths and weaknesses.

1. **Support Vector Machine (SVM):**
    - SVM is a supervised learning model that analyzes data for classification and regression analysis.
    - It works by finding the hyperplane that best separates different classes in the feature space.
    - SVM is effective in high-dimensional spaces and is robust against overfitting.
    - However, it may not perform well with large datasets and complex nonlinear relationships between features.
2. **Long Short-Term Memory (LSTM):**
    - LSTM is a type of recurrent neural network (RNN) architecture designed to model sequential data.
    - It is well-suited for analyzing time-series data and has memory cells that can maintain information over time steps.
    - LSTM is effective in capturing long-term dependencies in sequential data, making it suitable for analyzing audio signals.
    - However, training LSTM models can be computationally expensive and requires careful tuning of hyperparameters.

3. **Convolutional Neural Network (CNN):**

    - CNN is a deep learning model commonly used for image recognition tasks.
    - It consists of convolutional layers that extract spatial features from input data.
    - CNN can be adapted for analysing one-dimensional data like audio signals by treating them as spectrograms or time-frequency representations.
    - While CNN is efficient in learning hierarchical representations from data, it may require larger datasets for training and can be sensitive to variations in input data.

GRADIO

Gradio is used as a user interface framework for creating an interactive web-based interface for the Speech Emotion Recognition (SER) system. Gradio simplifies the process of building and deploying machine learning models by providing a high-level interface that allows developers to create user-friendly interfaces with minimal code.

Here's how Gradio is utilized:

1. Interface Creation: Gradio is used to create a user interface that allows users to upload an audio file for emotion recognition.
2. Input Handling: Gradio handles the input of the audio file uploaded by the user.
3. Output Display: Gradio displays the predicted emotion as the output of the SER system.
4. Launching the Interface: Gradio's launch() function is used to start the web-based interface, making it accessible to users via a web browser.

2.4 FEASIBILITY STUDY

2.4.1 TECHNICAL FEASIBILITY

From a technical standpoint, our project is feasible as it leverages well-established machine learning frameworks and libraries such as TensorFlow, Keras and scikit-learn. These tools provide comprehensive support for building, training, and evaluating complex models for emotion recognition. Additionally, the availability of open-source datasets and pre-trained models further enhances the technical feasibility of our project.

2.4.2 OPERATIONAL FEASIBILITY

Operationally, our system can be integrated into various applications and platforms with ease. Once trained, the emotion recognition model can be deployed as a standalone service or integrated into existing systems through APIs. The system's user-friendly interface allows for seamless interaction, making it accessible to both developers and end-users.

2.4.3 ECONOMIC FEASIBILITY

Economically, our project offers significant potential for cost savings and efficiency improvements in various domains. By automating the process of emotion recognition, organizations can reduce the need for manual analysis and intervention, leading to lower operational costs and increased productivity. Additionally, the scalability of our system allows for widespread adoption across different industries, further enhancing its economic feasibility.

3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1 HARDWARE REQUIREMENTS

1. **Processor (CPU)**
    - Dual-core processor or higher.
    - Recommended: Intel Core i5 or Ryzen 5.
2. **Memory (RAM)**
    - Minimum: 4 GB RAM.
    - Recommended: 8 GB RAM or higher for better performance.
3. **Storage Space**
    - At least 3 GB of free disk space for storing datasets, audio files, and project files.
    - Additional space may be required depending on the size of datasets and generated files.

3.2 SOFTWARE REQUIREMENTS

1. **Operating System**

    Windows 10, macOS, or Linux-based operating system.

2. **Python**

    Ensure you have Python installed on your system. You can download and install Python from the official Python website https://www.python.org/downloads/

3. **Text editor**

    Vs code

3.3 SYSTEM REQUIREMENTS

PYTHON LIBRARIES

Os

Used for interacting with the operating system, such as creating directories

pandas

For data manipulation and handling Excel files.

scikit-learn

For machine learning algorithms and evaluation metrics.

NumPy

Essential for numerical computing and handling arrays, used here for processing audio data.

librosa

A library for audio and music analysis. It's used for loading audio files and extracting features like MFCCs.

pydub

For audio processing and manipulation.

SciPy

For signal processing and filtering.

Keras with TensorFlow backend

For building and training deep learning models.

seaborn

For statistical data visualization based on matplotlib.

matplotlib

For creating static, animated, and interactive visualizations in Python.

StandardScaler

Used for scaling features to a standard distribution

Soundfile

A library for reading and writing audio files, for saving temp audio files

Tensorflow

A deep learning framework used for loading and running pre-trained neural network models.

Gradio

Provides an easy-to-use interface for building web-based UIs for machine learning models.

Install these libraries using:

pip install pandas scikit-learn soundfile joblib numpy librosa pydub scipy keras tensorflow seaborn matplotlib gradio

DATASET

(RAVDESS) Ryerson Audio-Visual Database of Emotional Speech and Song is used as dataset for this

INTERNET

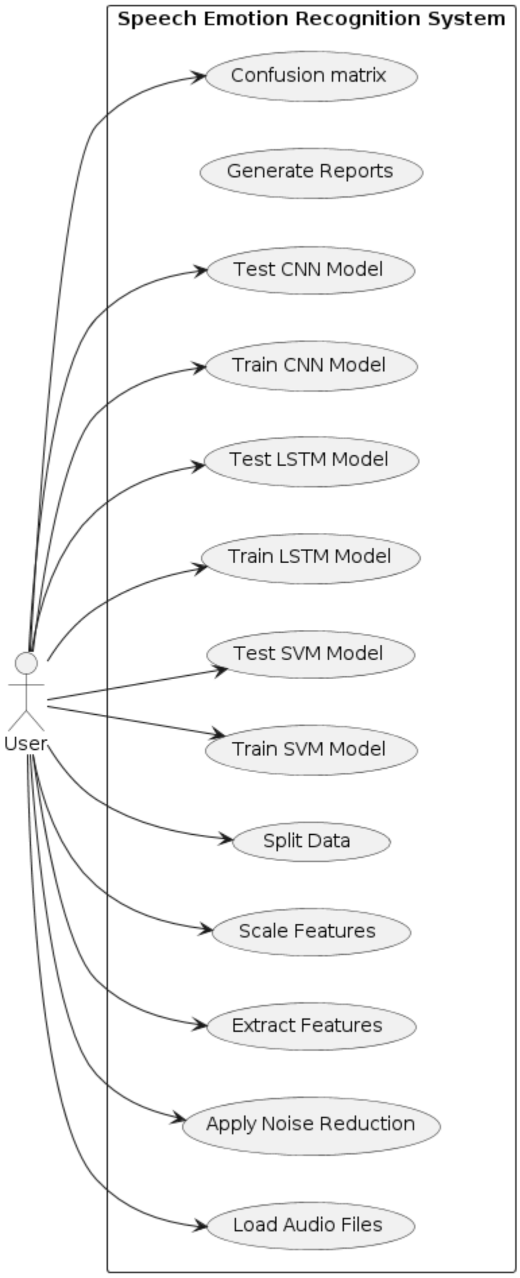
This software will require good internet connection to connect with servers and a good processing system to give best performance

4. SYSTEM DESIGN

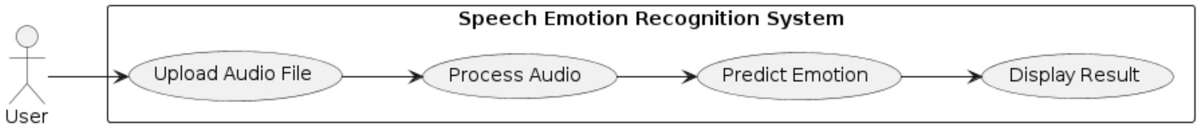
4.1 MODULE DESCRIPTION

1. **Data Loading and Preprocessing:**
    - This module handles the loading of audio data from the dataset folder and performs noise reduction to enhance the quality of audio files.
2. **Feature Extraction:**
    - Responsible for extracting relevant features from the audio files using libraries like librosa and pydub.
3. **Feature Scaling:**
    - Scales the extracted features to ensure uniformity and improve model performance during training.
4. **Data Splitting:**
    - Splits the dataset into training and testing subsets for model evaluation.
5. **Audio Classification:**
    - This module comprises three sub-modules:
        - SVM: Utilizes Support Vector Machine classifier for audio classification.
        - LSTM: Implements Long Short-Term Memory neural network for audio classification.
        - CNN: Employs Convolutional Neural Network for audio classification.
6. **Evaluation and Reporting:**
    - Generates classification reports and confusion matrices to evaluate the performance of each classification model.
7. **Implementation and Prediction:**
    - A small user interface is created to load the audio and predict the emotion.

4.2 USE CASE DIAGRAM



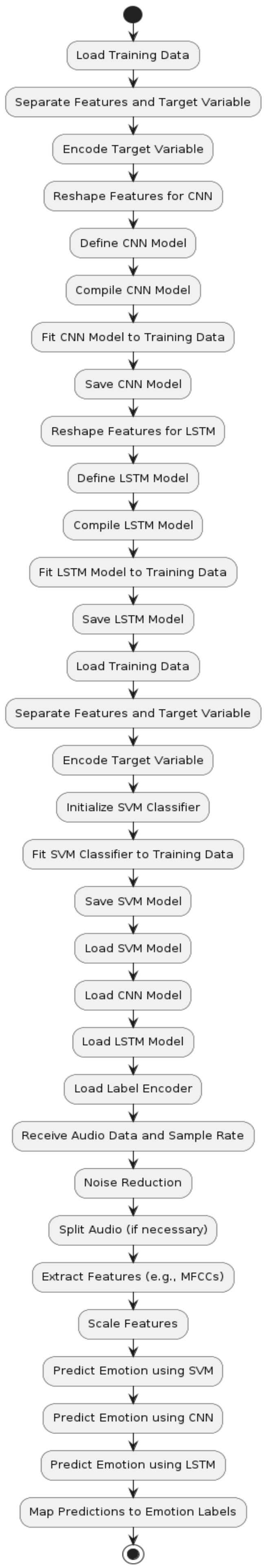
This describes the machine learning modules in my project



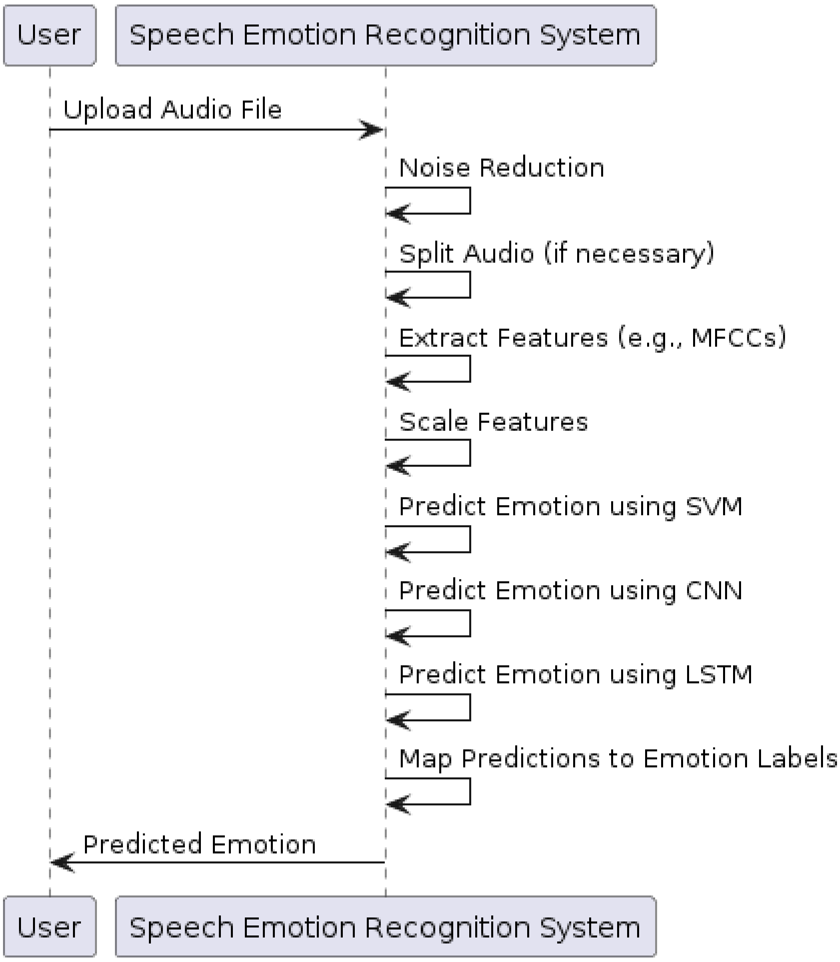
This describes the working of the SER application

4.3 ACTIVITY DIAGRAM

This diagram represents the step-by-step process of training and using machine learning models to recognize emotions in speech. It shows how data is prepared, models are trained and predictions are made to identify emotions in audio recordings.



4.4 SEQUENCE DIAGRAM



This sequence diagram illustrates the interaction between the user and the Speech Emotion Recognition System. It demonstrates how the system processes uploaded audio, predicts emotions using multiple models and returns the predicted emotion to the user.

5. IMPLEMENTATION

5.1 NOISE REDUCTION

This module preprocesses audio data by applying a low-pass filter to reduce background noise, improving the quality of audio recordings.

5.2 FEATURE EXTRACTION

Extracts relevant features from audio signals, such as Mel-Frequency Cepstral Coefficients (MFCCs), which are used as input for emotion classification.

5.3 FEATURE SCALING

Normalizes or scales extracted features to ensure consistency and improve the performance of machine learning models.

5.4 DATA SPLITTING

Splits the dataset into training and testing sets to evaluate model performance.

5.5 CLASSIFIER SELECTION

Chooses appropriate classifiers such as Support Vector Machines (SVM), Convolutional Neural Networks (CNNs), or Long Short-Term Memory (LSTM) networks for emotion classification.

5.6 MODEL TRAINING

Trains the selected classifier using the training data to learn patterns and relationships between input features and emotion labels.

5.7 MODEL EVALUATION

Evaluates the trained model's performance using the testing data, calculating metrics such as accuracy, precision, recall and F1-score.

5.8 TRAINING MODELS

Here, three machine learning models (SVM, CNN, LSTM) are trained to recognize emotions from audio data. The training process involves loading training data, separating features and target variables, encoding the target variable, and fitting the models to the training data. Once trained, these models are saved for later use in the SER system.

An ensemble model, also known as a combination model, is a machine learning technique that combines the predictions from multiple individual models to produce a single prediction. In the context of the SER system, an ensemble model could be constructed by combining the predictions from the SVM, CNN, and LSTM models. This ensemble approach often results in improved prediction accuracy and robustness compared to individual models.

5.9 A SIMPLE USER INTERFACE

Here, Gradio is employed to construct an interactive interface for the Speech Emotion Recognition (SER) system. The `predict_emotion` function processes uploaded audio data, applies noise reduction, feature extraction and machine learning models (SVM, CNN, LSTM) for emotion prediction. Input is an audio uploader and output is a text box displaying the predicted emotion. The `gr.Interface` function integrates these components, defining the title and description, while `launch()` initiates the interface, enabling users to upload audio files and obtain emotion predictions seamlessly through a web browser.

6. SYSTEM TESTING

6.1 SYSTEM IMPLEMENTATION

The system implementation involves developing each module of the project according to the design specifications using appropriate programming languages and libraries. Rigorous testing is performed during the implementation phase to identify and resolve any bugs or errors early on.

6.2 TESTING

6.2.1 UNIT TESTING

Unit testing is conducted for each module individually to ensure that they perform as expected. The correctness of functions and methods within each module is verified, and boundary cases and edge conditions are tested to check for robustness. Testing frameworks like pytest or unittest are used to automate the testing process.

6.2.2 VALIDATION TESTING

Validation testing is carried out to validate the system against the user requirements and specifications. The system's adherence to the intended objectives and functionalities is verified, and its accuracy in detecting and classifying emotions from speech recordings is evaluated. The system's performance in terms of accuracy, speed, and resource utilization is also assessed.

6.2.3 FUNCTIONAL TESTING

Functional testing is performed to validate the end-to-end functionality of the system. Each feature and functionality is tested to ensure they work as intended, and scenario-based testing is conducted to simulate real-world usage scenarios. The integration of different modules and components within the system is verified, and any discrepancies or deviations from the expected behavior are identified and addressed accordingly.

In the context of the Speech Emotion Recognition (SER) project, each module, including audio preprocessing, feature extraction, model training, and prediction, undergoes thorough testing to ensure the system's accuracy and reliability in recognizing emotions from speech recordings. The validation testing ensures that the system meets the specified requirements and delivers the expected functionalities, while functional testing verifies the seamless operation of all system components in various usage scenarios.

7. CONCLUSION

Based on the accuracy results of different models, SVM (Support Vector Machine) has proven to be the most accurate classifier for our speech emotion recognition task, with an accuracy of 0.75, compared to LSTM's 0.25 and CNN's 0.5. This high accuracy indicates that SVM effectively separates different emotional states in the feature space and performs well on our dataset. Given its superior performance, I have developed a robust Speech Emotion Recognition (SER) system that solely leverages the SVM model.

Implemented using the SVM model, this system aims to improve the accuracy and reliability of emotion detection from audio signals. The SVM model's high accuracy ensures that it effectively captures and distinguishes the emotional nuances present in speech.

To conclude, the SVM model has proven to be effective in enhancing the accuracy of speech emotion recognition. The Gradio interface makes the system user-friendly and accessible, allowing users to easily interact with the model and obtain emotion predictions. This project demonstrates the potential of using advanced machine learning techniques for real-world applications in emotion detection from audio signals.

7.1 FUTURE ENHANCEMENT

For future enhancements, the Speech Emotion Recognition (SER) project could explore real-time emotion recognition capabilities, integrating multimodal inputs such as facial expressions with speech analysis. Additionally, incorporating adaptive learning algorithms to personalize emotion detection for individual users would enhance accuracy and user engagement. Emotion generation techniques could be explored to enable the system to respond empathetically. Cross-cultural emotion recognition models could be developed to ensure applicability across diverse populations. Furthermore, integrating privacy-preserving techniques and continuous monitoring mechanisms would bolster user trust and data security. These enhancements aim to elevate the system's effectiveness, applicability and user experience in various domains.

APPENDIX I: SCREENSHOTS

CLEANED AUDIO SAVED IN ACTOR_01

```
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-05-02-01-01-01_angry.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-05-02-01-02-01_angry.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-05-02-02-01-01_angry.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-05-02-02-02-01_angry.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-06-01-01-01-01_fear.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-06-01-01-02-01_fear.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-06-01-02-01-01_fear.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-06-01-02-02-01_fear.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-06-02-01-01-01_fear.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-06-02-01-02-01_fear.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-06-02-02-01-01_fear.wav
Cleaned audio saved to D:/MCA/4th sem/SER3/output/Actor_01\cleaned_03-02-06-02-02-02-01_fear.wav
```

Here the cleaned audio files are stored in the output folder

FEATURE SCALING

```
PS D:\MCA\4th sem\SER3> & "C:/Program Files/Python312/python.exe" "d:/MCA/4th sem/SER3/3_feature_scaling.py"
                                        Audio File    MFCC_1    MFCC_2    MFCC_3    MFCC_4    MFCC_5  ...    MFCC_9   MFCC_10   MFCC_11   MFCC_12   MFCC_13  Emotion
0  D:/MCA/4th sem/SER3/output/Actor_01\cleaned_01... -1.402856  1.196844  1.128203  1.235225  0.276047  ... -0.330737  0.462209  1.852550  1.867828  0.138462  disgust
1  D:/MCA/4th sem/SER3/output/Actor_01\cleaned_01... -0.662524 -0.514808  1.537641  2.051415  2.032518  ...  1.688295  1.630015  1.995256  2.276237  1.156577     fear
2  D:/MCA/4th sem/SER3/output/Actor_01\cleaned_01... -1.313647 -1.633121  0.927738  1.558148  0.322233  ...  1.207835  1.275419  1.399727  1.085798  0.268260  neutral
3  D:/MCA/4th sem/SER3/output/Actor_01\cleaned_01... -1.116778 -0.902882  0.885017  1.333164  0.497132  ...  0.789694  0.712723  1.348356  1.469034  0.475289       ps
4  D:/MCA/4th sem/SER3/output/Actor_01\cleaned_01... -1.923467 -0.926845  0.739330  1.267543  0.862896  ...  1.233841  0.998790  1.116378  1.136657  0.579251      sad

[5 rows x 15 columns]
```

Here after feature scaling is done the scaled output files are stored in the output folder

DATA SPLITTING

```
PS D:\MCA\4th sem\SER3> & "C:/Program Files/Python312/python.exe" "d:/MCA/4th sem/SER3/4_split_data.py"
Data split and saved successfully.
```

The dataset is divided into training and testing phases and stored here

AUDIO CLASSIFICATION SVM

```
PS D:\MCA\4th sem\SER3> & "C:/Program Files/Python312/python.exe" "d:/MCA/4th sem/SER3/5_audio_classification_svm.py"
Classification report saved successfully.
PS D:\MCA\4th sem\SER3>
```

Here audio classification is done using SVM classifier

AUDIO CLASSIFICATION LSTM

```
PS D:\MCA\4th sem\SER3> & "C:/Program Files/Python312/python.exe" "d:/MCA/4th sem/SER3/5_audio_classification_lstm.py"
C:\Users\dheep\AppData\Roaming\Python\Python312\site-packages\keras\src\layers\rnn\rnn.py:204: UserWarning: Do not pass an `input_shape`/
`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model inst
ead.
  super().__init__(**kwargs)
Epoch 1/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.1015 - loss: 2.0730
Epoch 2/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.2263 - loss: 2.0298
Epoch 3/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.2222 - loss: 1.9924
Epoch 4/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.2581 - loss: 1.9623
Epoch 5/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.2971 - loss: 1.9130
Epoch 6/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.2602 - loss: 1.8493
Epoch 7/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.2456 - loss: 1.8092
Epoch 8/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.3226 - loss: 1.7953
Epoch 9/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.3564 - loss: 1.7425
Epoch 10/10
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.3793 - loss: 1.7236
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step
```

Here audio classification is done using LSTM classifier

AUDIO CLASSIFICATION CNN

```
Epoch 2/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.3397 - loss: 1.9800
Epoch 3/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.3067 - loss: 1.9077
Epoch 4/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.3449 - loss: 1.8432
Epoch 5/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.3336 - loss: 1.7794
Epoch 6/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.3622 - loss: 1.7275
Epoch 7/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.4229 - loss: 1.6458
Epoch 8/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.3977 - loss: 1.6398
Epoch 9/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4472 - loss: 1.5678
Epoch 10/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.4810 - loss: 1.5203
Epoch 11/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.4705 - loss: 1.4767
Epoch 12/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.4775 - loss: 1.4344
Epoch 13/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.5399 - loss: 1.4056
Epoch 14/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.5459 - loss: 1.3348
Epoch 15/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5528 - loss: 1.3064
Epoch 16/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5762 - loss: 1.2541
Epoch 17/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.5971 - loss: 1.1866
Epoch 18/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.6248 - loss: 1.1349
Epoch 19/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.6222 - loss: 1.1055
Epoch 20/20
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.6612 - loss: 1.0356
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
```

Here audio classification is done using CNN classifier

CONFUSION MATRIX CNN

```
Confusion Matrix (CNN):
          angry  calm  disgust  fear  happy  neutral  ps  sad
angry         3     0        0     0      0        0   1    0
calm          0     2        0     0      0        0   0    2
disgust       0     0        1     0      1        0   0    0
fear          0     0        0     2      0        0   0    0
happy         0     0        0     1      4        0   0    0
neutral       0     0        0     1      0        0   1    0
ps            0     0        0     1      0        0   1    0
sad           0     1        0     3      1        1   0    1

Accuracy: 0.5
Misclassification Rate: 0.5

Mean Precision: 0.53125
Mean Sensitivity (Recall): 0.5241071428571429
Mean Specificity: 0.9304292929292929
```

Here a confusion matrix is created using CNN classifier

CONFUSION MATRIX LSTM

```
Confusion Matrix (LSTM):
          angry  calm  disgust  fear  happy  neutral  ps  sad
angry         3     0        0     0      0        0   1    0
calm          0     2        0     0      0        0   0    2
disgust       2     0        0     0      0        0   0    0
fear          0     0        0     2      0        0   0    0
happy         2     0        0     3      0        0   0    0
neutral       1     1        0     0      0        0   0    0
ps            1     1        0     0      0        0   0    0
sad           0     3        0     4      0        0   0    0

Accuracy: 0.25
Misclassification Rate: 0.75

Mean Precision: 0.10515873015873016
Mean Sensitivity (Recall): 0.28125
Mean Specificity: 0.8984246138851402
```

Here a confusion matrix is created using LSTM classifier

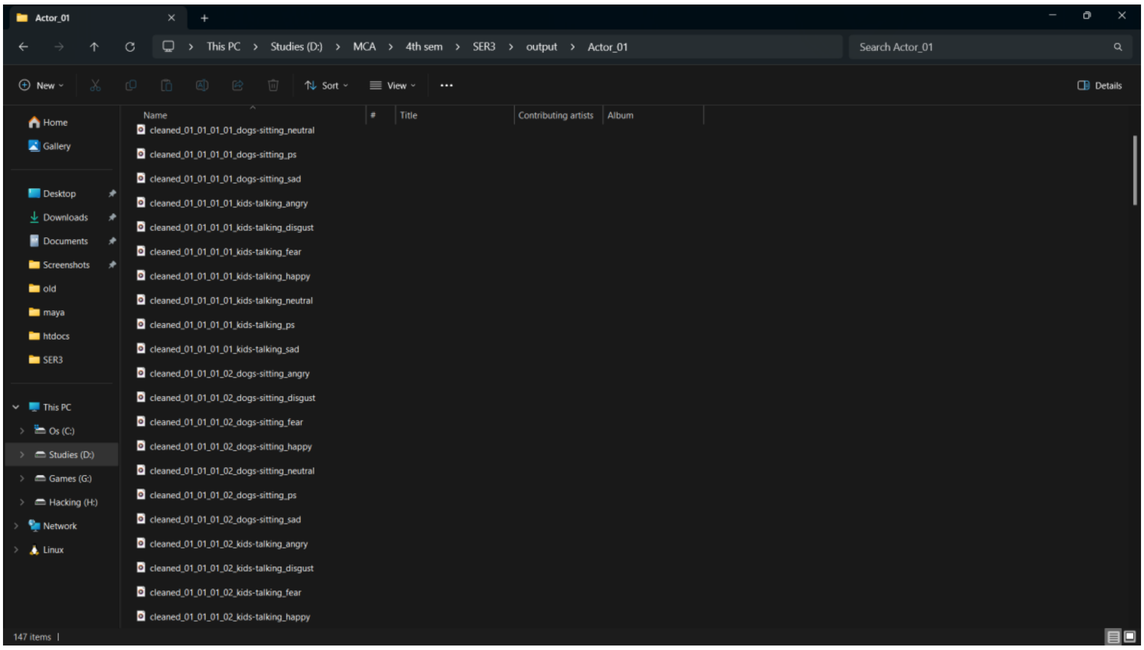
CONFUSION MATRIX SVM

```
Confusion Matrix (SVM):
         angry  calm  disgust  fear  happy  neutral  ps  sad
angry        2     0        0     0      0        0   2    0
calm         0     4        0     0      0        0   0    0
disgust      0     0        2     0      0        0   0    0
fear         0     0        0     1      1        0   0    0
happy        0     1        0     0      4        0   0    0
neutral      0     0        0     0      1        1   0    0
ps           0     0        0     0      0        0   2    0
sad          0     0        0     0      0        2   0    5

Accuracy: 0.75
Misclassification Rate: 0.25

Mean Precision: 0.7875
Mean Sensitivity (Recall): 0.7517857142857143
Mean Specificity: 0.9642036023557762
```

Here a confusion matrix is created using a SVM classifier

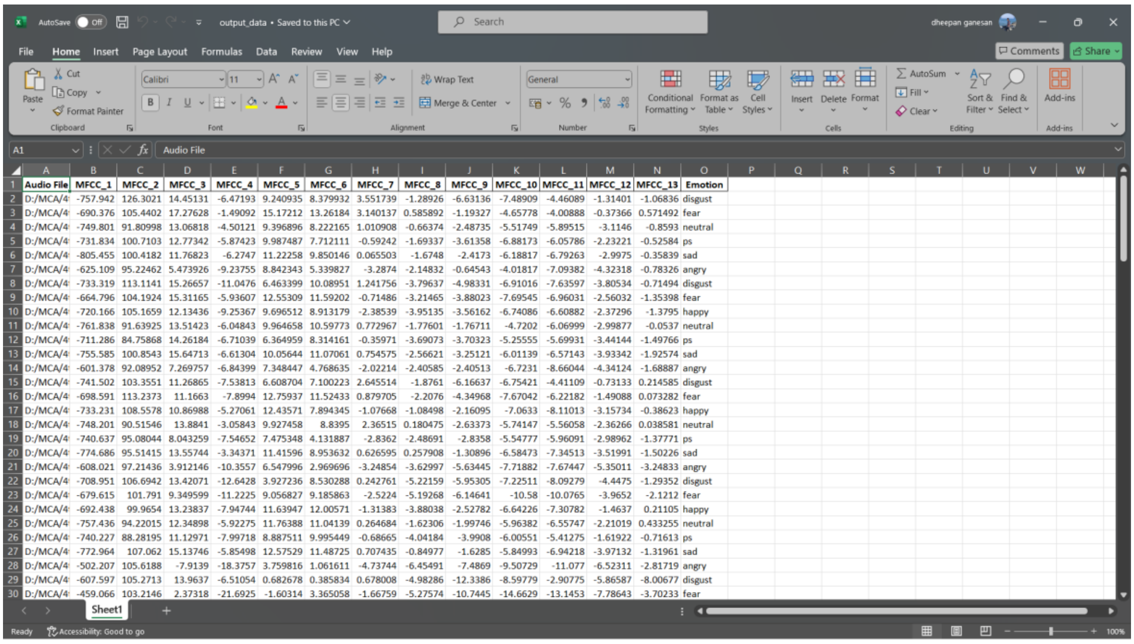
APPENDIX II: OUTPUTS

CLEANED AUDIO IS SAVED IN AUDIO_01 FOLDER



Noise filtering is done and the output is stored as cleaned audio in the audio_01 folder

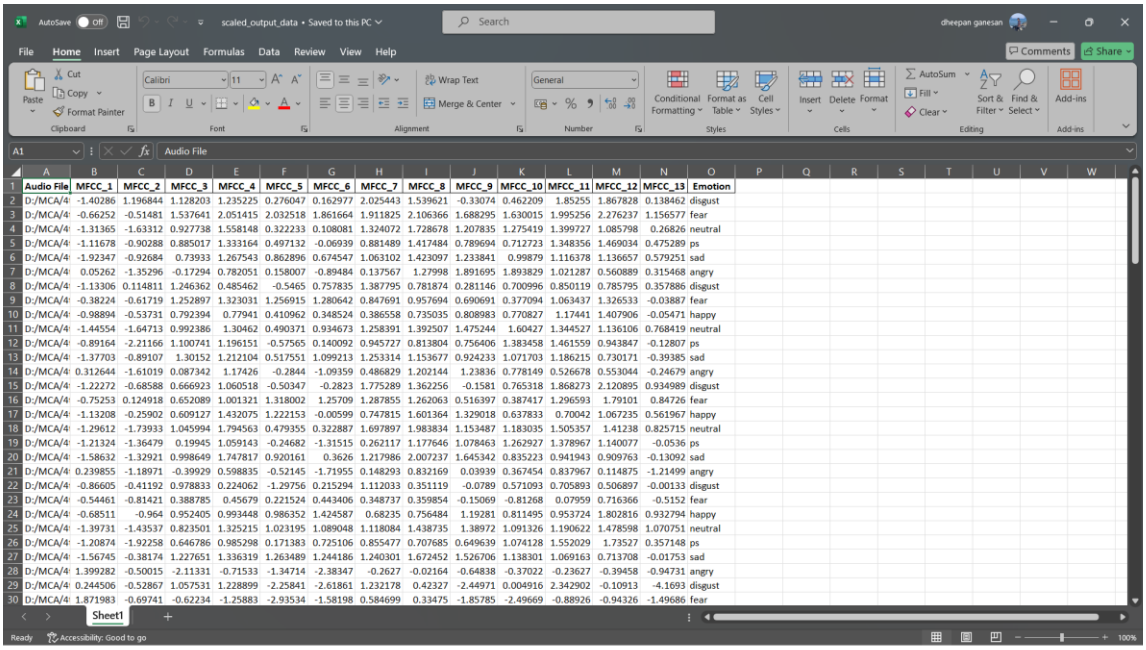
FEATURES EXTRACTED AND SAVED AS OUTPUT_DATA.XLSX



Audio File	MFCC_1	MFCC_2	MFCC_3	MFCC_4	MFCC_5	MFCC_6	MFCC_7	MFCC_8	MFCC_9	MFCC_10	MFCC_11	MFCC_12	MFCC_13	Emotion
D:/MCA/4	-757.942	126.3021	14.45131	-6.47193	9.240935	8.379932	3.551739	-1.28926	-6.63136	-7.48909	-4.46089	-1.31401	-1.06836	disgust
D:/MCA/4	-690.376	105.4402	17.27628	-1.49092	15.17212	13.26184	3.140137	0.585892	-1.19327	-4.65778	-4.00888	-0.37366	0.571492	fear
D:/MCA/4	-749.801	91.80998	13.06818	-4.50121	9.396896	8.222165	1.010908	-0.66374	-2.48735	-5.51749	-5.89515	-3.1146	-0.8593	neutral
D:/MCA/4	-731.834	100.7103	12.77342	-5.87423	9.987487	7.712111	-0.59242	-1.69337	-3.61358	-6.88173	-6.05786	-2.23221	-0.52584	ps
D:/MCA/4	-805.455	100.4182	11.76823	-6.2747	11.22258	9.850146	0.065503	-1.6748	-2.4173	-6.18817	-6.79263	-2.9975	-0.35839	sad
D:/MCA/4	-625.109	95.22462	5.473926	-9.23755	8.842343	5.339827	-3.2874	-2.14832	-0.64543	-4.01817	-7.09382	-4.32318	-0.78326	angry
D:/MCA/4	-733.319	113.1141	15.26657	-11.0476	6.463399	10.08951	1.241756	-3.79637	-4.98331	-6.91016	-7.63597	-3.80534	-0.71494	disgust
D:/MCA/4	-664.796	104.1924	15.31165	-5.93607	12.55309	11.59202	-0.71486	-3.21465	-3.88023	-7.69545	-6.96031	-2.56032	-1.35398	fear
D:/MCA/4	-720.166	105.1659	12.13436	-9.25367	9.696512	8.913179	-2.38539	-3.95135	-3.56162	-6.74086	-6.60882	-2.37296	-1.3795	happy
D:/MCA/4	-761.838	91.63925	13.51423	-6.04843	9.964658	10.59773	0.772967	-1.77601	-1.76711	-4.7202	-6.06999	-2.99877	-0.0537	neutral
D:/MCA/4	-711.286	84.75868	14.26184	-6.71039	6.364959	8.314161	-0.35971	-3.69073	-3.70323	-5.25555	-5.69931	-3.44144	-1.49766	ps
D:/MCA/4	-755.585	100.8543	15.64713	-6.61304	10.05644	11.07061	0.754575	-2.56621	-3.25121	-6.01139	-6.57143	-3.93342	-1.92574	sad
D:/MCA/4	-601.378	92.08952	7.269757	-6.84399	7.348447	4.768635	-2.02214	-2.40585	-2.40513	-6.7231	-8.66044	-4.34124	-1.68887	angry
D:/MCA/4	-741.502	103.3551	11.26865	-7.53813	6.608704	7.100223	2.645514	-1.8761	-6.16637	-6.75421	-4.41109	-0.73133	0.214585	disgust
D:/MCA/4	-698.591	113.2373	11.1663	-7.8994	12.75937	11.52433	0.879705	-2.2076	-4.34968	-7.67042	-6.22182	-1.49088	0.073282	fear
D:/MCA/4	-733.231	108.5578	10.86988	-5.27061	12.43571	7.894345	-1.07668	-1.08498	-2.16095	-7.0633	-8.11013	-3.15734	-0.38623	happy
D:/MCA/4	-748.201	90.51546	13.8841	-3.05843	9.927458	8.8395	2.36515	0.180475	-2.63373	-5.74147	-5.56058	-2.36266	0.038581	neutral
D:/MCA/4	-740.637	95.08044	8.043259	-7.54652	7.475348	4.131887	-2.8362	-2.48691	-2.8358	-5.54777	-5.96091	-2.98962	-1.37771	ps
D:/MCA/4	-774.686	95.51415	13.55744	-3.34371	11.41596	8.953632	0.626595	0.257908	-1.30896	-6.58473	-7.34513	-3.51991	-1.50226	sad
D:/MCA/4	-608.021	97.21436	3.912146	-10.3557	6.547996	2.969696	-3.24854	-3.62997	-5.63445	-7.71882	-7.67447	-5.35011	-3.24833	angry
D:/MCA/4	-708.951	106.6942	13.42071	-12.6428	3.927236	8.530288	0.242761	-5.22159	-5.95305	-7.22511	-8.09279	-4.4475	-1.29352	disgust
D:/MCA/4	-679.615	101.791	9.349599	-11.2225	9.056827	9.185863	-2.5224	-5.19268	-6.14641	-10.58	-10.0765	-3.9652	-2.1212	fear
D:/MCA/4	-692.438	99.9654	13.23837	-7.94744	11.63947	12.00571	-1.31383	-3.88038	-2.52782	-6.64226	-7.30782	-1.4637	0.21105	happy
D:/MCA/4	-757.436	94.22015	12.34898	-5.92275	11.76388	11.04139	0.264684	-1.62306	-1.99746	-5.96382	-6.55747	-2.21019	0.433255	neutral
D:/MCA/4	-740.227	88.28195	11.12971	-7.99718	8.887511	9.995449	-0.68665	-4.04184	-3.9908	-6.00551	-5.41275	-1.61922	-0.71613	ps
D:/MCA/4	-772.964	107.062	15.13746	-5.85498	12.57529	11.48725	0.707435	-0.84977	-1.6285	-5.84993	-6.94218	-3.97132	-1.31961	sad
D:/MCA/4	-502.207	105.6188	-7.9139	-18.3757	3.759816	1.061611	-4.73744	-6.45491	-7.4869	-9.50729	-11.077	-6.52311	-2.81719	angry
D:/MCA/4	-607.597	105.2713	13.9637	-6.51054	0.682678	0.385834	0.678008	-4.98286	-12.3386	-8.59779	-2.90775	-5.86587	-8.00677	disgust
D:/MCA/4	-459.066	103.2146	2.37318	-21.6925	-1.60314	3.365058	-1.66759	-5.27574	-10.7445	-14.6629	-13.1453	-7.78643	-3.70233	fear

The extracted features are stored in the output_data.xlsx file

FEATURE SCALING DONE AND SAVED AS SCALED_OUTPUT_DATA.XLSX

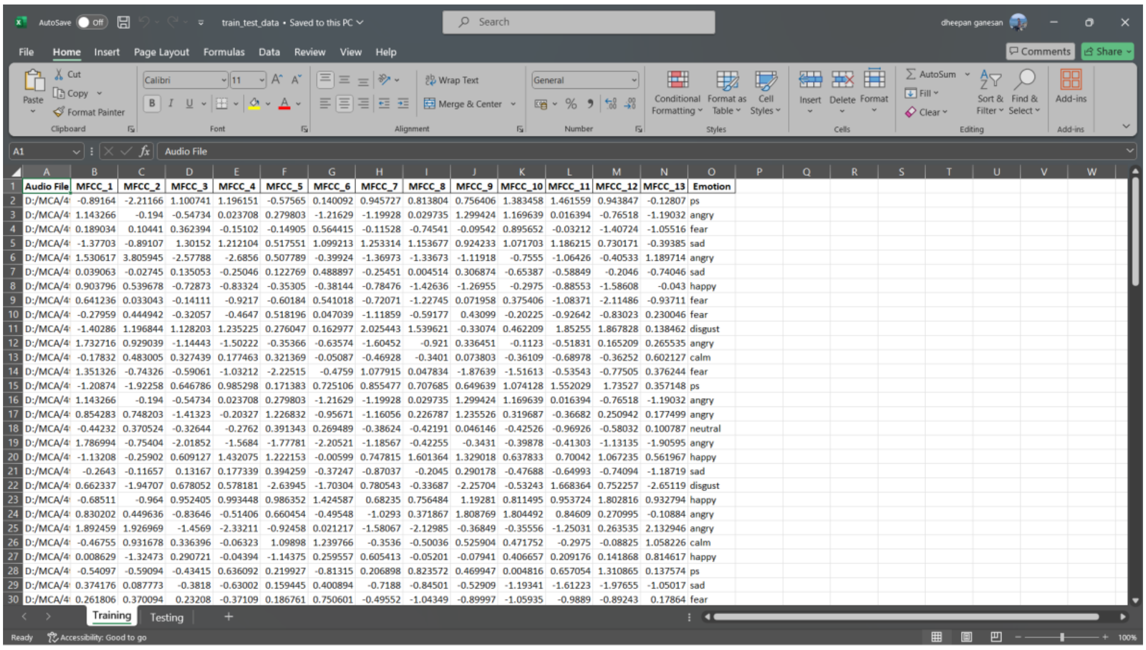


Audio File	MFCC_1	MFCC_2	MFCC_3	MFCC_4	MFCC_5	MFCC_6	MFCC_7	MFCC_8	MFCC_9	MFCC_10	MFCC_11	MFCC_12	MFCC_13	Emotion
D:/MCA/4	-1.40286	1.196844	1.128203	1.235225	0.276047	0.162977	2.025443	1.539621	-0.33074	0.462209	1.85255	1.867828	0.138462	disgust
D:/MCA/4	-0.66252	-0.51481	1.537641	2.051415	2.032518	1.861664	1.911825	2.106366	1.688295	1.630015	1.995256	2.276237	1.156577	fear
D:/MCA/4	-1.31365	-1.63312	0.927738	1.558148	0.322233	0.108081	1.324072	1.728678	1.207835	1.275419	1.399727	1.085798	0.26826	neutral
D:/MCA/4	-1.11678	-0.90288	0.885017	1.333164	0.497132	-0.06939	0.881489	1.417484	0.789694	0.712723	1.348356	1.469034	0.475289	ps
D:/MCA/4	-1.92347	-0.92684	0.73933	1.267543	0.862896	0.674547	1.063102	1.423097	1.233841	0.99879	1.116378	1.136657	0.579251	sad
D:/MCA/4	0.05262	-1.35296	-0.17294	0.782051	0.158007	-0.89484	0.137567	1.27998	1.891695	1.893829	1.021287	0.560889	0.315468	angry
D:/MCA/4	-1.13306	0.114811	1.246362	0.485462	-0.5465	0.757835	1.387795	0.781874	0.281146	0.700996	0.850119	0.785795	0.357886	disgust
D:/MCA/4	-0.38224	-0.61719	1.252897	1.323031	1.256915	1.280642	0.847691	0.957694	0.690691	0.377094	1.063437	1.326533	-0.03887	fear
D:/MCA/4	-0.98894	-0.53731	0.792394	0.77941	0.410962	0.348524	0.386558	0.735035	0.808983	0.770827	1.17441	1.407906	-0.05471	happy
D:/MCA/4	-1.44554	-1.64713	0.992386	1.30462	0.490371	0.934673	1.258391	1.392507	1.475244	1.60427	1.344527	1.136106	0.768419	neutral
D:/MCA/4	-0.89164	-2.21166	1.100741	1.196151	-0.57565	0.140092	0.945727	0.813804	0.756406	1.383458	1.461559	0.943847	-0.12807	ps
D:/MCA/4	-1.37703	-0.89107	1.30152	1.212104	0.517551	1.099213	1.253314	1.153677	0.924233	1.071703	1.186215	0.730171	-0.39385	sad
D:/MCA/4	0.312644	-1.61019	0.087342	1.17426	-0.2844	-1.09359	0.486829	1.202144	1.23836	0.778149	0.526678	0.553044	-0.24679	angry
D:/MCA/4	-1.22272	-0.68588	0.666923	1.060518	-0.50347	-0.2823	1.775289	1.362256	-0.1581	0.765318	1.868273	2.120895	0.934989	disgust
D:/MCA/4	-0.75253	0.124918	0.652089	1.001321	1.318002	1.25709	1.287855	1.262063	0.516397	0.387417	1.296593	1.79101	0.84726	fear
D:/MCA/4	-1.13208	-0.25902	0.609127	1.432075	1.222153	-0.00599	0.747815	1.601364	1.329018	0.637833	0.70042	1.067235	0.561967	happy
D:/MCA/4	-1.29612	-1.73933	1.045994	1.794563	0.479355	0.322887	1.697897	1.983834	1.153487	1.183035	1.505357	1.41238	0.825715	neutral
D:/MCA/4	-1.21324	-1.36479	0.19945	1.059143	-0.24682	-1.31515	0.262117	1.177646	1.078463	1.262927	1.378967	1.140077	-0.0536	ps
D:/MCA/4	-1.58632	-1.32921	0.998649	1.747817	0.920161	0.3626	1.217986	2.007237	1.645342	0.835223	0.941943	0.909763	-0.13092	sad
D:/MCA/4	0.239855	-1.18971	-0.39929	0.598835	-0.52145	-1.71955	0.148293	0.832169	0.03939	0.367454	0.837967	0.114875	-1.21499	angry
D:/MCA/4	-0.86605	-0.41192	0.978833	0.224062	-1.29756	0.215294	1.112033	0.351119	-0.0789	0.571093	0.705893	0.506897	-0.00133	disgust
D:/MCA/4	-0.54461	-0.81421	0.388785	0.45679	0.221524	0.443406	0.348737	0.359854	-0.15069	-0.81268	0.07959	0.716366	-0.5152	fear
D:/MCA/4	-0.68511	-0.964	0.952405	0.993448	0.986352	1.424587	0.68235	0.756484	1.19281	0.811495	0.953724	1.802816	0.932794	happy
D:/MCA/4	-1.39731	-1.43537	0.823501	1.325215	1.023195	1.089048	1.118084	1.438735	1.38972	1.091326	1.190622	1.478598	1.070751	neutral
D:/MCA/4	-1.20874	-1.92258	0.646786	0.985298	0.171383	0.725106	0.855477	0.707685	0.649639	1.074128	1.552029	1.73527	0.357148	ps
D:/MCA/4	-1.56745	-0.38174	1.227651	1.336319	1.263489	1.244186	1.240301	1.672452	1.526706	1.138301	1.069163	0.713708	-0.01753	sad
D:/MCA/4	1.399282	-0.50015	-2.11331	-0.71533	-1.34714	-2.38347	-0.2627	-0.02164	-0.64838	-0.37022	-0.23627	-0.39458	-0.94731	angry
D:/MCA/4	0.244506	-0.52867	1.057531	1.228899	-2.25841	-2.61861	1.232178	0.42327	-2.44971	0.004916	2.342902	-0.10913	-4.1693	disgust
D:/MCA/4	1.871983	-0.69741	-0.62234	-1.25883	-2.93534	-1.58198	0.584699	0.33475	-1.85785	-2.49669	-0.88926	-0.94326	-1.49686	fear

The scaled features are stored in the scaled_output.xlsx file

DATA SPLITTING

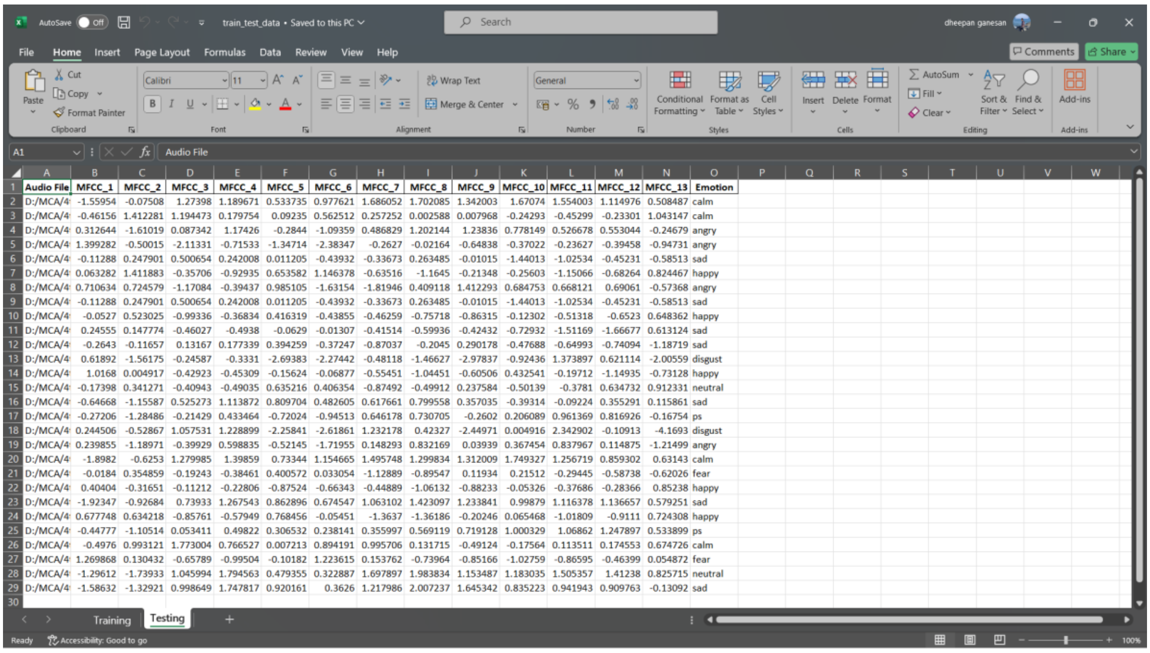
TRAINING DATASET



Audio File	MFCC_1	MFCC_2	MFCC_3	MFCC_4	MFCC_5	MFCC_6	MFCC_7	MFCC_8	MFCC_9	MFCC_10	MFCC_11	MFCC_12	MFCC_13	Emotion
D:/MCA/4	-0.89164	-2.21166	1.100741	1.196151	-0.57565	0.140092	0.945727	0.813804	0.756406	1.383458	1.461559	0.943847	-0.12807	ps
D:/MCA/4	1.143266	-0.194	-0.54734	0.023708	0.279803	-1.21629	-1.19928	0.029735	1.299424	1.169639	0.016394	-0.76518	-1.19032	angry
D:/MCA/4	0.189034	0.10441	0.362394	-0.15102	-0.14905	0.564415	-0.11528	-0.74541	-0.09542	0.895652	-0.03212	-1.40724	-1.05516	fear
D:/MCA/4	-1.37703	-0.89107	1.30152	1.212104	0.517551	1.099213	1.253314	1.153677	0.924233	1.071703	1.186215	0.730171	-0.39385	sad
D:/MCA/4	1.530617	3.805945	-2.57788	-2.6856	0.507789	-0.39924	-1.36973	-1.33673	-1.11918	-0.7555	-1.06426	-0.40533	1.189714	angry
D:/MCA/4	0.039063	-0.02745	0.135053	-0.25046	0.122769	0.488897	-0.25451	0.004514	0.306874	-0.65387	-0.58849	-0.2046	-0.74046	sad
D:/MCA/4	0.903796	0.539678	-0.72873	-0.83324	-0.35305	-0.38144	-0.78476	-1.42636	-1.26955	-0.2975	-0.88553	-1.58608	-0.043	happy
D:/MCA/4	0.641236	0.033043	-0.14111	-0.9217	-0.60184	0.541018	-0.72071	-1.22745	0.071958	0.375406	-1.08371	-2.11486	-0.93711	fear
D:/MCA/4	-0.27959	0.444942	-0.32057	-0.4647	0.518196	0.047039	-1.11859	-0.59177	0.43099	-0.20225	-0.92642	-0.83023	0.230046	fear
D:/MCA/4	-1.40286	1.196844	1.128203	1.235225	0.276047	0.162977	2.025443	1.539621	-0.33074	0.462209	1.85255	1.867828	0.138462	disgust
D:/MCA/4	1.732716	0.929039	-1.14443	-1.50222	-0.35366	-0.63574	-1.60452	-0.921	0.336451	-0.1123	-0.51831	0.165209	0.265535	angry
D:/MCA/4	-0.17832	0.483005	0.327439	0.177463	0.321369	-0.05087	-0.46928	-0.3401	0.073803	-0.36109	-0.68978	-0.36252	0.602127	calm
D:/MCA/4	1.351326	-0.74326	-0.59061	-1.03212	-2.22515	-0.4759	1.077915	0.047834	-1.87639	-1.51613	-0.53543	-0.77505	0.376244	fear
D:/MCA/4	-1.20874	-1.92258	0.646786	0.985298	0.171383	0.725106	0.855477	0.707685	0.649639	1.074128	1.552029	1.73527	0.357148	ps
D:/MCA/4	1.143266	-0.194	-0.54734	0.023708	0.279803	-1.21629	-1.19928	0.029735	1.299424	1.169639	0.016394	-0.76518	-1.19032	angry
D:/MCA/4	0.854283	0.748203	-1.41323	-0.20327	1.226832	-0.95671	-1.16056	0.226787	1.235526	0.319687	-0.36682	0.250942	0.177499	angry
D:/MCA/4	-0.44232	0.370524	-0.32644	-0.2762	0.391343	0.269489	-0.38624	-0.42191	0.046146	-0.42526	-0.96926	-0.58032	0.100787	neutral
D:/MCA/4	1.786994	-0.75404	-2.01852	-1.5684	-1.77781	-2.20521	-1.18567	-0.42255	-0.3431	-0.39878	-0.41303	-1.13135	-1.90595	angry
D:/MCA/4	-1.13208	-0.25902	0.609127	1.432075	1.222153	-0.00599	0.747815	1.601364	1.329018	0.637833	0.70042	1.067235	0.561967	happy
D:/MCA/4	-0.2643	-0.11657	0.13167	0.177339	0.394259	-0.37247	-0.87037	-0.2045	0.290178	-0.47688	-0.64993	-0.74094	-1.18719	sad
D:/MCA/4	0.662337	-1.94707	0.678052	0.578181	-2.63945	-1.70304	0.780543	-0.33687	-2.25704	-0.53243	1.668364	0.752257	-2.65119	disgust
D:/MCA/4	-0.68511	-0.964	0.952405	0.993448	0.986352	1.424587	0.68235	0.756484	1.19281	0.811495	0.953724	1.802816	0.932794	happy
D:/MCA/4	0.830202	0.449636	-0.83646	-0.51406	0.660454	-0.49548	-1.0293	0.371867	1.808769	1.804492	0.84609	0.270995	-0.10884	angry
D:/MCA/4	1.892459	1.926969	-1.4569	-2.33211	-0.92458	0.021217	-1.58067	-2.12985	-0.36849	-0.35556	-1.25031	0.263535	2.132946	angry
D:/MCA/4	-0.46755	0.931678	0.336396	-0.06323	1.09898	1.239766	-0.3536	-0.50036	0.525904	0.471752	-0.2975	-0.08825	1.058226	calm
D:/MCA/4	0.008629	-1.32473	0.290721	-0.04394	-1.14375	0.259557	0.605413	-0.05201	-0.07941	0.406657	0.209176	0.141868	0.814617	happy
D:/MCA/4	-0.54097	-0.59094	-0.43415	0.636092	0.219927	-0.81315	0.206898	0.823572	0.469947	0.004816	0.657054	1.310865	0.137574	ps
D:/MCA/4	0.374176	0.087773	-0.3818	-0.63002	0.159445	0.400894	-0.7188	-0.84501	-0.52909	-1.19341	-1.61223	-1.97655	-1.05017	sad
D:/MCA/4	0.261806	0.370094	0.23208	-0.37109	0.186761	0.750601	-0.49552	-1.04349	-0.89997	-1.05935	-0.9889	-0.89243	0.17864	fear

In data splitting the scaled data is splitted into training phase and stored as training sheet in train_test_data.xlsx file

TESTING DATASET



Audio File	MFCC_1	MFCC_2	MFCC_3	MFCC_4	MFCC_5	MFCC_6	MFCC_7	MFCC_8	MFCC_9	MFCC_10	MFCC_11	MFCC_12	MFCC_13	Emotion
D:/MCA/4	-1.55954	-0.07508	1.27398	1.189671	0.533735	0.977621	1.686052	1.702085	1.342003	1.67074	1.554003	1.114976	0.508487	calm
D:/MCA/4	-0.46156	1.412281	1.194473	0.179754	0.09235	0.562512	0.257252	0.002588	0.007968	-0.24293	-0.45299	-0.23301	1.043147	calm
D:/MCA/4	0.312644	-1.61019	0.087342	1.17426	-0.2844	-1.09359	0.486829	1.202144	1.23836	0.778149	0.526678	0.553044	-0.24679	angry
D:/MCA/4	1.399282	-0.50015	-2.11331	-0.71533	-1.34714	-2.38347	-0.2627	-0.02164	-0.64838	-0.37022	-0.23627	-0.39458	-0.94731	angry
D:/MCA/4	-0.11288	0.247901	0.500654	0.242008	0.011205	-0.43932	-0.33673	0.263485	-0.01015	-1.44013	-1.02534	-0.45231	-0.58513	sad
D:/MCA/4	0.063282	1.411883	-0.35706	-0.92935	0.653582	1.146378	-0.63516	-1.1645	-0.21348	-0.25603	-1.15066	-0.68264	0.824467	happy
D:/MCA/4	0.710634	0.724579	-1.17084	-0.39437	0.985105	-1.63154	-1.81946	0.409118	1.412293	0.684753	0.668121	0.69061	-0.57368	angry
D:/MCA/4	-0.11288	0.247901	0.500654	0.242008	0.011205	-0.43932	-0.33673	0.263485	-0.01015	-1.44013	-1.02534	-0.45231	-0.58513	sad
D:/MCA/4	-0.0527	0.523025	-0.99336	-0.36834	0.416319	-0.43855	-0.46259	-0.75718	-0.86315	-0.12302	-0.51318	-0.6523	0.648362	happy
D:/MCA/4	0.24555	0.147774	-0.46027	-0.4938	-0.0629	-0.01307	-0.41514	-0.59936	-0.42432	-0.72932	-1.51169	-1.66677	0.613124	sad
D:/MCA/4	-0.2643	-0.11657	0.13167	0.177339	0.394259	-0.37247	-0.87037	-0.2045	0.290178	-0.47688	-0.64993	-0.74094	-1.18719	sad
D:/MCA/4	0.61892	-1.56175	-0.24587	-0.3331	-2.69383	-2.27442	-0.48118	-1.46627	-2.97837	-0.92436	1.373897	0.621114	-2.00559	disgust
D:/MCA/4	1.0168	0.004917	-0.42923	-0.45309	-0.15624	-0.06877	-0.55451	-1.04451	-0.60506	0.432541	-0.19712	-1.14935	-0.73128	happy
D:/MCA/4	-0.17398	0.341271	-0.40943	-0.49035	0.635216	0.406354	-0.87492	-0.49912	0.237584	-0.50139	-0.3781	0.634732	0.912331	neutral
D:/MCA/4	-0.64668	-1.15587	0.525273	1.113872	0.809704	0.482605	0.617661	0.799558	0.357035	-0.39314	-0.09224	0.355291	0.115861	sad
D:/MCA/4	-0.27206	-1.28486	-0.21429	0.433464	-0.72024	-0.94513	0.646178	0.730705	-0.2602	0.206089	0.961369	0.816926	-0.16754	ps
D:/MCA/4	0.244506	-0.52867	1.057531	1.228899	-2.25841	-2.61861	1.232178	0.42327	-2.44971	0.004916	2.342902	-0.10913	-4.1693	disgust
D:/MCA/4	0.239855	-1.18971	-0.39929	0.598835	-0.52145	-1.71955	0.148293	0.832169	0.03939	0.367454	0.837967	0.114875	-1.21499	angry
D:/MCA/4	-1.8982	-0.6253	1.279985	1.39859	0.73344	1.154665	1.495748	1.299834	1.312009	1.749327	1.256719	0.859302	0.63143	calm
D:/MCA/4	-0.0184	0.354859	-0.19243	-0.38461	0.400572	0.033054	-1.12889	-0.89547	0.11934	0.21512	-0.29445	-0.58738	-0.62026	fear
D:/MCA/4	0.40404	-0.31651	-0.11212	-0.22806	-0.87524	-0.66343	-0.44889	-1.06132	-0.88233	-0.05326	-0.37686	-0.28366	0.85238	happy
D:/MCA/4	-1.92347	-0.92684	0.73933	1.267543	0.862896	0.674547	1.063102	1.423097	1.233841	0.99879	1.116378	1.136657	0.579251	sad
D:/MCA/4	0.677748	0.634218	-0.85761	-0.57949	0.768456	-0.05451	-1.3637	-1.36186	-0.20246	0.065468	-1.01809	-0.9111	0.724308	happy
D:/MCA/4	-0.44777	-1.10514	0.053411	0.49822	0.306532	0.238141	0.355997	0.569119	0.719128	1.000329	1.06862	1.247897	0.533899	ps
D:/MCA/4	-0.4976	0.993121	1.773004	0.766527	0.007213	0.894191	0.995706	0.131715	-0.49124	-0.17564	0.113511	0.174553	0.674726	calm
D:/MCA/4	1.269868	0.130432	-0.65789	-0.99504	-0.10182	1.223615	0.153762	-0.73964	-0.85166	-1.02759	-0.86595	-0.46399	0.054872	fear
D:/MCA/4	-1.29612	-1.73933	1.045994	1.794563	0.479355	0.322887	1.697897	1.983834	1.153487	1.183035	1.505357	1.41238	0.825715	neutral
D:/MCA/4	-1.58632	-1.32921	0.998649	1.747817	0.920161	0.3626	1.217986	2.007237	1.645342	0.835223	0.941943	0.909763	-0.13092	sad

In data splitting the scaled data is splitted into testing phase and stored as testing sheet in train_test_data.xlsx file

CLASSIFY REPORT CNN

	precision	recall	f1-score	support
angry	1	0.5	0.666667	4
calm	0.4	0.5	0.444444	4
disgust	1	0.5	0.666667	2
fear	0.285714	1	0.444444	2
happy	0.5	0.8	0.615385	5
neutral	0	0	0	2
ps	0.5	0.5	0.5	2
sad	0.333333	0.142857	0.2	7
accuracy	0.464286	0.464286	0.464286	0.464286
macro avg	0.502381	0.492857	0.442201	28
weighted avg	0.50017	0.464286	0.4337	28

This is the performance analysis report of the CNN classifier

CLASSIFY REPORT LSTM

	precision	recall	f1-score	support
angry	0.5	0.25	0.333333	4
calm	0.5	1	0.666667	4
disgust	0	0	0	2
fear	0.111111	0.5	0.181818	2
happy	0.285714	0.4	0.333333	5
neutral	0	0	0	2
ps	0.5	0.5	0.5	2
sad	0	0	0	7
accuracy	0.321429	0.321429	0.321429	0.321429
macro avg	0.237103	0.33125	0.251894	28
weighted avg	0.237528	0.321429	0.251082	28

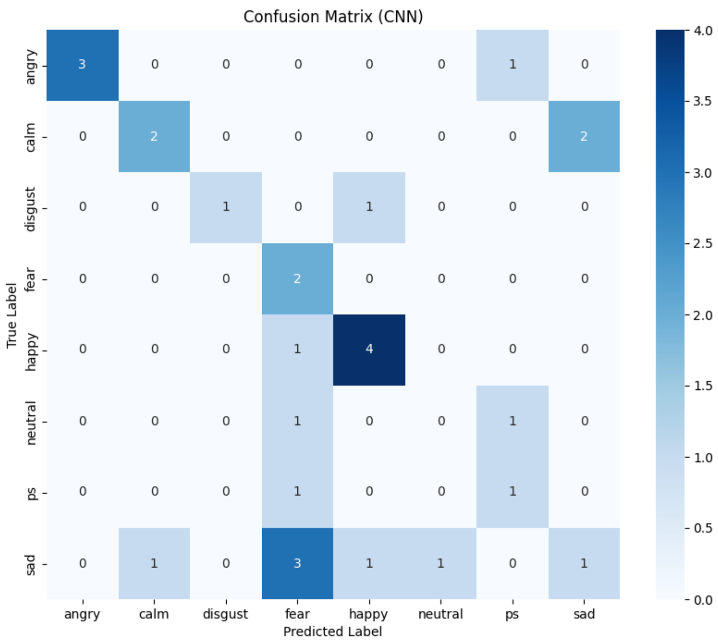
This is the performance analysis report of the LSTM classifier

CLASSIFY REPORT SVM

	precision	recall	f1-score	support
angry	1	0.5	0.666667	4
calm	0.8	1	0.888889	4
disgust	1	1	1	2
fear	1	0.5	0.666667	2
happy	0.666667	0.8	0.727273	5
neutral	0.333333	0.5	0.4	2
ps	0.5	1	0.666667	2
sad	1	0.714286	0.833333	7
accuracy	0.75	0.75	0.75	0.75
macro avg	0.7875	0.751786	0.731187	28
weighted avg	0.828571	0.75	0.755664	28

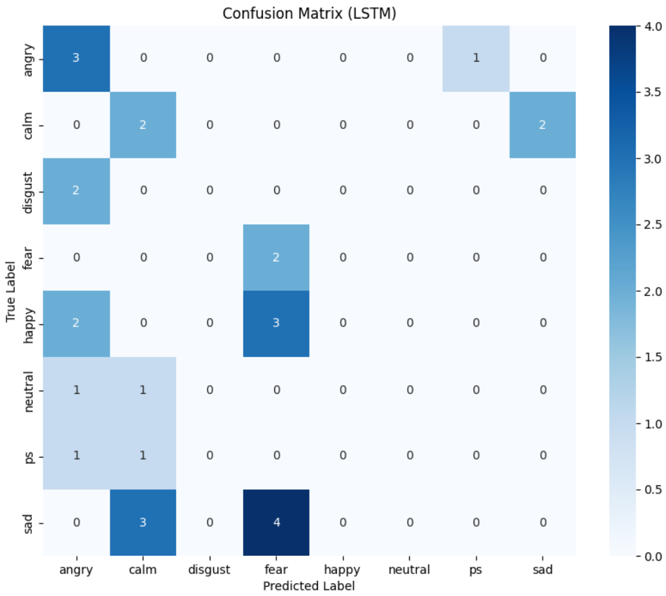
This is the performance analysis report of the SVM classifier

CONFUSION MATRIX CNN



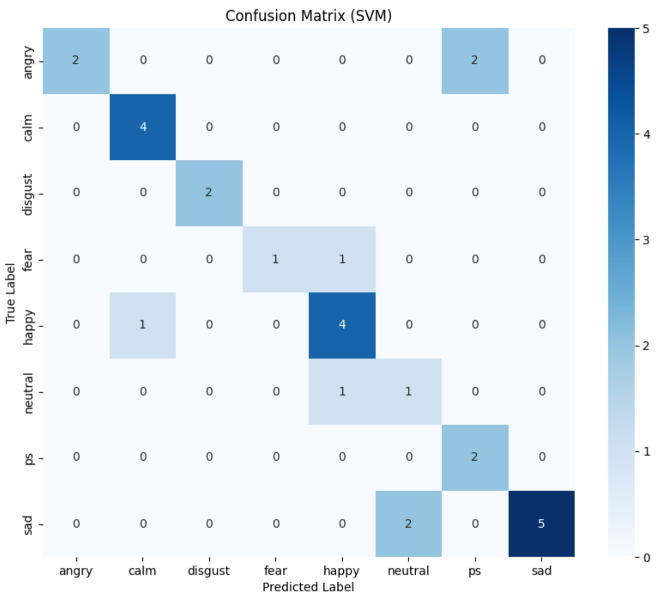
This is the confusion matrix generated using CNN classifier

CONFUSION MATRIX LSTM



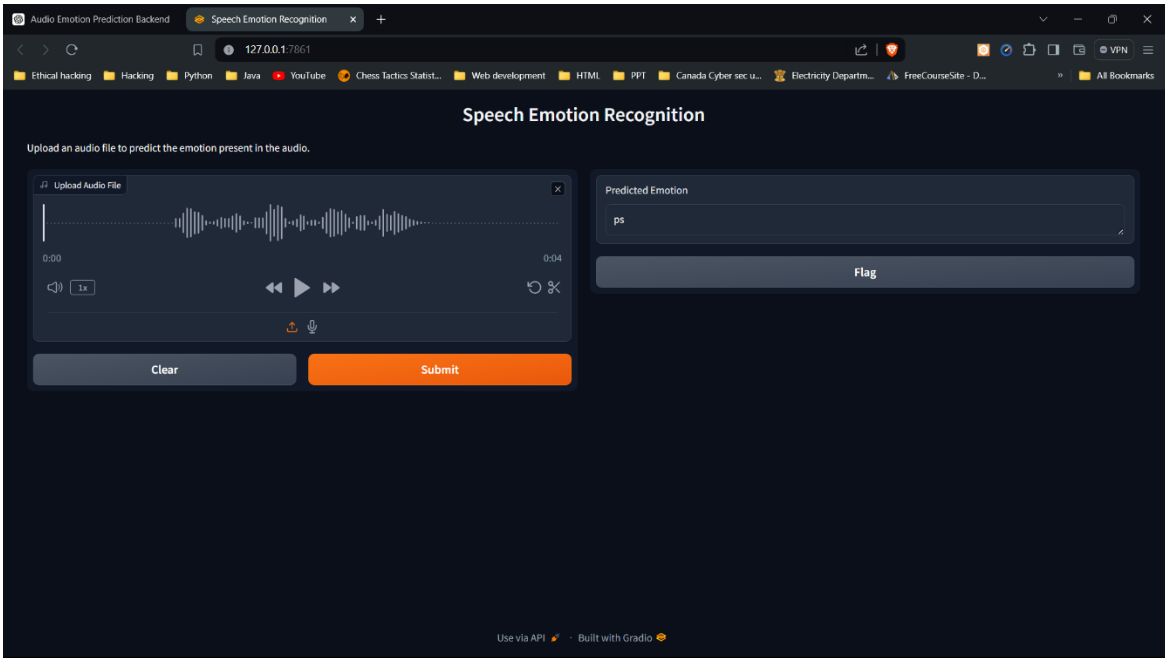
This is the confusion matrix generated using LSTM classifier

CONFUSION MATRIX SVM



This is the confusion matrix generated using SVM classifier

SER UI



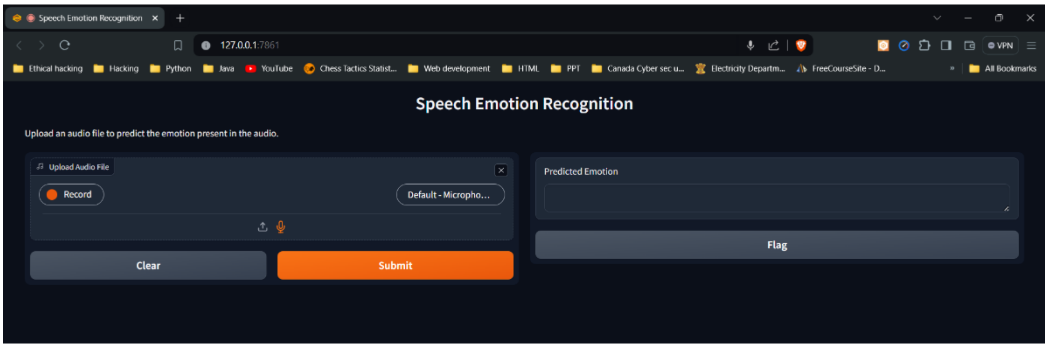
A simple application to load a audio file and predict the emotion of the speaker using SVM classifier.

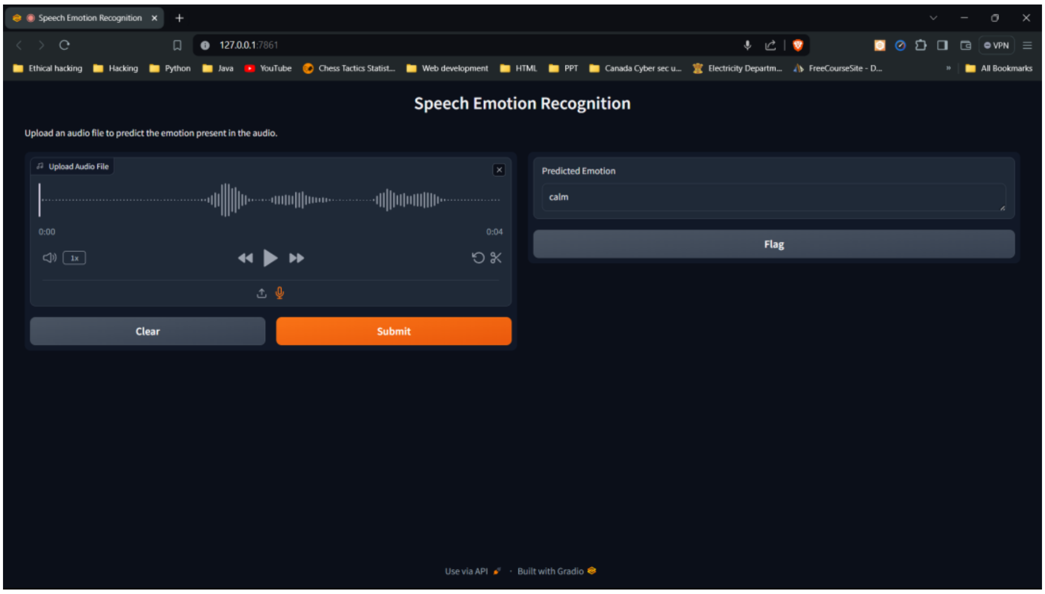
OUTPUT TERMINAL

```
PS D:\MCA\4th sem\SER3> & "C:/Program Files/Python312/python.exe" "d:/MCA/4th sem/SER3/ser_ui_1.py"
Running on local URL:  http://127.0.0.1:7861

To create a public link, set `share=True` in `launch()`.
Received audio data and sample rate
Audio tuple: (48000, array([0, 0, 0, ..., 0, 0, 0], dtype=int16))
Audio saved successfully
Audio loaded successfully
Noise reduction step
Audio splitting step
Loading models
Feature extraction step
Feature scaling step
Predicting emotion using SVM
```

These things happen in the background when a audio file is loaded and asked to predict

SER UI





A simple application to record a audio file and predict the emotion of the speaker using SVM classifier.

OUTPUT TERMINAL

```
PS D:\MCA\4th sem\SER3> & "C:/Program Files/Python312/python.exe" "d:/MCA/4th sem/SER3/ser_ui_1.py"
Running on local URL:  http://127.0.0.1:7861

To create a public link, set `share=True` in `launch()`.
Received audio data and sample rate
Audio tuple: (44100, array([  0,   0,   0, ..., -77, -46, -54], dtype=int16))
Audio saved successfully
Audio loaded successfully
Noise reduction step
Audio splitting step
Loading models
Feature extraction step
Feature scaling step
Predicting emotion using SVM
```

These things happen in the background when a audio file is recorded and asked to predict

8. BIBLIOGRAPHY

PAPERS REFERRED

1. https://www.irjet.net/archives/V9/i1/IRJET-V9I1101.pdf
2. https://arxiv.org/ftp/arxiv/papers/2002/2002.07590.pdf

WEBSITES REFERRED

1. https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio
2. https://zenodo.org/records/1188976
3. https://github.com/topics/speech-emotion-recognition
4. https://www.sciencedirect.com/science/article/pii/S2667305323000911
5. https://www.projectpro.io/article/speech-emotion-recognition-project-using-machine-learning/573
6. https://www.sciencedirect.com/topics/engineering/confusion-matrix
7. https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/#:~:text=A%20Confusion%20matrix%20is%20an,by%20the%20machine%20learning%20model.