**EmoVoz: An Emotion Recognition System**

A

Minor Project (CC3270)

Report

Submitted in the partial fulfillment of the requirement for the award of Bachelor of Technology

in

Computer and Communication Engineering

By:

**Aditya Mahesh Chougule**

**(219303041)**

Under the guidance of

**Vivek Sharma**



April 2024

Department of Computer and Communication Engineering

School of Computing and Intelligent Systems Manipal University Jaipur

VPO. Dehmi Kalan, Jaipur, Rajasthan, India – 303007

Department of Computer and Communication Engineering

School of Computing and Intelligent Systems, Manipal University Jaipur, Dehmi Kalan, Jaipur, Rajasthan, India- 303007

# STUDENT DECLARATION

*I hereby declare that this project* ***EmoVoz:An Emotion Recognition System*** *is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the University or other Institute, except where due acknowledgements has been made in the text.*

Place: Manipal University Jaipur **ADITYA MAHESH CHOUGULE**

Date: 15/04/2024 (219303171)

B.Tech (CCE) 6th Semester

Department of Computer and Communication Engineering

School of Computing and Intelligent Systems, Manipal University Jaipur,

Dehmi Kalan, Jaipur, Rajasthan, India- 303007

Date:

# CERTIFICATE FROM GUIDE

*This is to certify that the work entitled “****EmoVoz: An Emotion Recognition System”*** *submitted by* ***ADITYA MAHESH CHOUGULE*** *(219303041) to* ***Manipal University Jaipur*** *for the award of the degree of* ***Bachelor of Technology*** *in* ***Computer and Communication Engineering*** *is a bonafide record of the work carried out by him/ her under my supervision and guidance from Jan 08, 2024 to April 26,*

*2024.*

**Vivek Sharma**

*Department of Computer and Communication Engineering*

*Manipal University Jaipur*

# ACKNOWLEDGEMENT

Presentation, inspiration, and motivation have always played a key role in the success of any venture. I extend my heartfelt gratitude to Mr. Vivek Sharma, my esteemed guide, whose unwavering support and expertise have been instrumental in the successful completion of this project. His guidance and mentorship have not only enriched my knowledge but also inspired me to delve deeper into the intricacies of EmoVoz: An Emotion Recognition System.

I am deeply appreciative of Vivek Sharma's dedication and commitment to excellence, which have consistently motivated me to strive for the highest standards of research and innovation. His insightful feedback, constructive criticism, and encouragement have been invaluable assets that have shaped my understanding and approach towards Deep Learning.

I am privileged to have had the opportunity to work under the guidance of Vivek Sharma, whose profound knowledge, passion for guiding, and unwavering support have been pivotal in my academic and professional growth. I am grateful for his mentorship and grateful for the invaluable lessons learned during this enriching journey.

**Aditya Mahesh Chougule**

**(219303041)**

# ABSTRACT

Speech emotion recognition systems play a vital role in understanding human affective states, with applications ranging from human-computer interaction to mental health assessment. Speech Emotion Recognition (SER) presents a complex challenge within Human-Computer Interaction (HCI) systems. The project aims to develop a deep learning model for emotion recognition from speech signals. Emotion recognition is a crucial task in various fields such as human-computer interaction, psychology, and healthcare. By analyzing audio data, the model can classify human emotions into predefined categories such as happy, sad, angry, etc. The project utilizes Convolutional Neural Networks (CNNs) for their effectiveness in processing sequential data like audio signals.

The dataset used for training and testing the model consists of audio recordings of human speech with corresponding emotion labels. The dataset includes speech samples from different speakers and emotional states to ensure diversity and robustness in the model's training.

The workflow of the project involves preprocessing the audio data, extracting relevant features (MFCC,RMSE value, Zero Crossing rate,Mel Spectrogram) using librosa library, building and training a CNN model using TensorFlow and Keras, evaluating the model's performance using classification metrics, and finally saving the trained model for future use.

The performance of the proposed method has been evaluated using two datasets,

TESS (Toronto emotional speech set data) and RAVDESS (Ryerson Audio Visual Database of Emotional Speech and Song). The results show that the proposed method can recognize speech emotions in datasets with an average accuracy of 84% and 83%, respectively.

## TABLE OF CONTENTS

Student declaration 2

Certificate from Guide 3

Acknowledgement 4

Abstract 5

1. Introduction 7
2. Software 8
3. Methodology and Implementation 9
4. Results & Observation 17
5. Testing 22
6. Conclusion 29

Bibliography 30

1. **INTRODUCTION**

In the era of rapidly evolving Artificial Intelligence (AI), there's a deep focus on studying human-computer interactions (HCI). With technologies like Siri and Alexa becoming ubiquitous, the proximity between humans and AI is palpable. The ability to comprehend human emotions opens pathways to understanding people's requirements better. Speech Emotion Recognition (SER) systems play a pivotal role in classifying emotions within speech utterances, significantly contributing to the advancement of HCI, healthcare, customer satisfaction, social media analysis, stress monitoring, and intelligent systems. Furthermore, SER systems find utility in online tutorials, language translation, intelligent driving, and therapy sessions. In specific scenarios, human interaction can be substituted by computer-generated characters capable of natural actions and convincing communication through the expression of human-like emotions.

For this synergy between humans and machines to thrive, machines must interpret emotions conveyed through speech utterances. Only then can we achieve a fully expressive dialogue rooted in mutual trust and understanding between humans and machines.

The project at hand harnesses the power of Convolutional Neural Networks (CNNs) to create an efficient Speech Emotion Recognition system. Leveraging datasets like TESS (Toronto emotional speech set data) and RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song), the model is trained to accurately classify emotions such as happiness, sadness, anger, and more from audio inputs.

By integrating CNNs with emotion recognition, this project contributes to the growing field of affective computing, which aims to imbue AI systems with the ability to understand and respond appropriately to human emotions. The outcomes of this project not only enhance HCI but also pave the way for more emotionally intelligent AI systems that can adapt and empathize with human users in various contexts.

1. **SOFTWARE**

* ***Anaconda Distribution*:**
  + Version: Latest stable version

(Anaconda 2.4.2)

* + Description: Anaconda provides a comprehensive Python distribution with essential packages for data science and machine learning projects.
* ***Jupyter Notebook*:**
  + Version: Jupyter Notebook 6.5.4
  + Description: Jupyter Notebook is an interactive computing environment that allows you to create and share documents containing live code, visualizations, and explanatory text.
* ***Python*:**
  + Version: Python 3
  + Description: Python is a programming language used for developing and executing the code in Jupyter Notebook.
* ***Libraries and Packages*:**
  + NumPy: Version 1.24.3
  + Pandas: Version 1.5.3
  + Matplotlib: Version 3.7.1
  + Seaborn: Version 0.12.2
  + scikit-learn: Version 1.3.0
  + TensorFlow: Version 2.15.0
  + Keras: Version 2.15.0
  + librosa: Version 0.10.1
  + Ipython: Version 8.23.0
  + StreamLit: Version 1.33.0
* ***Other Tools:***
  + Spyder IDE: Python ide with advance editing, interactive testing, debugging and introspecting features

**3. METHODOLOGY & IMPLEMENTATION**

* **Data Acquisition:** Data sets used are from Kaggle :

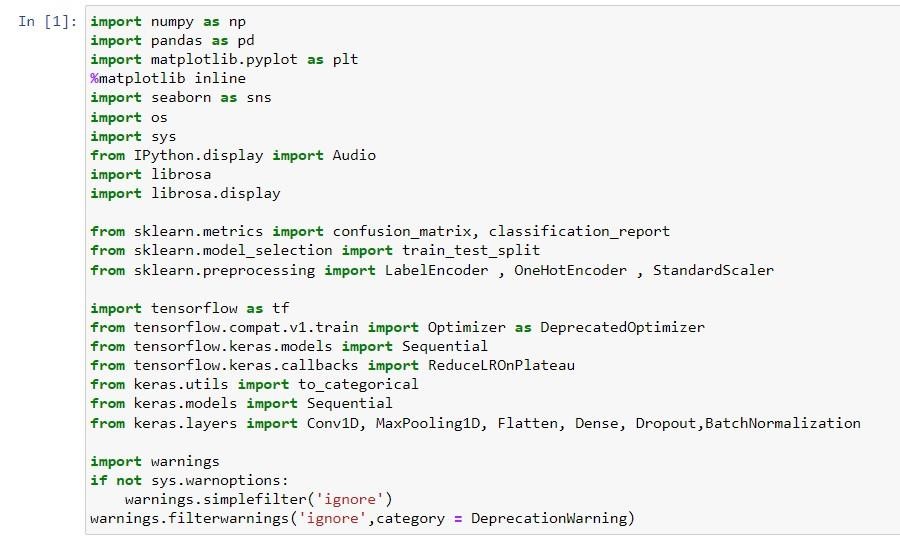
* + ***Ryerson Audio-Visual Database of Emotional Speech and Song (Ravdess):*** This portion of the RAVDESS contains 1440 files: 60 trials per actor x 24 actors = 1440. The RAVDESS contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech emotions includes calm, happy, sad, angry, fearful, surprise, and disgust expressions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.
  + ***Toronto emotional speech set (Tess):*** There are a set of 200 target words were spoken in the carrier phrase "Say the word \_' by two actresses (aged 26 and 64 years) and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). There are 2800 data points (audio files) in total.The dataset is organised such that each of the two female actor and their emotions are contain within its own folder. And within that, all 200 target words audio file can be found. The format of the audio file is a WAV format

* **Data Preparation**:

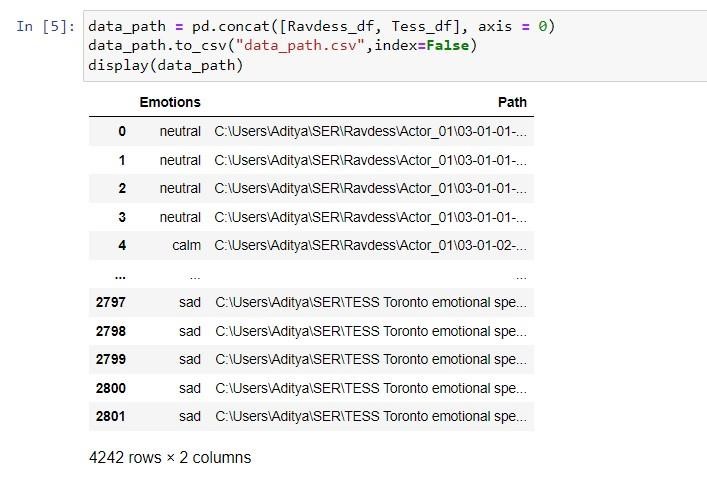
As we are working with two different datasets, so we will be creating a data frame using pandas library storing all emotions of the data in data frame with their paths.

We will use this data frame to extract features for our model training.

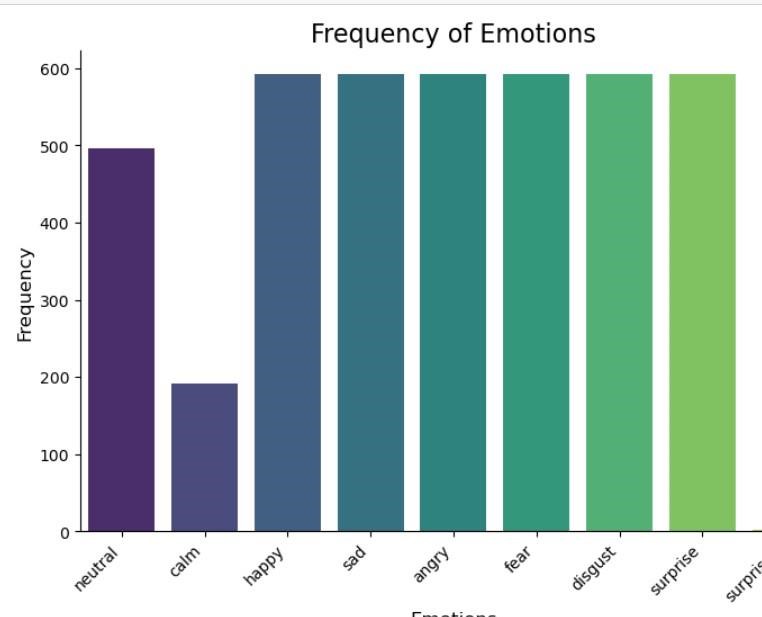
Importing required libraries:



Merging the datasets:



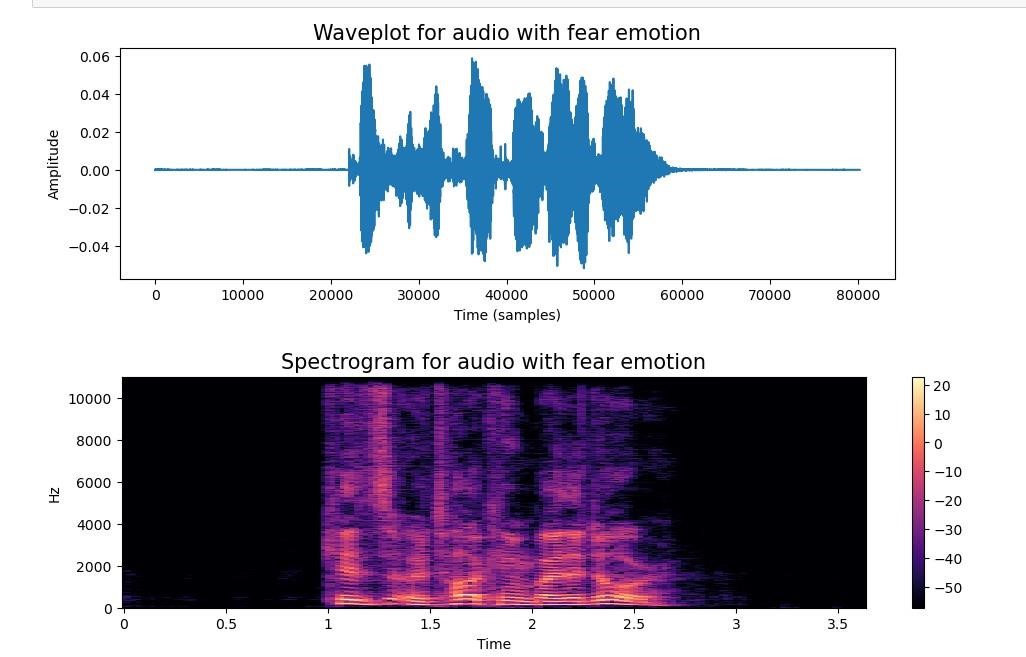
* **Data Visualization:**
  + *Plotting the count of each emotion:*



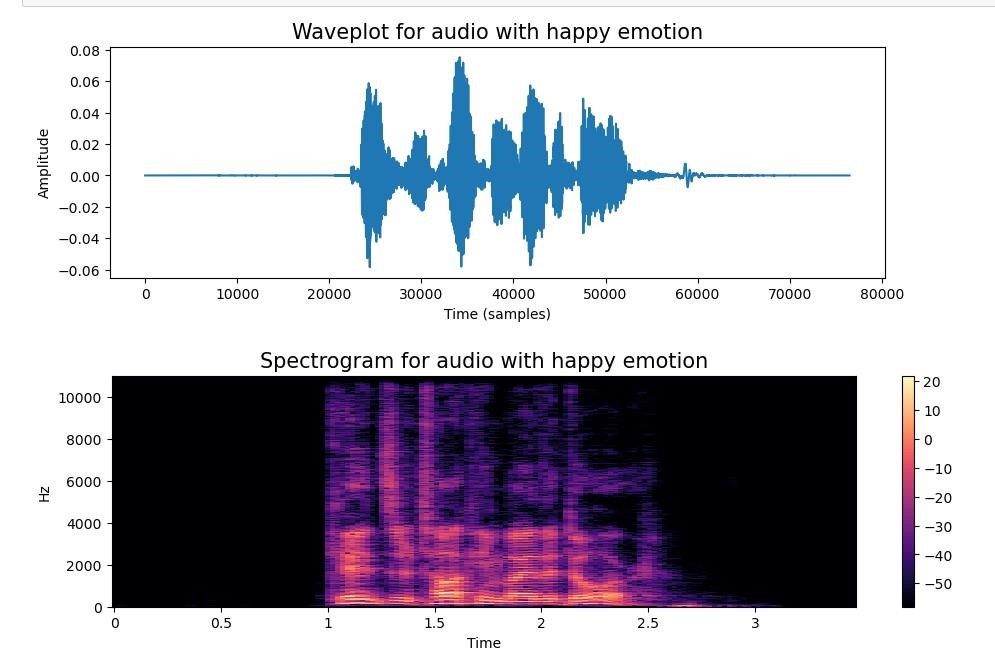
* + *Plotting Waveplots and spectrograms for audio signals:*

* Waveplots - Waveplots let us know the loudness of the audio at a given time.
* Spectrograms - A spectrogram is a visual representation of the spectrum of frequencies of sound or other signals as they vary with time. It’s a representation of frequencies changing with respect to time for given audio signals.

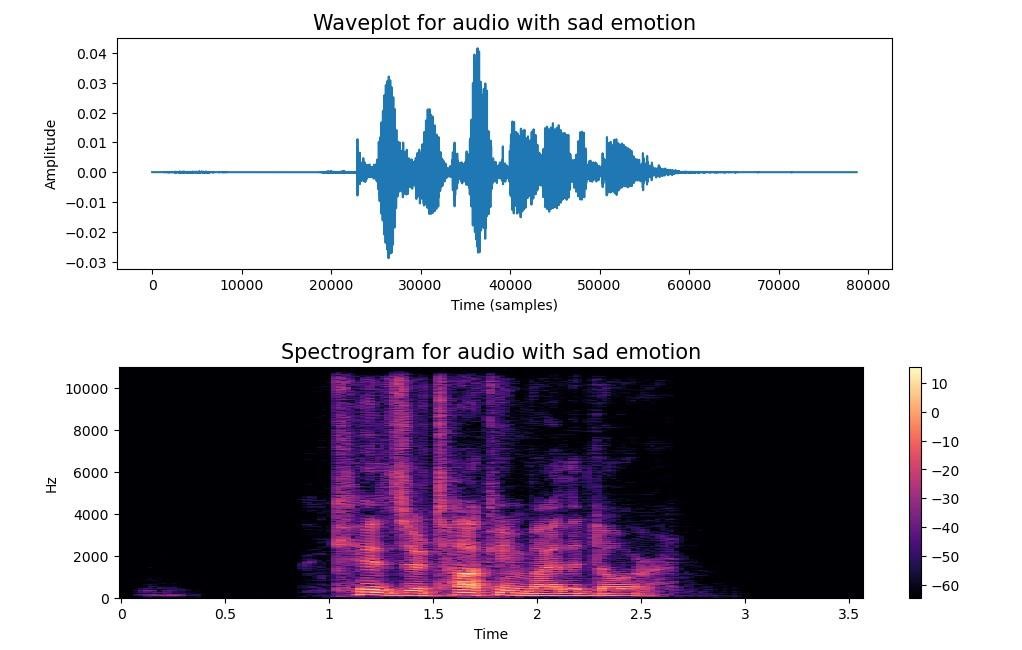
Fear Emotion:



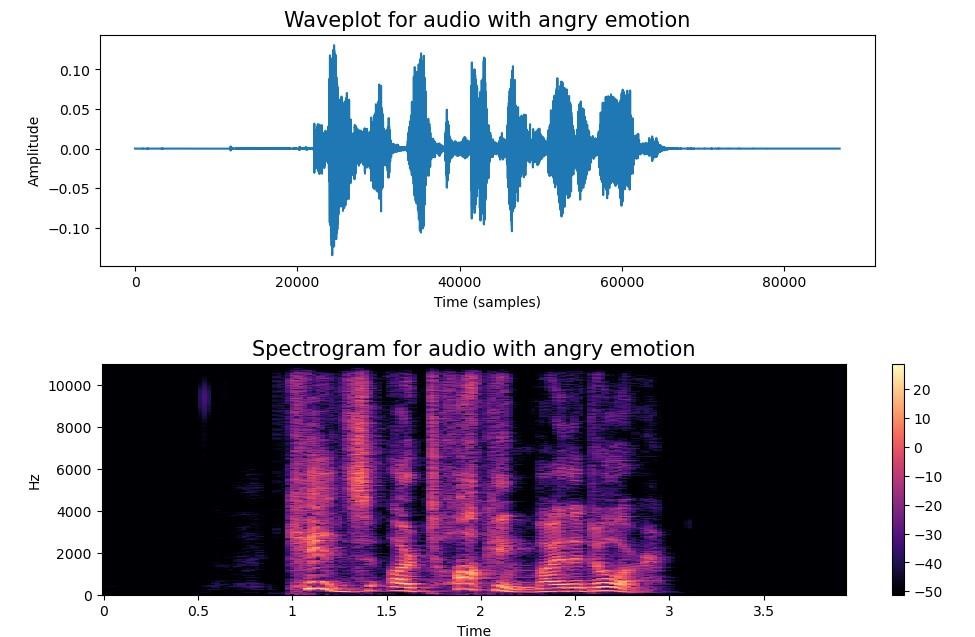
Happy Emotion:



Sad Emotion:



Angry Emotion:



* **Data Augmentation:**
* Data augmentation is the process by which we create new synthetic data samples by adding small perturbations on our initial training set.
* Data augmentation is a technique used in machine learning and deep learning to artificially increase the size of a training dataset by creating modified versions of existing data samples. The goal of data augmentation is to improve the generalization and robustness of a machine learning model by exposing it to a wider variety of training examples without collecting new data from the real world.

Data augmentation techniques are used for audio data in this project are:

**Noise Injection (noise function):**

* Adds random noise to the audio data to simulate real-world variations in audio signals. The amount of noise added is controlled by a random amplitude factor.

**Time Stretching (stretch function):**

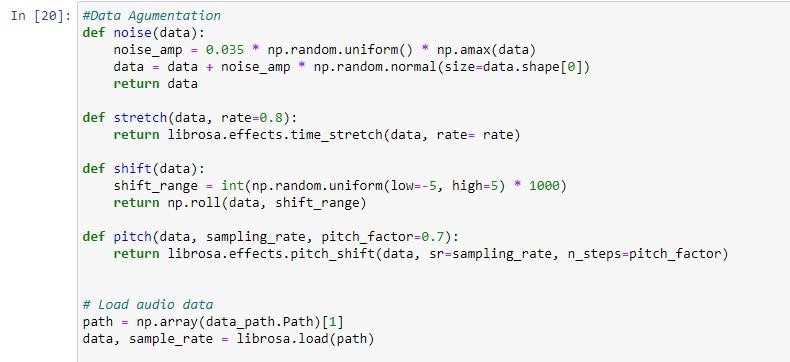
* Stretches or compresses the time scale of the audio data. This technique alters the duration of the audio signal while preserving its pitch.

**Shifting (shift function):**

* Shifts the audio data by a random time range. This technique simulates changes in the starting point or temporal alignment of the audio signal.

**Pitch Shifting (pitch function):**

* Changes the pitch of the audio data by a specified pitch factor. This technique modifies the frequency content of the audio signal while preserving its temporal structure



* **Feature Extraction:**

The feature extraction methods used in the code you provided are primarily based on audio signal processing techniques. These methods extract relevant features from audio data that are used as input to machine learning models, such as Convolutional Neural Networks (CNNs), for Speech Emotion Recognition (SER).

**Zero Crossing Rate (ZCR):**

* + - Calculates the rate at which the audio signal changes its sign (crosses the zero amplitude line). ZCR is a measure of the frequency of changes in the audio waveform.
    - Extracted using librosa.feature.zero\_crossing\_rate().

**Chroma-based Features (Chroma\_stft):**

* + - Represents the pitch content of the audio signal by computing the chroma energy normalized by the total energy.
    - Extracted using librosa.feature.chroma\_stft().

**Mel-frequency Cepstral Coefficients (MFCC):**

* + - Captures the spectral characteristics of the audio signal by computing the Mel-frequency cepstral coefficients, which are a representation of the short-term power spectrum of sound.
    - Extracted using librosa.feature.mfcc()

**Root Mean Square (RMS) Value:**

* + - Computes the root mean square value of the audio signal, which represents the average power of the signal.
    - Extracted using librosa.feature.rms().

**Mel Spectrogram (Mel):**

* + - Computes the Mel spectrogram of the audio signal, which represents the frequency content of the signal in Mel scale.
    - Extracted using librosa.feature.melspectrogram()



* **Data Preparation for modelling:** Normalize the split after extraction



* **Model Building:**

The model being used is a Convolutional Neural Network (CNN) architecture implemented using the Keras library. CNNs are particularly effective for processing and extracting features from sequential data like audio signals, making them well-suited for tasks such as Speech Emotion Recognition (SER).

**Initialization:**

• The Sequential model is initialized using **model = Sequential()**. This creates an empty neural network model that can be sequentially built by adding layers.

Here are the key components of the CNN model used in the project:

* **Conv1D Layers:** Convolutional layers (**Conv1D**) are employed to perform one-dimensional convolutions on the input audio data. These layers are capable of learning spatial patterns in the input signals, which is crucial for capturing relevant features in audio data.
* **MaxPooling1D Layers:** Max Pooling layers (**MaxPooling1D**) are utilized to down sample the feature maps generated by the convolutional layers. Max Pooling helps in retaining the most important features while reducing the spatial dimensions of the data, thereby enhancing computational efficiency and reducing overfitting.
* **Flatten Layer:** After the convolutional and pooling layers, a Flatten layer (**Flatten**) is added to transform the multidimensional feature maps into a one-dimensional vector. This step is necessary to feed the extracted features into the subsequent fully connected layers.
* **Dense Layers:** Dense layers (**Dense**) represent fully connected layers in the network. These layers integrate the extracted features and perform nonlinear transformations to learn complex patterns and relationships within the data.
* **Dropout Layers:** Dropout layers (**Dropout**) are included to prevent overfitting by randomly deactivating a fraction of neurons during training. This regularization technique helps in improving the generalization ability of the model.
* **Output Layer:** The final output layer consists of a Dense layer with **units=8** and **activation='softmax'**. This layer uses softmax activation to generate probability distributions over the eight emotion classes (neutral, calm, happy, sad, angry, fear, disgust, surprise) for SER classification.



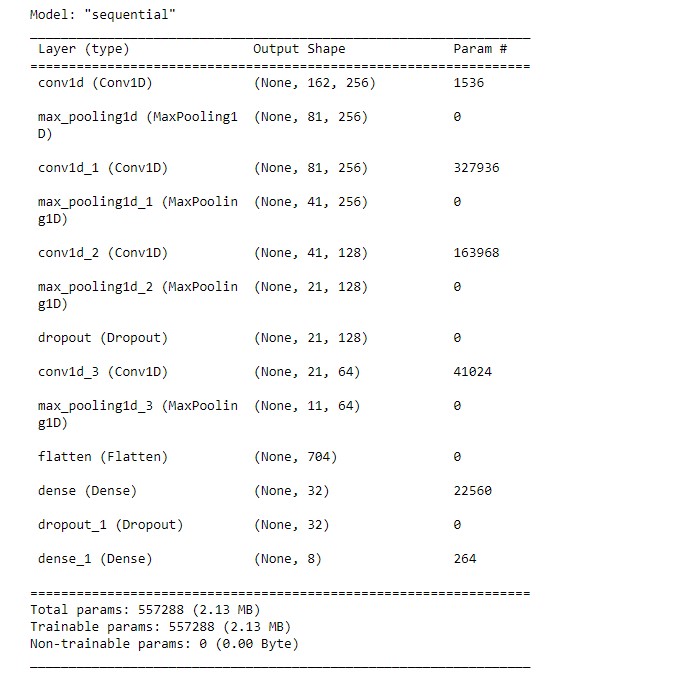
* **Deployment:**

Using streamlit app a basic frontend is made to accept audio file from user and predict the emotion.

Save the model is h5 format or make a pickle model to load it for deployment.

**RESULTS & OBSERVATION**

* **Model Summary:**



* ***Conv1D Layers*:** 
  + The first Conv1D layer (conv1d) has 256 filters/kernels with a kernel size of 5 and a stride of 1. It operates on input data with a shape of (None, 162, 1), where 162 represents the time steps (or sequence length) of the input.
  + Output shape: (None, 162, 256)
  + Parameters: 256 filters \* (5 weights + 1 bias) = 1536 parameters

* ***MaxPooling1D Layers:*** 
  + The first MaxPooling1D layer (max\_pooling1d) performs max pooling with a pool size of 5 and a stride of 2 on the output of the first Conv1D layer.
  + Output shape: (None, 81, 256) after pooling (halves the length)
  + No parameters (pooling is parameter-free)

* ***Additional Conv1D and MaxPooling1D Layers*:**

• Similar Conv1D and MaxPooling1D layers are repeated two more times (conv1d\_1, max\_pooling1d\_1, conv1d\_2, max\_pooling1d\_2) but with different filter sizes and number of filters. These layers progressively reduce the temporal dimension of the data while increasing the number of feature maps.

* ***Dropout Layer:***

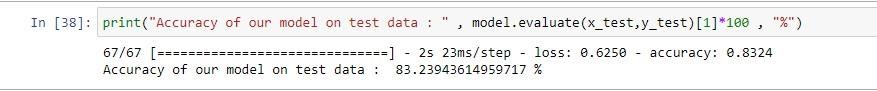
• A Dropout layer (dropout) with a dropout rate of 0.2 is added after the third Conv1D layer. Dropout is used for regularization to prevent overfitting by randomly deactivating neurons during training.

* **Flatten Layer:** 
  + The Flatten layer (flatten) reshapes the output from the last MaxPooling1D layer into a one-dimensional vector, preparing it for the subsequent Dense layers.
  + Output shape: (None, 704) after flattening

* **Dense Layers:** 
  + Two Dense layers (dense, dense\_1) follow the Flatten layer. The first Dense layer has 32 units/neurons with ReLU activation and the second Dense layer has 8 units with softmax activation, which corresponds to the number of output classes (emotions).
  + The first Dense layer introduces non-linearity and higher-level feature representation, while the second Dense layer produces probabilities for each emotion class.
  + Total trainable parameters: 557,288 (2.13 MB), including weights and biases across all layers.
* **Performance Evaluation :**
* ***Accuracy***:

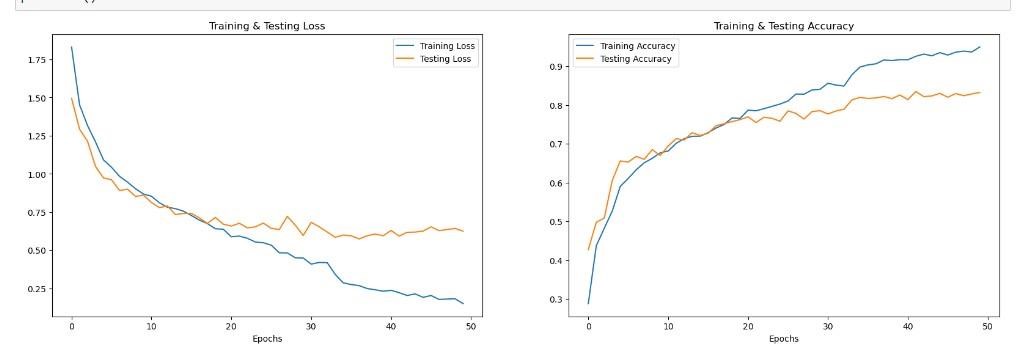
It represents the proportion of correctly classified instances (samples or observations) out of the total number of instances. In other words, accuracy measures how often the model's predictions match the actual labels in the dataset.

The formula for accuracy is:



Obersvation: The built model is 83% accurate.

* ***Training and Testing loss & Accuracy:***



An epoch refers to one complete pass through the entire training dataset during the training of a machine learning model.**Epochs=50** in the **model.fit** function, which means that the model will go through 50 iterations over the training data during model training.

In the 1st graph the loss decreases as the epoch increases.

In the 2nd graph the accuracy increases as epoch increases.

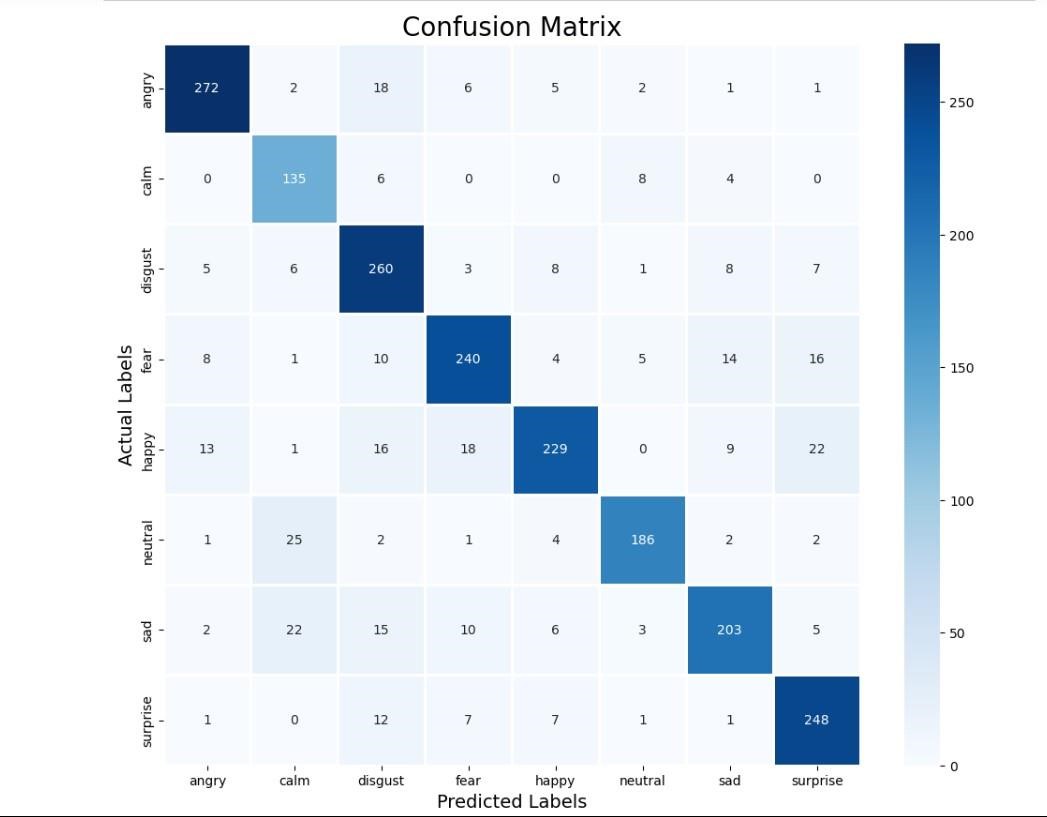
* ***Confusion Matrix:***

A confusion matrix is a table that is often used to evaluate the performance of a classification model. It allows us to visualize the performance of a model by comparing the actual labels of the dataset with the predicted labels generated by the model.

Here's how a confusion matrix is structured:

* True Positives (TP): The number of instances where the actual class is positive, and the model correctly predicts it as positive.
* True Negatives (TN): The number of instances where the actual class is negative, and the model correctly predicts it as negative.
* False Positives (FP): Also known as Type I errors, these are instances where the actual class is negative, but the model incorrectly predicts it as positive.
* False Negatives (FN): Also known as Type II errors, these are instances where the actual class is positive, but the model incorrectly predicts it as negative.

For multi-class classification, the confusion matrix will have rows and columns corresponding to each class, showing how the model's predictions align with the actual labels for each class.



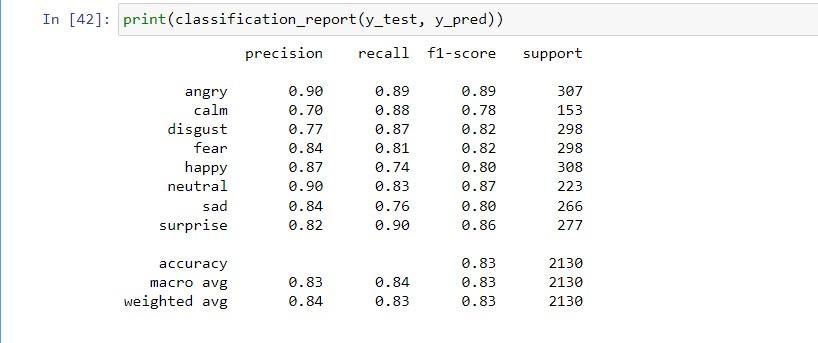
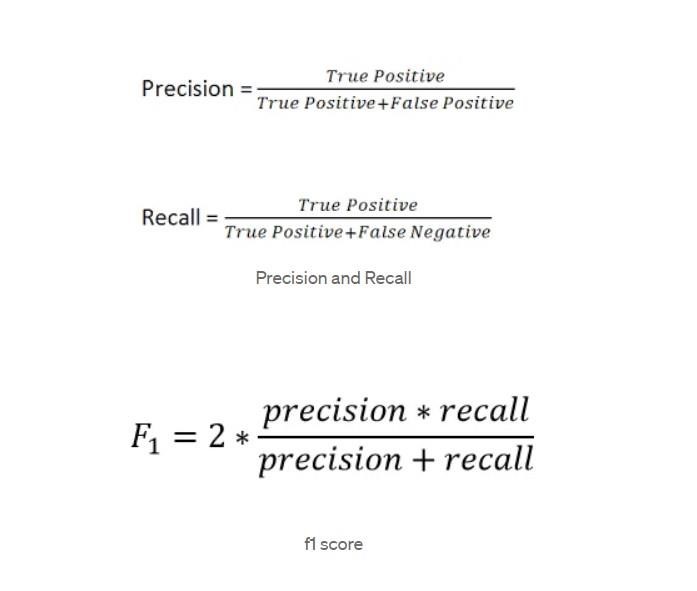
*For angry there are 272 labels which are truly labelled as angry and have been correctly predicted as angry by the model.*

* ***Classification Report***:

The classification report provides a detailed overview of the model's performance across different emotion categories.

**Overall Model Performance:**

* The model achieves an overall accuracy of 83%, indicating that it performs reasonably well in classifying emotions in speech data.
* The macro average precision, recall, and F1-score are all around 0.83, suggesting balanced performance across different emotion classes.



**Class-wise Performance:**

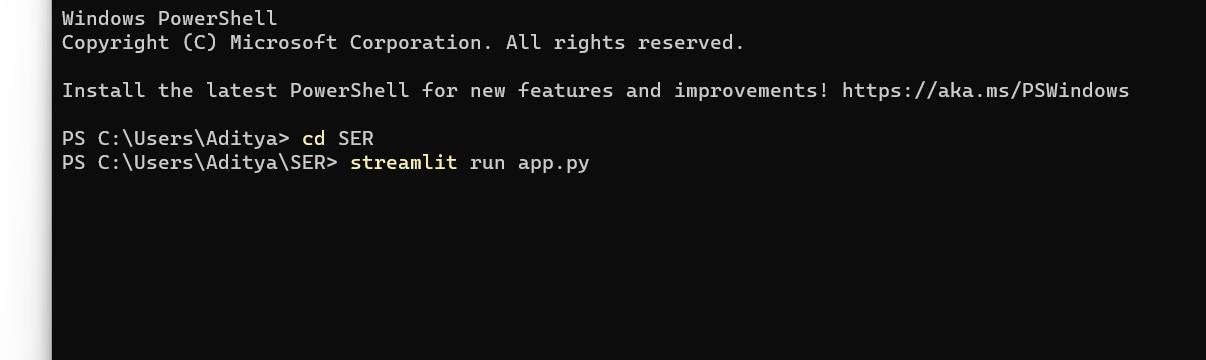
* **Angry:** The model shows high precision, recall, and F1-score for detecting angry emotions, with scores above 0.89, indicating robust performance.
* **Calm:** While the precision for calm emotions is good at 0.70, the recall and F1-score are higher, indicating that the model is better at identifying calm emotions when they are present.
* **Disgust, Fear, Happy:** These emotions also show good performance, with precision, recall, and F1-score above 0.77 for disgust, fear, and happy categories.
* **Neutral, Sad, Surprise:** These categories exhibit varying but acceptable performance, with scores ranging from 0.76 to 0.90, indicating that the model can classify these emotions reasonably well.

**Weighted Average Metrics:**

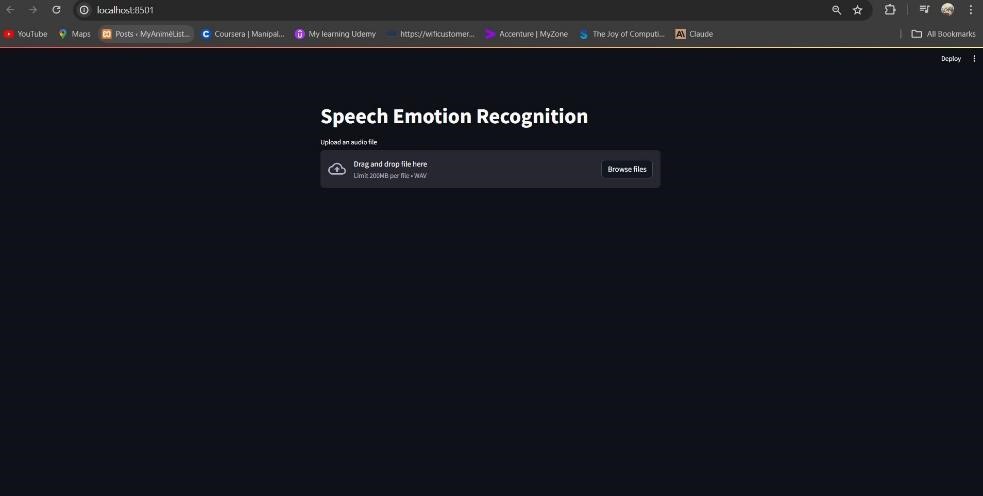
* The weighted average precision, recall, and F1-score are around 0.84, indicating good overall performance when considering class imbalances.

## TESTING

Open the StreamLit App:



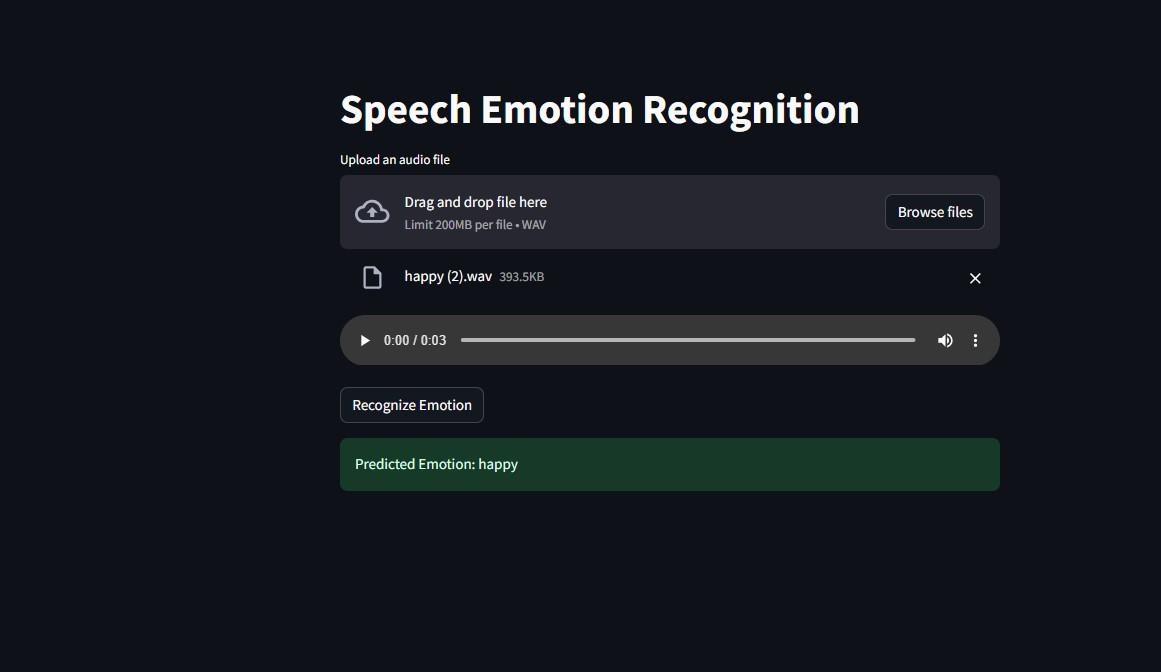
User Interface:



Input Audio file:



Output:



**CONCLUSION**

The model demonstrates strong performance in detecting angry emotions, which are often crucial in speech emotion recognition tasks.

It also performs well in identifying neutral, calm, sad and happy emotions, showcasing its versatility across different emotional states.

While the performance for disgust, surprise, fear emotions is slightly lower, the model still achieves acceptable scores, making it suitable for a wide range of applications.

Overall, the model's accuracy and balanced metrics across classes make it a reliable choice for speech emotion recognition tasks, with potential for further optimization and fine-tuning to enhance performance in specific emotion categories if required.

**BIBLIOGRAPHY**

**Datasets :**

* RAVEDESS: [https://www.kaggle.com/datasets/uwrfkaggler/ravdessemotional-speech-audio](https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio)
* TESS : [https://www.kaggle.com/datasets/ejlok1/toronto-emotionalspeech-set-tess](https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess)

**Feature extraction by Akash Malik**:

[https://medium.com/heuristics/audio-signal-feature-extraction-and-clustering935319d2225](https://medium.com/heuristics/audio-signal-feature-extraction-and-clustering-935319d2225)

**SER project by Tarun Lochab :**

<https://tarun-lohex.medium.com/speech-emotion-recognition-258e88826a98>

**Basics of Machine Learning By Jose Portilla (Udemy):**

[https://www.udemy.com/share/101Wjc3@oxlcO9glGyUKNplDmXM0l6QPygMF UtlL8EX3US6\_s7oZdwE4IYMupaCvzTmhBQR-Ng==/](https://www.udemy.com/share/101Wjc3@oxlcO9glGyUKNplDmXM0l6QPygMFUtlL8EX3US6_s7oZdwE4IYMupaCvzTmhBQR-Ng==/)

**Deployment using streamLit** :

<https://www.youtube.com/watch?v=5XnHlluw-Eo&t=696s>

[https://charumakhijani.medium.com/machine-learning-model-deployment-asa-web-app-using-streamlit-4e542d0adf15](https://charumakhijani.medium.com/machine-learning-model-deployment-as-a-web-app-using-streamlit-4e542d0adf15)

**Speech-Emotion with Machine Learning using CNN by mbam vianney:**

<https://www.youtube.com/watch?v=ApaIKjK2PhU>

**Project implementation:**