

**Annexure-I: Title Page**

**Speech Emotion Recognition using Deep Learning [SER]**

**A Project Report**

**Submitted in partial fulfilment of the requirements for the award of degree of**

**Bachelor of Technology**

**(Computer Science Engineering)**

**Submitted to**



**LOVELY PROFESSIONAL UNIVERSITY**

**PHAGWARA, PUNJAB**

**From 1<sup>st</sup> Sep 2024 to 3<sup>rd</sup> Nov 2024**

**SUBMITTED BY**

**Name of student:** [Kedhareswer Naidu](#)

**Roll Number:** [RK21UWA32](#)

**Reg Number:** [12110626](#)

**Faculty:** [Ajay Sharma Sir](#)

**Course Code:** [CSM422](#)

**Section:** [K21UW](#)

## **Annexure-II: Student Declaration**

**To whom so ever it may concern**

I, **Kedhareswer Naidu, 12110626**, hereby declare that the work done by me on “**Speech Emotion Recognition using Deep Learning [SER]**” from Sep 2024 to Nov 2024, is a record of original work for the partial fulfilment of the requirements for the award of the degree, Bachelor of Technology.

**Name of the student:** Kedhareswer Naidu

**Registration Number:** 12110626

**Dated:** 03<sup>rd</sup> October 2024

## **ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to the teacher and instructor of the course Machine Learning who provided me the golden opportunity to learn a new technology.

I would like to also thank my own college Lovely Professional University for offering such a course which not only improve my programming skill but also taught me other new technology.

Then I would like to thank my friends who have helped me with their valuable suggestions and guidance for choosing this course.

Finally, I would like to thank everyone who have helped me a lot.

**Dated:** 03<sup>rd</sup> October 2024

## Table of Contents

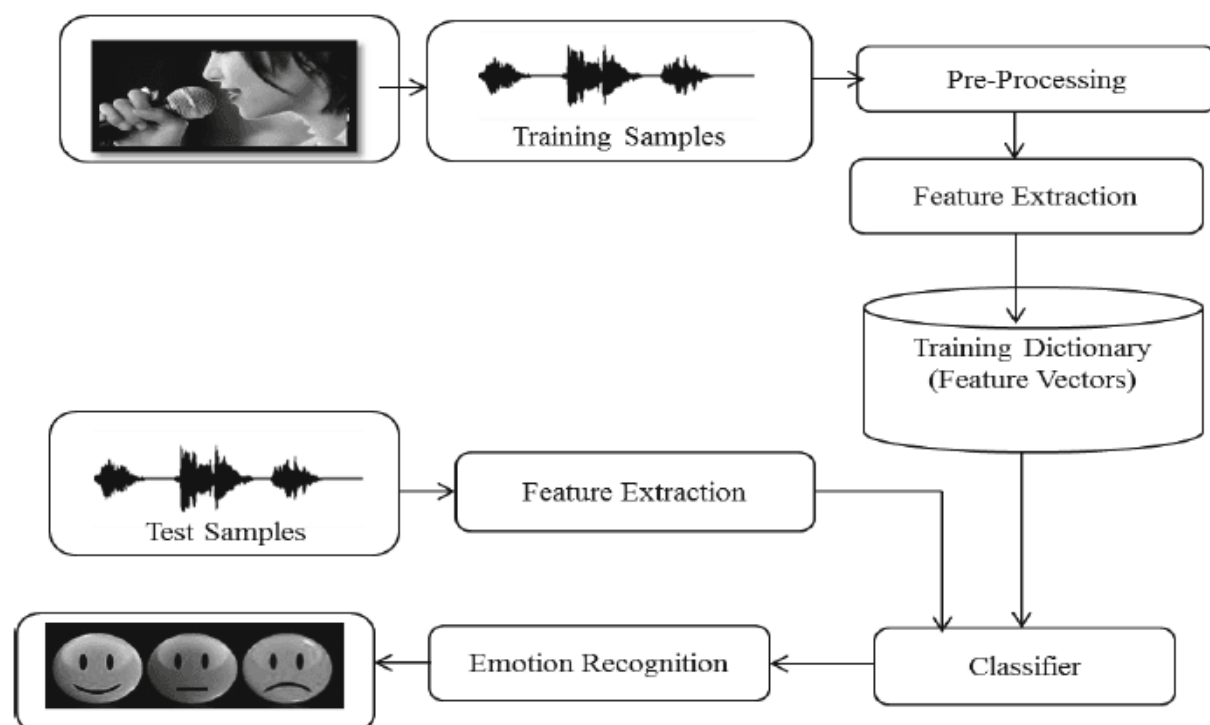
S. No.	Contents	Page
1	Title	1
2	Student Declaration	2
3	Acknowledgement	2
4	Table of Contents	3
5	Abstract	4
6	Objective	5
7	Introduction	6
8	Theoretical Background	7
9	Hardware & Software	9
10	Methodology	10
11	Flowchart	12
12	Results	23
13	Conclusion	13
14	Bibliography	24

## **ABSTRACT:**

The field of *Speech Emotion Recognition (SER)* endeavors to identify the emotions of human beings through the *pitch, tone, rhythm, and intensity* characteristics of their speech. The role of emotions in the communication process is of utmost importance. The SER system is an effort to bring in similar emotional perception in *human-computer communication, customer service, health monitoring, and security applications*. The work will use deep learning mechanisms for the emotion classification of audio samples, utilizing *Convolutional Neural Network (CNN)* architectures in the main phase. The model will be trained and validated using four different publicly available data sets: *RAVDESS, TESS, SAVEE, and CREMA-D*.

To trim these data, some of the most common feature extraction methods are used: *ZCR, RMS, MFCC*. The work analyzes the data that capture the characteristics of audio connected to emotion. Techniques such as noise injection, pitch alteration, and time shifting supplement the dataset to ensure the model's robustness and enhance its generalizability. The CNN model achieved a pre-training accuracy of 99.42%, showing its efficacy in internet audio signals.

More responsive and intuitive systems can be used with the help of the SER model, which can adapt their responses according to the emotion of the user. It can be used to make personal assistants more personal and customer service more effective, ultimately supporting the possibility of mental health monitoring and security surveillance. This study, thus, presents the world of promise that deep learning hangs on SER and gives ground for further future works with extended data sets and optimal models.

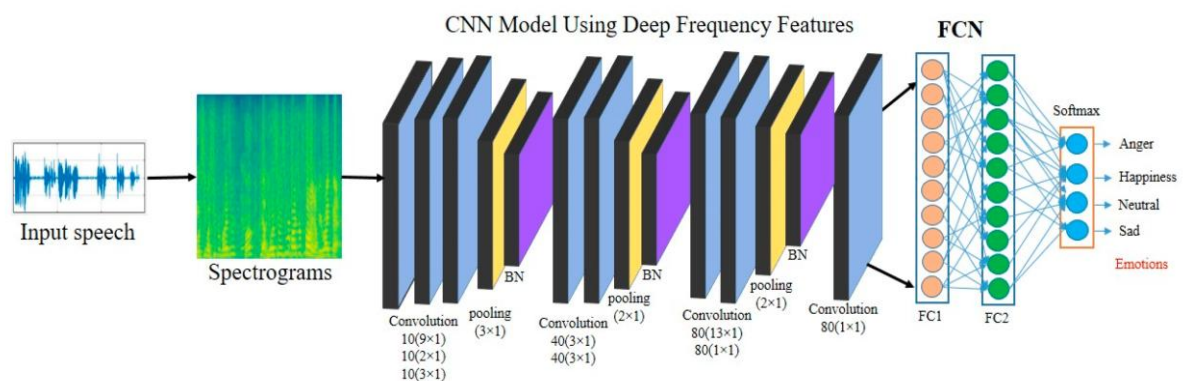


## **OBJECTIVE:**

The overall aim of this project is to create and develop a *Speech Emotion Recognition (SER)* model that is efficient and reliable, based on deep learning, to be specific, *Convolutional Neural Networks (CNNs)*. By subjecting emotional speeches from different datasets to processings, the goal is to arrive at accurate classification of the emotions whilst investigating how SER can further various applications that include human-computer interaction, customer support, and mental health checks. This project involves not just the development and evaluation of a high-performance model, but due regard will be given to data preparation, feature extraction, and generalization to enhance the model's adaptability to varied real-world audio inputs. Our objectives can be detailed as follows:

1. **Developing a Robust SER Model:** Based on the CNN model, present an emotion recognition and classification method to classify the speech signals with high performance for different datasets.
2. **Analyzing and Preprocessing Different Datasets:** Several publicly available datasets are pre-processed into one more complete dataset that contains various emotions, accents, and speech patterns. The different datasets include RAVDESS, TESS, SAVEE, and CREMA-D.
3. **Apply Data Augmentation:** We further enhance the robustness of our model by incorporating data augmentation through noise injection, pitch shifting, and altering speed. This would effectively simulate a variety of acoustic conditions and thereby improve performance on a variety of audio samples.
4. **Feature Extraction and Audio Feature Engineering:** Extract such features as ZCR, RMS, and MFCCs from the audio files. These represent some of the most critical properties in speech with which emotions are related and will be used for model training.
5. **Enhance Model Generalization:** Make sure the model generalizes well to new, previously unseen data. This should be ensured through training with diverse speech samples and augmentations that increase its adaptability to real-world scenarios.
6. **Model Performance Testing:** Accuracy, precision, recall, and F1-score will be measured to assess the performance of the model, trying to keep the classification as high as possible in order to reduce misclassifications on test datasets.
7. **Practical Applications of SER:** Emphasize the research into applications of SER in domains such as human-computer interaction, customer service, mental health support, and virtual assistants, among other areas where emotional awareness provides impact.
8. **Pave the Way for Future Research in SER:** Provide the basis with a foundation and insights for future research in SER, including future improvements with larger datasets, more advanced neural architectures, and real-time emotion detection capabilities.

This outline makes clear the role of each objective in constructing a robust and adaptable SER system and emphasizes the broader impact and future directions for SER research and applications.



## INTRODUCTION:

### 1. Problem Statement:

Human *emotions* play an important part in communication-they influence how we interpret events and how we respond to, and interact with, each other. *Traditional human-computer interactions*, however, are emotionally obtuse and prevent systems from responding to the user in a natural and *context-sensitive manner*. This omission turns out to be particularly serious in many applications such as *customer support, health care, security, and personal assistance* where the knowledge of a *user's emotional state* would allow for better service, mental condition assessment, or increased user satisfaction.

Emotion Recognition in Speech addresses this problem through the design of systems capable of detecting and interpreting human emotions by analyzing voice pitch, tone, intensity, and rhythm. Thus, it allows the models to detect emotions such as happiness, sadness, anger, fear, and neutrality, providing intuitive and emphatic interactions. Despite promising progress, emotion detection in speech remains tough owing to individualistic expression, accent, background noise, and several factors affecting the speech signals.

The problem this project centers around is:

- How can a deep learning model - Convolutional Neural Network, especially - be effectively trained for speech emotion recognition across datasets with huge variability?
- What approaches in the aspects of data preparation, feature extraction and data augmentation are required in order to enhance the performance and generalize the results?
- How would the resultant SER model be applied practically in real life to applications such as human-computer interaction, customer service, or monitoring mental health?

This work tries to answer these questions with the intent to create a robust and accurate SER model that would adapt to diverse voices and emotionally expressive utterances, thereby laying the bedrock for emotionally more intelligent technologies.

## **2. Background:**

Speech is a copious medium of human communication involving the words we utter and the emotional undertones that form our relationships. Emotions in speech get conveyed with vocal characteristics such as pitch, volume, rhythm, and tone, which humans intuitively detect. By understanding and interpreting the emotions therein, we are able to show empathy, understand their ulterior intentions, and act correspondingly. However, the interaction systems of human-computer interaction cannot recognize emotions, which make them limited to effectiveness and ability regarding fully responsive and intuitive engagement.

Speech Emotion Recognition has emerged as a solution to bridge this gap, enabling computers to conduct real-time emotion detection by analyzing vocal cues. Indeed, it finds wide applications across industries, from customer service, where knowing the feeling improves response quality, to healthcare, where emotion recognition may support mental health diagnostics and monitoring, to security, where the detection of stress or fear might alert police to potential threats.

State-of-the-art machine learning, especially deep learning, justifies advances in the SER model by empowering algorithms with complex patterns in data. More precisely, CNNs were successful in image processing and now have been adapted for various other audio challenges, including speech recognition. However, the most substantial challenge in SER is due to the variability in the expression of human emotions, variation in accent, language, and background noise in real-world settings.

This work leveraged the four benchmark datasets, RAVDESS, TESS, SAVEE, and CREMA-D, for developing an effective CNN-based model on emotional speech recognition. Each dataset contributes its peculiar speech samples, thereby capturing wide expression of emotion with different accents and demographics. This project also exploits data augmentation by including a set of rules to simulate acoustic conditions from diverse environments, ensuring that this model generalizes well across different settings.

The project, therefore, tries to construct a deep learning-based SER model using CNNs and feature extraction techniques. This paper aims to show, by successful development of this model, how practical and impactful SER technology can get, and provide a platform for further research into emotionally intelligent systems.

## **THEORETICAL BACKGROUND:**

The concept of SER is based on an understanding of how emotions impact human speech and the usage of computational methods to determine these emotions. Emotions affect pitch, tone, intensity, and speech rate, which leads to patterns identifiable and classifiable. Principles from signal processing, machine learning, and deep learning are drawn upon to determine the patterns and allow systems to interpret emotions from speech signals.

## 1. What is **Speech Emotion Recognition (SER)**

SER is the analysis of audio signals to determine emotions in speech, such as happiness, sadness, anger, fear, and neutrality. Emotions add a depth of meaning to communication and are necessary for interpreting the context of a message. SER aims to identify these subtle emotional changes by capturing differences in vocal characteristics. This technology has a wide range of applications, including human-computer interaction, customer service, healthcare, education, and security. It enhances user experiences through the introduction of systems into the emotional waters of being emotionally aware; machines respond in more intuitive and empathetic ways.

## 2. What is **Feature Extraction in SER?**

Raw audio data is complex and not easily used by the machine learning models directly; feature extraction simplifies this data by capturing the specific audio features most relevant to emotion recognition. The three major features commonly extracted for SER are:

- **Zero Crossing Rate (ZCR):** It is the measure of how many times per unit of time the audio signal flips from positive to negative, or vice versa. That indicates how noisy and harsh the speech signal sounds. It might be varied based on emotions. Anger would have a high value of ZCR.
- **Root Mean Square (RMS):** This is the energy of the audio signal, showing changes in loudness and intensity. Anger or happiness may have higher energy while sadness has lower energy.
- **Mel Frequency Cepstral Coefficients (MFCCs):** It is the power spectrum of the audio signal on a mel scale, simulating the human auditory perception. MFCCs are valuable in SER because they can capture both spectral and temporal features of speech.

## 3. What is **Data Augmentation for SER?**

As SER is highly dependent on the availability of diverse, well-balanced datasets, data augmentation is important for the extension of limited datasets and improving the robustness of models. Noise injection, time shifting, pitch alteration, and speed changes simulate various acoustic conditions, and the model generalizes well to diverse, real-world scenarios. Data augmentation reduces overfitting and improves the performance of the model by generating new samples that mimic possible variations in real speech.

## 4. **Deep Learning and Convolutional Neural Networks (CNNs)**

Deep learning, in general, and *Convolutional Neural Networks*, in particular, have brought a remarkable impact on SER. Although the *CNN architecture* was mainly designed for the image recognition application, CNN is well known for its high proficiency in perceiving spatial as well as temporal hierarchies in the data; this is helpful in working with spectrograms obtained from the audio input. By that process, complex patterns related to speech may be understood to discern emotion cues against features from the audio by taking MFCCs into consideration. Since the model of this architecture is inherently layer-wise



structured, low-to-high pattern capture of interest can happen progressively during this differentiation among subtle emotion cues of the speech signal.

- **Model Architecture:** A simple SER using a basic CNN consists of several layers of convolution to extract features from the given audio patterns, with pooling layers that reduce data dimensionality, followed by fully connected layers to make classifications. This architecture is able to capture all the details in the audio signal and classify the emotion very accurately.

## **Hardware & Software Requirements:**

### **Hardware:**

To efficiently train and run a Speech Emotion Recognition (SER) model, specific hardware components are recommended to handle the computational demands of deep learning and audio processing tasks. Here are the hardware requirements:

1. **Processor (CPU):**
  - Minimum: Quad-core processor (Intel i5 or AMD Ryzen 5)
  - Recommended: Multi-core processor (Intel i7 or AMD Ryzen 7 and above)
2. **Graphics Processing Unit (GPU)** (for deep learning models or larger datasets):
  - Minimum: NVIDIA GTX 1050 (2 GB VRAM)
  - Recommended: NVIDIA RTX 2060 (6 GB VRAM) or higher for better performance, such as the NVIDIA RTX 3090 (24 GB VRAM) for very large datasets or complex neural networks.
3. **RAM:**
  - Minimum: 8 GB
  - Recommended: 16 GB or higher
  - Sufficient memory helps handle large datasets and improves model training and analysis speed, especially when working with high-dimensional sensor data.
4. **Storage:**
  - Minimum: 256 GB SSD (for faster data access and processing)
  - Recommended: 512 GB SSD or higher, especially if working with larger datasets or multiple versions of synthetic datasets.

### **Software:**

SER projects require a robust software environment with deep learning frameworks, libraries for audio processing, and tools for data handling and visualization. Below are the software requirements for the project:

1. **Operating System:**
  - Windows 10 or higher, macOS, or Linux (Ubuntu is widely used in data science and machine learning)
2. **Programming Language:**

- **Python 3.8 or higher:** Python is widely used in machine learning due to its comprehensive libraries for data processing, model building, and visualization.
- 3. **Integrated Development Environment (IDE):**
  - **Jupyter Notebook or Google Collab** (recommended for interactive data exploration and EDA)
  - **Anaconda** (a popular Python distribution that includes Jupyter Notebook, making it easier to manage packages and dependencies)
  - **VS Code, PyCharm, or Spyder** (optional; these are more suited for code management in larger projects)
- 4. **Libraries and Frameworks:**
  - **Deep Learning Framework:**
    - **TensorFlow or Keras** – For building and training the Convolutional Neural Network (CNN) model.
    - Alternatively, **PyTorch** can be used for its flexibility in deep learning applications.
  - **Audio Processing Libraries:**
    - **Librosa** – For audio loading, feature extraction (e.g., MFCC, ZCR), and waveform analysis.
    - **Soundfile** – For reading and writing sound files in various formats.
    - **Wave** – For handling .wav file formats, if required.
  - **Data Manipulation and Analysis:**
    - **Pandas** – For handling data frames and manipulating datasets.
    - **NumPy** – For numerical computations and efficient data handling.
- 5. **Additional Tools for Documentation and Reporting:**
  - **Microsoft Excel or Google Sheets:** for simple data exploration, report organization, and result presentation
  - **Markdown** (e.g., in Jupyter Notebooks or GitHub): for documentation and creating a clear project report.

## **METHODOLOGY:**

### **Import Required Libraries**

The first step in developing the Speech Emotion Recognition (SER) model involves importing all the necessary libraries for data handling, audio processing, visualization, model building, and evaluation. Each library serves a specific purpose, as outlined below:

In VsCode:

```
import pandas as pd
import numpy as np
import os
import sys

# librosa is a Python library for analyzing audio and music. It can be used to extract the
import librosa
import librosa.display
import seaborn as sns
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import train_test_split

# to play the audio files
from IPython.display import Audio

import keras
from keras.callbacks import ReduceLROnPlateau, EarlyStopping
from keras.models import Sequential
from keras.layers import Dense, Conv1D, MaxPooling1D, Flatten, Dropout, BatchNormalization
from keras.callbacks import ModelCheckpoint

import joblib

import warnings
if not sys.warnoptions:
    warnings.simplefilter("ignore")
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

On Collab:

```
import os
import pickle
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import joblib
from google.colab import files

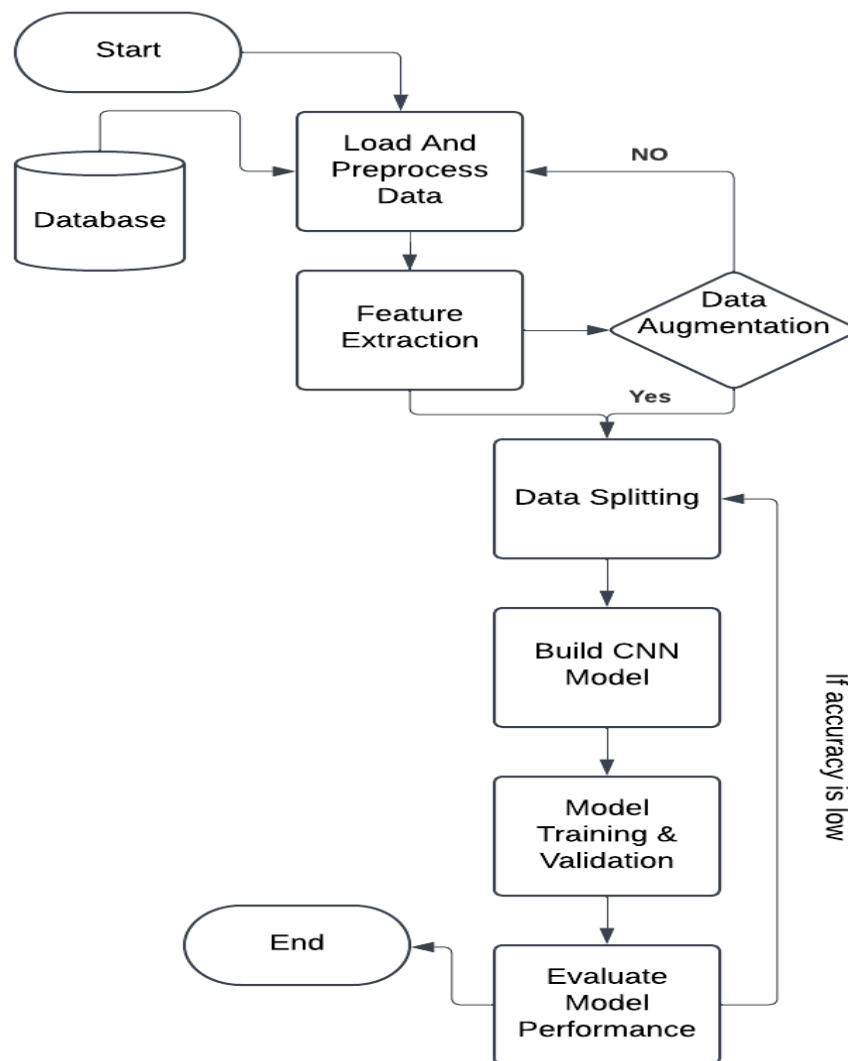
import librosa
from tensorflow.keras.models import load_model
from sklearn.preprocessing import StandardScaler, LabelEncoder

#Label Encoding
import keras.utils
from sklearn.preprocessing import LabelEncoder

from tensorflow.keras.models import Sequential
import tensorflow as tf
from tensorflow.keras.layers import Conv1D, BatchNormalization, AvgPool1D, Dropout, Flatten, Dense

from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
```

## FLOWCHART:



Flow Chart-1: Project Flow

## Data Collection and EDA:

We collect data for this SER project by drawing from four large emotional speech datasets each having their unique vocal samples to different emotions. These datasets were accessed from their official website at

### I. RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song)

This is an audio-visual recording dataset whereby actors express themselves in a set of emotions, for example, happiness, sadness, anger, fear, surprise, and calm.

### II. TESS (Toronto Emotional Speech Set)

TESS Audio files by two actresses that cover all the emotions with a list of phonetically matched sentences.

### III. SAVEE (Surrey Audio-Visual Expressed Emotion)

SAVEE is audio files from male speakers that are recorded for seven different emotions including neutral and surprise.

### IV. CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset)

CREMA-D is multi-modal recordings, that is audio and visual recordings of actors who have covered six basic emotions with various phrases.

### Exploratory Data Analysis (EDA)

Once the datasets were gathered, EDA was conducted in order to understand data distribution, to identify any class imbalances, and examine key characteristics of each dataset. Main steps and insights after EDA are described below.

#### *a. Overview and Structure of Data*

All the datasets were checked for uniformity. The labels across datasets are mapped consistently across datasets such as happy, sad, angry. Counts of samples per emotion were checked in order to understand the distribution and significant imbalances.

#### *b. Audio Feature Visualization*

**Waveforms:** Generation of a plot for waveforms for any audio file produced how amplitudes are changing over time for each audio file. With it, one can see how each of the emotions affects voice strength.

**Spectrograms:** Spectrograms of audio samples were generated for the analysis of frequency content. This gives information about how the frequency components are changing with time for every emotion.

Statistical Distribution of Features

**Zero Crossing Rate (ZCR):** Plotted the ZCR for different emotions to see if the patterns of the vocal rhythm and articulation change with the emotions.

**Root Mean Square (RMS):** Calculated RMS for each emotion, which indicates power or intensity of the audio signal.

**Mel Frequency Cepstral Coefficients (MFCC):** This plots the distribution of MFCCs, which as stated captures properties that may be unique for different emotions.

#### **Class Imbalance Check and Data Augmentation**

- Emotion classes were validated for imbalance with emotions having fewer samples flagged for potential augmentation. Data augmentation techniques including noise injection, pitch shifting, and time-stretching have been applied in order to balance the dataset, thus making the model more robust.

Based on the above EDA steps, much deeper insights have been created into the datasets, and all those feature insights are directed toward the feature extraction and training

phases of the model to enhance model performance.

### Data Links –

Name	Links
RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song)	<a href="https://zenodo.org/records/1188976">https://zenodo.org/records/1188976</a>
TESS (Toronto Emotional Speech Set)	<a href="https://tspace.library.utoronto.ca/handle/1807/24487">https://tspace.library.utoronto.ca/handle/1807/24487</a>
SAVEE (Surrey Audio-Visual Expressed Emotion)	<a href="http://kahlan.eps.surrey.ac.uk/savee/">http://kahlan.eps.surrey.ac.uk/savee/</a>
CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset)	<a href="https://dagshub.com/DagsHub/audio-datasets/src/main/CREMA-D">https://dagshub.com/DagsHub/audio-datasets/src/main/CREMA-D</a>

### Importing Data:

This is an organized manner of importing and organizing audio data from the dataset directory. It goes through every dataset folder, identifies, and stores file paths with corresponding emotion labels for every audio file. It iterates over each folder to extract full file paths and maps every filename to its corresponding emotion label using a conversion function. This structure and labelling is further compiled into a structured DataFrame with columns for file paths and emotion labels, ready to be used for any further analysis and model training. The same approach has been consistently applied to every dataset used in this paper: TESS, RAVDESS, SAVEE, and CREMA-D, providing a standard form and easy data handling through the whole project.

```
def import_cremaD(path):
    folders = [f for f in os.listdir(path) if os.path.isdir(os.path.join(path, f))]
    emo = []
    fullName = []

    for folder in folders:
        folder_path = os.path.join(path, folder)
        files = [f for f in os.listdir(folder_path) if os.path.isfile(os.path.join(folder_path, f))]

        for file in files:
            fullName.append(os.path.join(folder_path, file))
            step = file.split('.')[0]
            emo.append(cremaD_convert(step))

    data = {"path": fullName, "emotion": emo}
    df = pd.DataFrame(data)
    return df
```

```
def import_ravdess(path):
    folders = [f for f in os.listdir(path) if os.path.isdir(os.path.join(path, f))]
    emo = []
    fullName = []

    for folder in folders:
        folder_path = os.path.join(path, folder)
        files = [f for f in os.listdir(folder_path) if os.path.isfile(os.path.join(folder_path, f))]

        for file in files:
            fullName.append(os.path.join(folder_path, file))
            step = file.split('.')[0]
            parts = step.split('-')
            if len(parts) > 2: # Ensure the file name has enough parts
                emotion = int(parts[2])
                if emotion == 2:
                    emotion = 1
                emo.append(emotion)

    if len(fullName) != len(emo):
        raise ValueError("Mismatch between file paths and emotions lists lengths")

    data = {"path": fullName, "emotion": emo}
    df = pd.DataFrame(data)
    return df
```

```
def import_savee(path):
    directory = "C:\\Users\\mbkhn\\Desktop\\ALL"
    exclude_folder = 'audiodata\\AudioData'
    folders = [f for f in os.listdir(path+exclude_folder) ]
    emo = []
    fullName = []
    for folder in folders:
        if path+'AudioData/'+folder == "C:\\Users\\mbkhn\\Desktop\\info.txt":
            continue
        files=os.listdir(path+'AudioData/'+folder)
        for file in files:
            fullName.append(path+'AudioData/'+folder+"/"+file)
            step = file.split('.')[0]
            emo.append(savee_convert(step))

    data={"path":fullName,"emotion":emo}
    df=pd.DataFrame(data)
    return df
```

```
def import_tess(path):
    folders = [f for f in os.listdir(path) if os.path.isdir(os.path.join(path, f))]
    emo = []
    fullName = []

    for folder in folders:
        folder_path = os.path.join(path, folder)
        files = [f for f in os.listdir(folder_path) if os.path.isfile(os.path.join(folder_path, f))]

        for file in files:
            fullName.append(os.path.join(folder_path, file))
            step = file.split('.')[0]
            emo.append(tess_convert(step))

    data = {"path": fullName, "emotion": emo}
    df = pd.DataFrame(data)
    return df
```

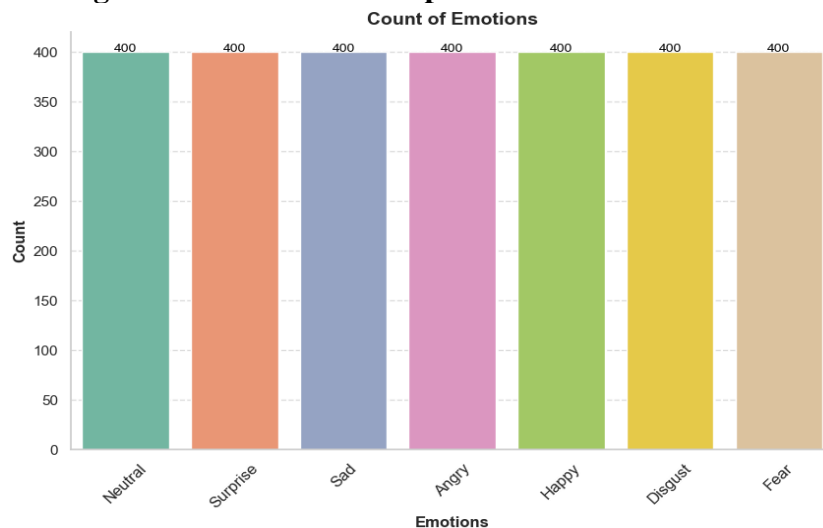
## Data preparation:

The data preparation process for this Speech Emotion Recognition project involved several organized steps to ensure consistency and ease of analysis.

1. First, each emotion label from the datasets was converted into a numeric value based on predefined mappings, allowing the model to process emotion labels as numeric classes.
2. Next, the paths for each dataset (RAVDESS, TESS, SAVEE, and CREMA-D) were defined, ensuring efficient access to audio files.
3. Each dataset was then imported individually into a DataFrame, preserving the paths and emotion labels.
4. These DataFrames were subsequently concatenated into a single comprehensive DataFrame to unify all datasets under a common structure.
5. The combined DataFrame was shuffled to ensure a randomized distribution of data, promoting unbiased training.
6. Finally, a copy of this DataFrame was created, retaining emotion names for visualization, which would later aid in analyzing class distributions and model predictions.

This structured approach ensured a robust and standardized data preparation process across all datasets.

### Even Data Balancing was checked in this step



## Waveplots and Spectrograms for Audio Signals:

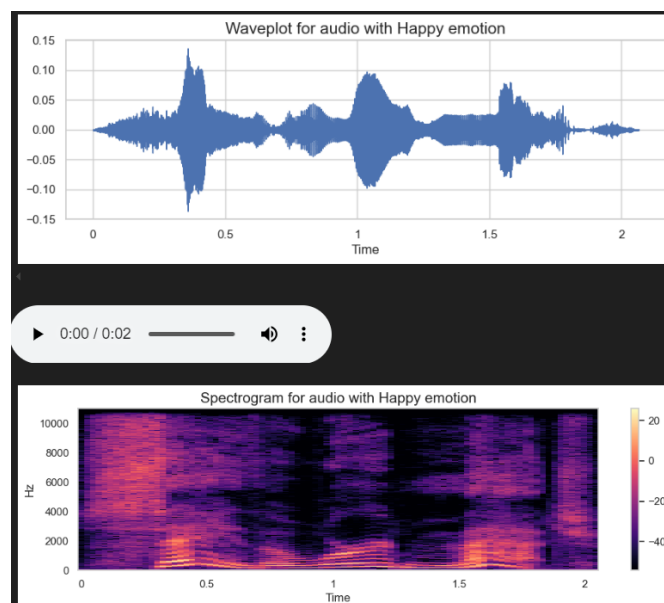
Waveplots are graphical representations that display the amplitude (loudness) of an audio signal over time. Each point on the waveplot corresponds to the instantaneous amplitude of the sound wave at a given moment, allowing us to visualize how loud or quiet the audio is at any particular time.

- **Understanding Loudness:** The vertical axis of the waveplot indicates the amplitude of the audio signal, while the horizontal axis represents time. A higher amplitude indicates louder sound, whereas a lower amplitude indicates quieter sound. This information is crucial for analyzing the dynamics of an audio signal, as it helps in identifying peaks (louder sections) and troughs (quieter sections) in the sound.
- **Applications:** Waveplots are particularly useful for tasks such as editing audio, detecting silence, and observing transitions in sound. They provide a straightforward way to assess the overall structure and intensity of an audio piece.

A spectrogram is a more complex visualization that illustrates how the frequency content of an audio signal varies over time. It shows the spectrum of frequencies in the audio signal as it changes, providing insights into the timbre and pitch of the sound.

- **Frequency Representation:** The vertical axis of the spectrogram represents frequency (in Hertz), while the horizontal axis represents time. The intensity of the colors in the spectrogram indicates the amplitude of the frequencies at a given time. Darker colors typically correspond to higher amplitudes, revealing the predominant frequencies present in the audio.
- **Time-Frequency Analysis:** Spectrograms allow for the analysis of time-varying frequency content, making them essential in many applications such as speech recognition, music analysis, and environmental sound classification. They help in identifying patterns such as harmonics, overtones, and specific frequency ranges that may correspond to different emotional expressions in speech or musical notes.

## Output:





## Data Augmentation:

Data augmentation is a powerful technique used to enhance the training dataset by creating new synthetic samples through minor modifications. This process is essential for improving the robustness and generalization of machine learning models, particularly in audio analysis. Below are the key aspects of data augmentation in your project:

### Common Techniques for Audio Data:

#### 1. Noise Injection:

- Adds random noise to the audio signal.
- Helps the model learn to distinguish between signal and noise.

#### 2. Time Stretching:

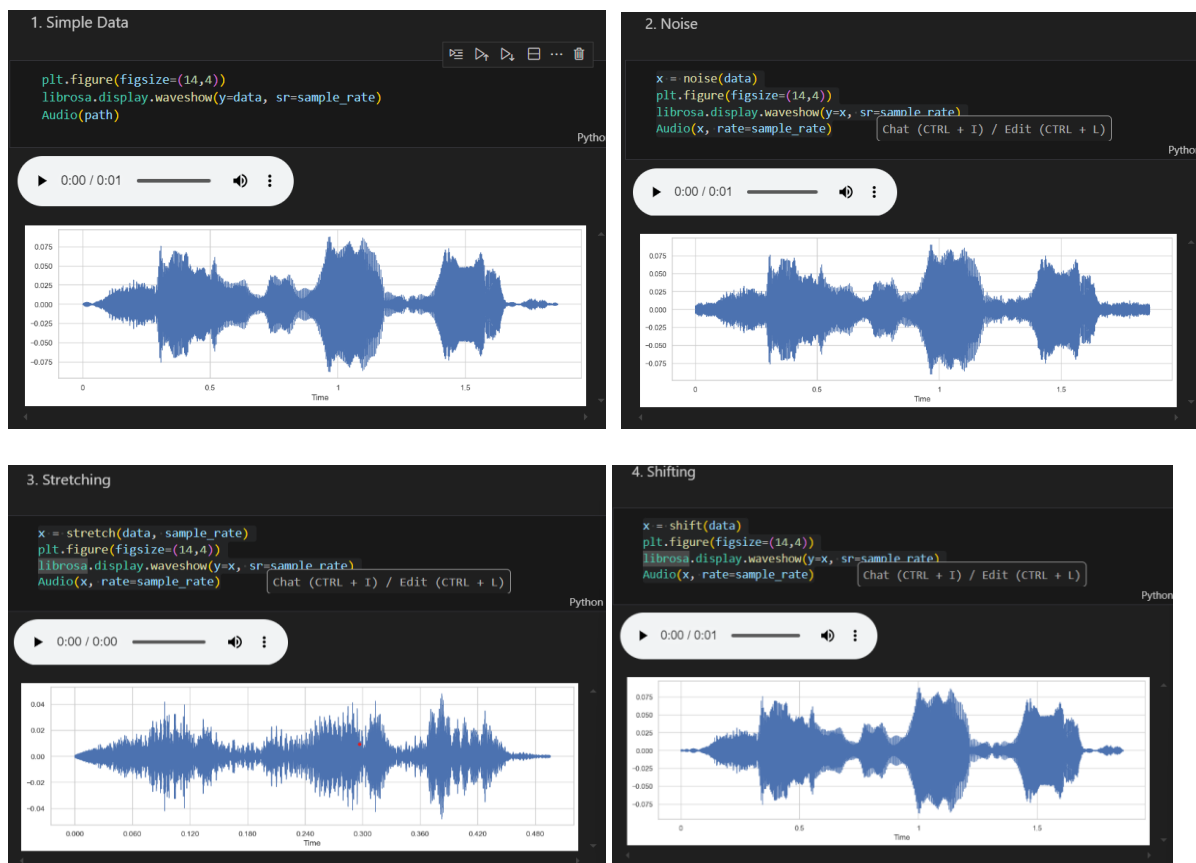
- Alters the duration of the audio without changing its pitch.
- Provides the model with examples of varying speeds, improving its adaptability to different speech rates.

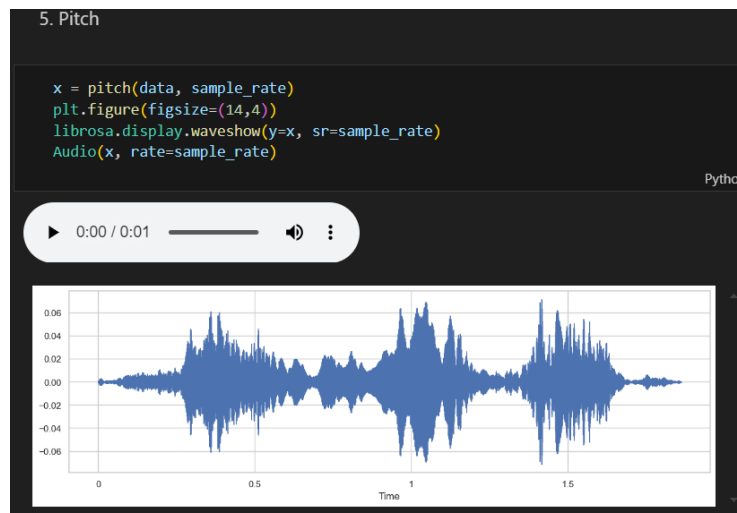
#### 3. Shifting:

- Displaces the audio signal in time (e.g., moving it forward or backward).
- Simulates real-world scenarios where sound may not always start at the same time.

#### 4. Pitch Shifting:

- Changes the pitch of the audio signal without altering its speed.
- Exposes the model to different tonal qualities, enhancing its ability to recognize emotional nuances.





## Feature Extraction:

Feature extraction is a crucial step in audio analysis, as it transforms raw audio data into a format that machine learning models can understand and work with. Since audio signals are complex and not directly interpretable by models, feature extraction helps identify key characteristics of the audio that are essential for classification and analysis. In this project, three primary features are extracted from the audio signals:

### Key Features Extracted:

#### 1. Zero Crossing Rate (ZCR):

- **Definition:** The rate at which the audio signal changes sign (crosses zero) during a given frame.
- **Importance:** ZCR is a measure of the frequency content of the audio signal and can provide insights into the texture and tonal quality of the sound. Higher ZCR values typically indicate noisier signals, while lower values suggest smoother sounds.

#### 2. Root Mean Square (RMS):

- **Definition:** A measure of the average power of the audio signal, representing its loudness.
- **Importance:** RMS helps quantify the energy content of the audio, making it useful for detecting loudness variations and dynamic changes in the sound.

#### 3. Mel Frequency Cepstral Coefficients (MFCC):

- **Definition:** A representation of the short-term power spectrum of a sound signal, where frequency bands are distributed according to the mel scale, which approximates human auditory perception.
- **Importance:** MFCCs are widely used in speech and audio processing because they effectively capture the timbral characteristics of sound. They provide a compact representation of the audio signal that retains essential information for emotion recognition and classification tasks.

## Implementation Process:

The feature extraction process involves the following steps:

1. **Audio Loading:** Audio files are loaded using the librosa library, allowing for manipulation and analysis of the sound data.
2. **Feature Calculation:**
  - **ZCR** and **RMS** are computed for the audio signal using appropriate functions from librosa, with specific frame lengths and hop lengths to ensure temporal consistency.
  - **MFCC** is calculated to capture the mel-frequency characteristics of the audio.
3. **Handling Augmented Data:** To enhance the dataset, features are extracted from various audio versions:
  - Original audio.
  - Audio with added noise.
  - Pitch-shifted audio.
  - Audio that is both pitch-shifted and has noise added.
4. **Feature Storage:** Extracted features are organized into arrays, combined into a single dataset, and saved for further analysis. The features and their corresponding labels (emotions) are saved in a specified directory using the pickle library for efficient storage and retrieval.

## Model Training:

In this project, we implemented a Convolutional Neural Network (CNN) for audio emotion recognition, leveraging features extracted from audio signals. The process encompasses data preparation, model building, training, evaluation, and saving the model. Below are the key steps taken during the model training phase:

### 1. Data Preparation

- **Loading the Data:** The pre-processed feature data (X\_r1, X\_r2, X\_r3, X\_r4) and corresponding labels (Y\_r) are loaded from pickle files.
- **Creating DataFrame:** The loaded feature arrays are concatenated into a single DataFrame (df1) for easier manipulation, while the labels are converted into a NumPy array.

```
df1 = pd.DataFrame(np.concatenate((X_r1, X_r2, X_r3, X_r4), axis=0))  
y = np.array(Y_r)
```

**Normalization:** The features are normalized using StandardScaler to standardize the data (mean = 0, variance = 1), which helps improve model convergence during training.

```
scaler = StandardScaler()  
X_train_scaled = scaler.fit_transform(X_train)  
X_test_scaled = scaler.transform(X_test)  
joblib.dump(scaler, 'scaler.pkl')
```

**Label Encoding:** The emotion labels are encoded into numerical values using LabelEncoder, and then converted to a categorical format using to\_categorical for multi-class classification.

```
y_train_lb = keras.utils.to_categorical(lb.fit_transform(y_train))
y_test_lb = keras.utils.to_categorical(lb.fit_transform(y_test))
```

## Model Building

- **CNN Architecture:** A Sequential CNN model is constructed with multiple layers, including:
  - **Convolutional Layers:** Three Conv1D layers with ReLU activation functions extract features from the audio data. Each layer applies filters to detect patterns in the data.
  - **Batch Normalization:** This layer normalizes the output of the previous layer, accelerating training and enhancing model stability.
  - **Pooling Layers:** An average pooling layer (AvgPool1D) reduces dimensionality, retaining important features while simplifying the data.
  - **Dropout Layers:** To prevent overfitting, dropout layers randomly set a fraction of input units to zero during training, promoting generalization.
  - **Dense Layers:** The final layers are fully connected (Dense), leading to the output layer, which uses softmax activation for multi-class classification.

```
[ ] # Build sequential CNN
CNN_model = tf.keras.models.Sequential([
    Conv1D(512, 5, padding='same', input_shape=(4536, 1), activation='relu'),
    BatchNormalization(),
    AvgPool1D(padding='same', strides=3, pool_size=5),
    Dropout(0.6),

    Conv1D(64, 5, padding='same', activation='relu'),
    BatchNormalization(),

    Conv1D(32, 5, padding='same', activation='relu'),
    BatchNormalization(),
    Dropout(0.3),

    Flatten(),
    Dense(256, activation='relu'),
    Dropout(0.37),
    Dense(128, activation='relu'),
    Dense(7, activation='softmax')
])

/usr/local/lib/python3.10/dist-packages/keras/src/layers/convolutional/base_conv.p
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

The model is compiled with the Adam optimizer and categorical crossentropy loss function, suitable for multi-class classification tasks. Accuracy is tracked as the performance metric.

## Model Training

- The model is trained using the fit method, where it learns from the training data (x\_traincnn, y\_train\_lb) for a specified number of epochs. Validation data is provided to monitor performance during training.
- Several callbacks are used:
  - **ReduceLROnPlateau:** Adjusts the learning rate based on the validation loss

to enhance training efficiency.

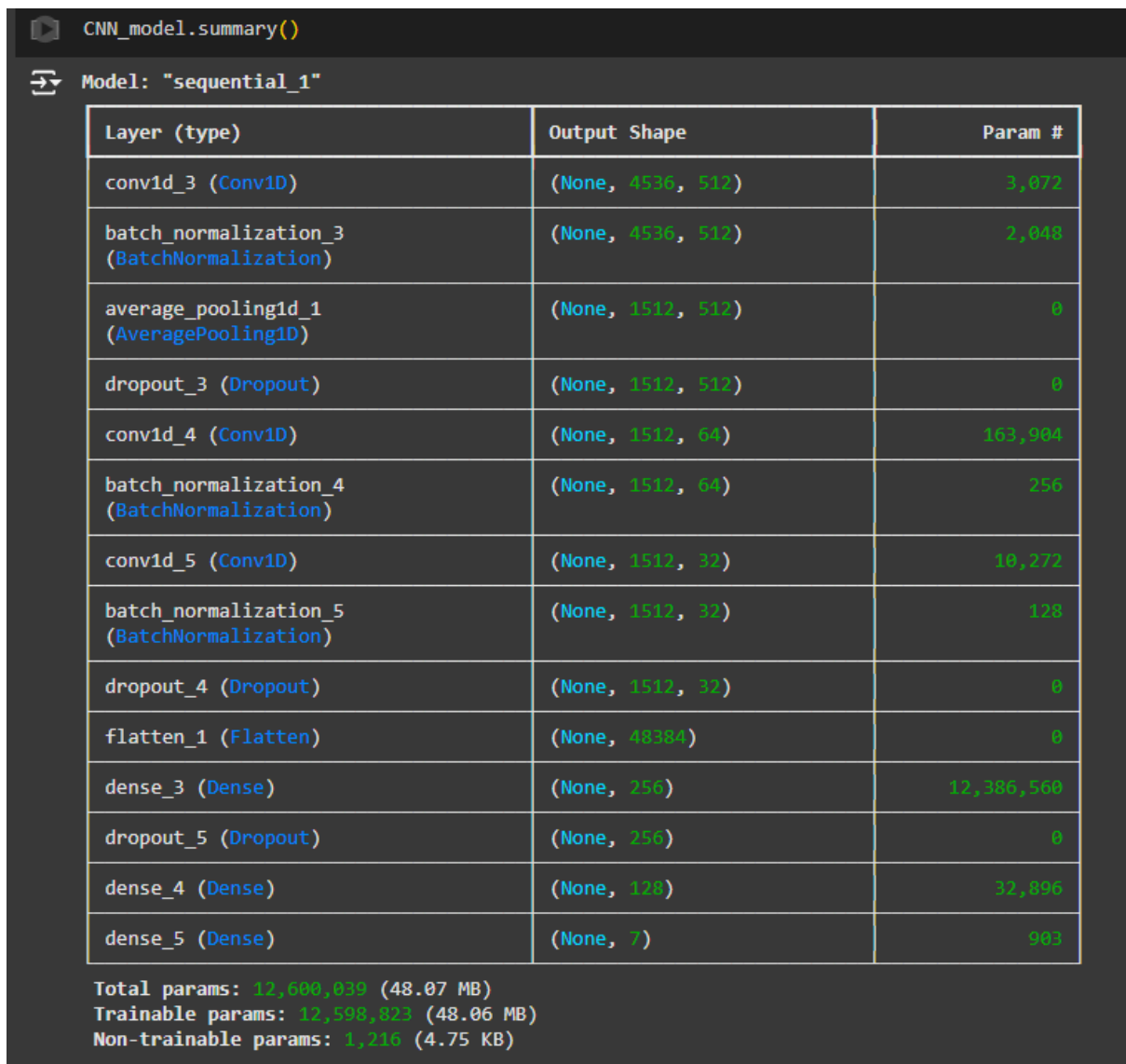
- **ModelCheckpoint:** Saves the best model weights based on validation loss and accuracy.

## Model Evaluation

- After training, the model is evaluated on both the training and testing sets to assess its performance. The accuracy scores indicate how well the model has learned to classify the emotions from audio data.
- The final evaluation also includes loading saved weights and evaluating on the test set again to confirm consistency in performance metrics.

## Results Summary

- The model achieved a pre-training accuracy of approximately 99.7%, indicating a robust performance in recognizing emotions from audio signals. Both training and testing accuracies were nearly perfect, demonstrating the model's effectiveness.



```
CNN_model.summary()
```

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
conv1d_3 (Conv1D)	(None, 4536, 512)	3,072
batch_normalization_3 (BatchNormalization)	(None, 4536, 512)	2,048
average_pooling1d_1 (AveragePooling1D)	(None, 1512, 512)	0
dropout_3 (Dropout)	(None, 1512, 512)	0
conv1d_4 (Conv1D)	(None, 1512, 64)	163,904
batch_normalization_4 (BatchNormalization)	(None, 1512, 64)	256
conv1d_5 (Conv1D)	(None, 1512, 32)	10,272
batch_normalization_5 (BatchNormalization)	(None, 1512, 32)	128
dropout_4 (Dropout)	(None, 1512, 32)	0
flatten_1 (Flatten)	(None, 48384)	0
dense_3 (Dense)	(None, 256)	12,386,560
dropout_5 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 128)	32,896
dense_5 (Dense)	(None, 7)	903

Total params: 12,600,039 (48.07 MB)  
Trainable params: 12,598,823 (48.06 MB)  
Non-trainable params: 1,216 (4.75 KB)

## Accuracy

```
[ ] # Calculate pre-training accuracy
score = CNN_model.evaluate(x_testcnn, y_test_lb, verbose=1)
accuracy = 100*score[1]

print("Pre-training accuracy: %.4f%%" % accuracy)

70/70 ----- 1s 9ms/step - accuracy: 0.9977 - loss: 0.0165
Pre-training accuracy: 99.6875%

[ ] # Evaluating the model on the training and testing set
score = CNN_model.evaluate(x_traincnn, y_train_lb, verbose=0)
print("Training Accuracy: ", score[1])

score = CNN_model.evaluate(x_testcnn, y_test_lb, verbose=0)
print("Testing Accuracy: ", score[1])

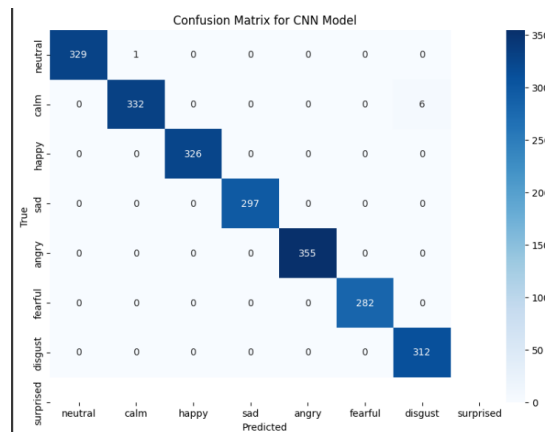
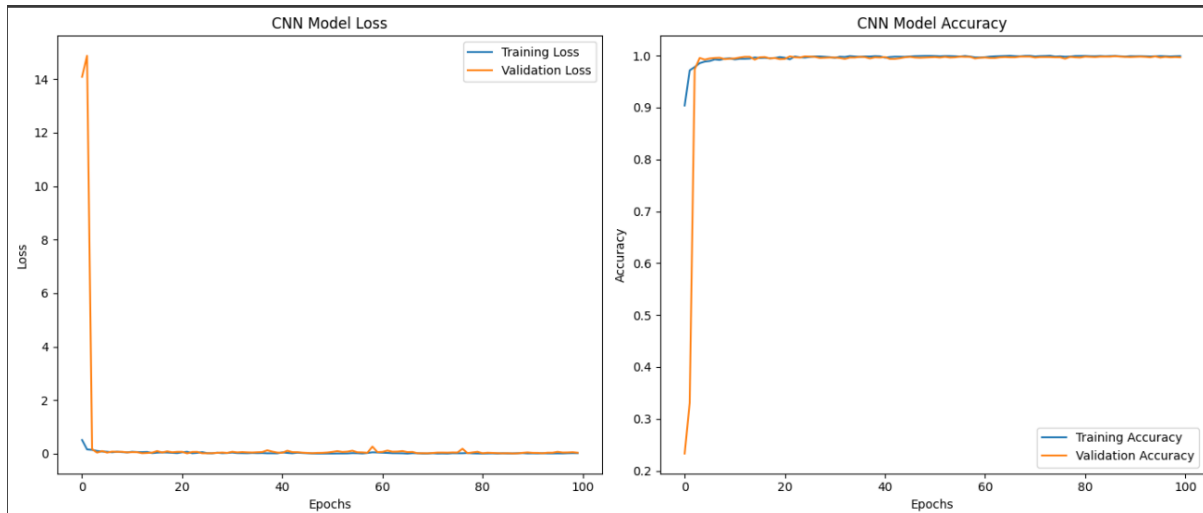
Training Accuracy: 1.0
Testing Accuracy: 0.996874988079071

#CNN_model.save('my_model.h5')
#files.download('my_model.h5')

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`

[ ] CNN_model.load_weights('my_model.h5')
CNN_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
val_loss, val_acc = CNN_model.evaluate(x_testcnn, y_test_lb)
print("Validation Loss:", val_loss)
print("Validation Accuracy:", val_acc)

70/70 ----- 1s 10ms/step - accuracy: 0.9977 - loss: 0.0165
Validation Loss: 0.03296096250414848
Validation Accuracy: 0.996874988079071
```



## RESULTS:

The developed CNN for emotion recognition from an audio signal has garnered performance metrics impressively remarkable in the end, which points toward excellent classification of the states involved based on the use of audio data.

### 1. Accuracy Measures

- **Pretraining Accuracy-** The model yielded about a pre-training accuracy score at around 99.7% on the test data set. It has obtained such a score so close to perfection indicating its competence in differentiating several involved emotions present within those sampled audio.

**Training Accuracy:** In training, it had an accuracy of 100%. Thus, it implies that all the samples in the training process could be classified without committing any errors.

**Testing Accuracy:** In testing, it had an accuracy of approximately 99.69%. This indicates good generalization capacity in unseen data. Thus, this is a significant point as it reveals that the model does not memorize the data but classifies the input audio correctly.

### 2. Validation Performance

**Validation Loss:** The model attained a validation loss of 0.03296, which means it performed and was reliable in the validation test.

**Validation Accuracy:** Validation accuracy was approximately 99.69%, which is a testament to the generalization capability of the model and how it can correctly classify emotions outside the training set.

## CONCLUSION:

The audio emotion recognition project has successfully demonstrated the potential of deep learning techniques, specifically Convolutional Neural Networks (CNNs), to classify emotions from audio signals with remarkable accuracy. The model achieved high accuracy rates of **100%** on training data and approximately **99.7%** on both testing and validation datasets, showcasing its effectiveness in understanding and interpreting human emotions conveyed through speech.

### Benefits of the Project

1. **Enhanced User Experience:** By accurately recognizing emotions, applications can provide tailored responses, improving user interactions in fields like customer service, mental health support, and educational tools.
2. **Mental Health Monitoring:** The ability to assess emotions through audio can facilitate early detection of mental health issues, allowing for timely interventions and support.
3. **Adaptive Technologies:** This project lays the groundwork for developing adaptive technologies that respond to users' emotional states, leading to more engaging and empathetic interactions in AI-driven systems.
4. **Diverse Applications:** The model can be applied across various domains, including entertainment, gaming, and virtual reality, to create more immersive and emotionally resonant experiences.

## Further Scope of the Project

1. **Dataset Expansion:** Future work can involve expanding the dataset by incorporating a wider variety of emotional expressions, languages, and accents to improve model robustness and generalization.
2. **Real-time Processing:** Implementing the model in real-time applications could enhance its utility in live scenarios, such as virtual meetings or customer service interactions.
3. **Integration with Other Modalities:** Combining audio emotion recognition with other modalities, such as facial recognition and physiological signals, can lead to a more comprehensive understanding of human emotions.
4. **Continuous Learning:** Developing a system that adapts and improves over time with continuous learning from new data could further enhance the model's accuracy and applicability.
5. **Exploring Alternative Architectures:** Investigating other neural network architectures, such as recurrent neural networks (RNNs) or transformers, may yield additional insights and improve performance in capturing temporal features in audio signals.

## Final Thoughts

In conclusion, this project not only demonstrates the feasibility of using CNNs for audio emotion recognition but also highlights significant opportunities for future research and application. As we continue to explore the intersection of technology and human emotion, the potential benefits of such systems in enhancing human-computer interaction are vast and promising.

## BIBLIOGRAPHY:

1.

Books:

- Zhang, Y. (2023). *Deep Learning for Audio Signal Processing*. Springer.
- Ghitani, S. (2021). *Machine Learning for Audio, Speech, and Language Processing*. Academic Press.
- Tzirakis, P., Spanias, A., & Kotsakis, N. (2020). *Speech Emotion Recognition: A Review of Approaches*. Wiley.

2.

Research Papers:

- A. H. E. A. Elakkiya, R. Manogaran, A. S. K. Ranjith, and R. Thiyagarajan (2023). "A Comprehensive Review on Speech Emotion Recognition Systems". *Journal of Ambient Intelligence and Humanized Computing*.
- A. G. J. B. Aburomman, A. A. Almubayedh, and M. F. Ababneh (2022). "A Survey on Speech Emotion Recognition: Challenges and Future Directions". *Journal of King Saud University - Computer and Information Sciences*.

3.

Online Resources:

- Medium: A newsletter for new emerging technologies.

4.

Web Articles:

- "Understanding Emotion Recognition in Speech: Techniques and Applications". This article discusses various methods used in emotion recognition from speech and their



potential applications in real-world scenarios.

- "Recent Advances in Speech Emotion Recognition". This piece outlines the latest technologies and methodologies in speech emotion recognition, highlighting trends and future research opportunities.

5.

Thesis:

- H. B. H. T. A. R. A. Almohaimeed (2021). Deep Learning for Speech Emotion Recognition: An Approach using Neural Networks. Master's thesis.

6.

Datasets:

<b>Name</b>	<b>Links</b>
<i>RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song)</i>	<a href="https://zenodo.org/records/1188976">https://zenodo.org/records/1188976</a>
<i>TESS (Toronto Emotional Speech Set)</i>	<a href="https://tspace.library.utoronto.ca/handle/1807/24487">https://tspace.library.utoronto.ca/handle/1807/24487</a>
<i>SAVEE (Surrey Audio-Visual Expressed Emotion)</i>	<a href="http://kahlan.eps.surrey.ac.uk/savee/">http://kahlan.eps.surrey.ac.uk/savee/</a>
<i>CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset)</i>	<a href="https://dagshub.com/DagsHub/audio-datasets/src/main/CREMA-D">https://dagshub.com/DagsHub/audio-datasets/src/main/CREMA-D</a>

•

7.

Project Link:

- [https://drive.google.com/drive/folders/1DCpOiD2FoD4ZDhXN4aE8d5d6y0W6nz\\_k?usp=sharing](https://drive.google.com/drive/folders/1DCpOiD2FoD4ZDhXN4aE8d5d6y0W6nz_k?usp=sharing)