



# Speech Emotion Recognition

25.07.2024

---

## Team Members

Sai Pranay Deep J

Rithvik B

Tejaswini P

Saketh B

## Introduction

The aim of the project is to develop a robust system for emotion detection from speech, utilizing advanced machine learning techniques. The system should accurately classify emotions, including happiness, sadness, anger, and more, based on the acoustic features and patterns present in the speech signal.

## Prerequisites

1. Basic knowledge of Python
2. Python libraries like NumPy, Pandas, Librosa.
3. CNN, LSTM, MLP

## Procedure

The model was made by using CNN and MLP Classifier trained on four different datasets namely RAVDESS, SAVEE, TESS, CREMA available on Kaggle. The entire project is divided into three parts: Feature Extraction, Training the model, Predicting. In the first part we have extracted the main features of audio like MEL, CHROMA, MFCC. Then these features are trained using MLP Classifier. At last, predicting the emotions of one's own speeches.

## Features Extraction

First We have loaded the datasets and taken the paths of each audio file. In each dataset the emotion is embedded in the path. So we have extracted the emotion of each audio file from its path. Then we have created functions for getting features for each audio file. We have augmented the data to make it more generalized for all kinds of audio inputs.

Coming to the features MFCC stands for Mel Frequency Cepstral Coefficients. It is a widely used feature extraction technique in the field of speech and audio signal processing. MFCCs are used to represent the spectral characteristics of an audio signal, particularly in applications such as speech recognition, speaker identification, and audio classification. A high-pass filter is applied to emphasize higher frequencies in the signal. The signal is divided into short overlapping frames to capture the temporal characteristics of the audio. A window function (e.g., Hamming window) is applied to each frame to reduce spectral leakage. The Fourier Transform is applied to convert the audio signal from the time domain to the frequency domain. A set of triangular filters spaced on the Mel scale is used to filter the power spectrum of the signal. The logarithm of the filterbank energies is taken to

convert the amplitude to decibel scale. Finally, DCT is applied to decorrelate the filterbank energies and obtain the MFCC coefficients.

Chroma feature is a representation of the twelve different pitch classes in the musical octave. It is a powerful tool for analyzing harmony and chords in audio signals. The chroma feature is particularly useful in music information retrieval, automatic chord recognition, and genre classification tasks.

The audio signal is first transformed into the frequency domain using the Short-Time Fourier Transform (STFT).

The energy in each frequency band is summed across octaves, and the twelve pitch classes (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) are computed.

The chroma values are often normalized to have a constant energy sum, which is useful for comparing different audio signals.

The Mel scale is a perceptual scale of pitches based on the human ear's perception of different frequencies. It is widely used in audio signal processing to better model human auditory perception. The Mel scale is non-linear and more closely represents how humans perceive pitch differences. The conversion from linear frequency to Mel frequency is achieved using the following formula:

$$\text{Mel}(f) = 2595 * \log_{10}(1 + f / 700)$$


where  $f$  is the frequency in Hertz.

## Training the Model

We have used MLP and CNN models.

The MLP Classifier is a type of feedforward artificial neural network used for classification tasks. It is a variant of the multilayer perceptron (MLP) model, which consists of one or more hidden layers of interconnected neurons (nodes). Each node in a layer is connected to every node in the previous and subsequent layers, forming a directed graph-like structure. Despite their effectiveness, MLP classifiers may suffer from overfitting if the model is too complex or the training dataset is limited. Regularization techniques and proper hyperparameter tuning are often employed to mitigate this issue. Overall, the MLP Classifier is a popular choice for classification problems due to its ability to handle complex data and learn non-linear relationships, making it a foundational component of many deep learning models.

Convolutional Neural Network (CNN) is a specialized type of deep learning model designed for processing and analyzing visual data, such as images and videos. It has become the backbone of various computer vision tasks due to its ability to automatically learn hierarchical patterns and features from raw pixel data. It has Convolutional Layers, The core building block of CNNs is the convolutional layer. It applies convolutional filters (also



known as kernels) to the input data, scanning across the image to extract local features. Each filter learns to detect specific patterns, such as edges, textures, or more complex structures. And then Pooling Layers, After convolution, pooling layers are used to downsample the spatial dimensions of the feature maps, reducing the computational burden and preserving important information. Max pooling and average pooling are common techniques used for this purpose. After several convolutional and pooling layers, the output is flattened into a 1D vector to be fed into fully connected layers. After experimenting with multiple models, we achieved a maximum accuracy of approximately 73% using the MLP classifier model. However, we observed that as the validation accuracy increased, the model failed to generalize well with random or unfamiliar audio files (unless the speech was similar to the data we trained it on).

### Predictions:

At last using the saved pickles, audio file is uploaded and the audio file is split into multiple segments of three seconds as the dataset we have trained is of four seconds. Each three seconds audio was predicted by the model and outputs the maximum emotion throughout the audio. Nevertheless, the final selected model performed well with our own speeches. Occasionally, the model may become confused when multiple emotions are present in the audio file. We extensively tested it with our own speeches and some audio files from social media platforms like YouTube, and it produced very accurate predictions.

### Demo:

[https://drive.google.com/drive/folders/10M92FM0e1Sk8NcxkYW2Mz1L5gse\\_ZRB0?usp=sharing](https://drive.google.com/drive/folders/10M92FM0e1Sk8NcxkYW2Mz1L5gse_ZRB0?usp=sharing)

### Possible Improvements

Although we successfully predicted six basic emotions in this project, there is room for expansion to include additional emotions. Furthermore, we envision enhancing the model to predict the emotions and their corresponding starting and ending times in audio files that contain multiple emotions

### Conclusion

Apart from the improvements mentioned above, our team is delighted to present the conclusion of this **ML-based project, "Speech Emotion Recognition,"** developed under the auspices of **IITISoc'23**. We are extremely grateful for the opportunity to showcase our



curiosity and interest in ML. Throughout this process, we have gained valuable insights and deepened our fascination with ML.