Machine Learning Model for Speech Emotion Recognition

Vandana Bhatia, Om Pathak, Rounak Ranjan, Himanshu Sogarwal and Sarthak Janjuha
Computer Science Engineering, Netaji Subhas University of Technology,
Dwarka, New Delhi, India - 110075

Abstract— **As human beings, speech is amongst the most natural ways to express ourselves. We depend so much on it that we recognize its importance when resorting to other communication forms like emails and text messages. As we use more voice-controlled gadgets, emotion recognition will be an integral part of these devices in the near future. Emotion detection is a challenging task, because emotions are subjective. There is no common consensus on how to measure or categorize them. This paper proposes a model to recognize emotions from speech based on a multilayer perceptron network and making use of the functionalities of MFCC, MEL, and Chroma to recognize the emotion of the audio input. Experiments indicate that the model proposed in this paper can excellently predict speech emotion.**

Keywords - **MFCC, Chroma, MEL,MLP classifier**

I. INTRODUCTION

As emotions play a vital role in communication, the detection and analysis of the same is of vital importance in today's digital world of remote communication. For example: Voice-based assistants like Alexa and Siri can understand and respond to us, but it cannot give an adequate response based on our emotions. Voice-based gender identification has also been a challenging task for voice and audio analysts.

Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and affective states from speech. The human voice should be converted from the analogue to the digital form to extract useful features and then to construct classification models. Such a system can find use in a wide variety of application areas like interactive voice based-assistant or caller-agent conversation analysis.

II. LITERATURE

Speech structures refers to the variability of speech pronunciation in time. When people express different emotions, the timing of the speech is different. Mainly in two aspects, one is the length of continuous pronunciation time, the other is the average rate of pronunciation. One is the length of continuous pronunciation time and the next one is the average rate of pronunciation.

The study of Z.Li [10] showed that the different pronunciations of emotions differ in the length of pronunciation and the speed of pronunciation. Compared with the length of the quiet call period, the call time for joy, anger and surprise is greatly shortened. But compared to the quiet call time, the length of the sad call is longer. Compared with the silent call rate, the sad pronunciation rate is slower, while the excitement, anger, and surprise levels of surprise are relatively fast.

III. IMPLEMENTATION

Data Set for Speech Recognition : The dataset provided by RAVDESS is used for this model because of its compact size and variance provided in the data itself. Speech audio-only files (16bit, 48kHz .wav) from the RAVDESS have been used.

The RAVDESS contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech emotions include 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. The RAVDESS dataset contains 1440 files: 60 trials per actor x 24 actors = 1440.

The main library that is used is Librosa. Apart from that SoundFile and Pyaudio have also been used. Librosa is a Python library for analyzing audio and music.

Pre-processing on Dataset : Data pre-processing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for building and training Machine Learning models.

There are 7 steps in data pre-processing
1. Acquire the dataset
2. Import all the crucial libraries
3. Import the dataset
4. Identifying and handling the missing values
5. Encoding the categorical data
6. Splitting the dataset for testing & training
7. Feature scaling

Feature Selection : Audio features can be broadly classified into two categories, namely time-domain features and frequency-domain features. Time-domain features include the short-term energy of the signal, zero crossing rate, maximum amplitude, minimum energy, entropy of energy. These features are very easy to extract and provide a simpler way to analyze audio signals. Under limited data, frequency domain features reveal deeper patterns in the audio signal, which can potentially help us identify the underlying emotion of the signal. Frequency-domain features include mel spectrograms, Mel-Frequency Cepstral Coefficients (MFCCs) and chroma coefficients. Librosa module has been used for their extraction**.**

Mel Spectrogram : A spectrogram represents the spectrum of frequencies over time representation of an audio signal. Different emotions exhibit different patterns in the energy spectrum. Mel Spectrograms are spectrograms that visualize sounds on the Mel scale as opposed to the frequency domain**.** The logarithmic form of mel-spectrogram helps understand emotions better because humans perceive sound in logarithmic scale.

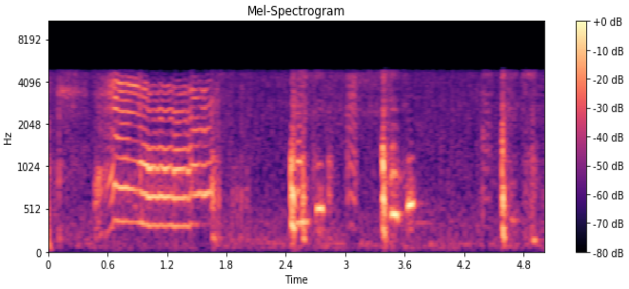
Mel scale - The Mel Scale is a logarithmic transformation of a signal's frequency.

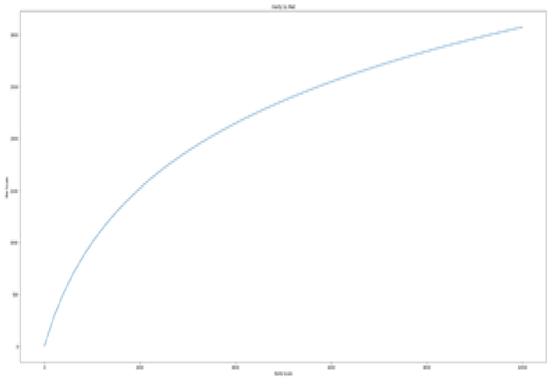
For example, most human beings can easily tell the difference between a 100 Hz and 200 Hz sound. However, by that same token, we should assume that we can tell the difference between 1000 and 1100 Hz, right? Wrong.

It is actually much harder for humans to be able to differentiate between higher frequencies, and easier for lower frequencies. So, even though the distance between the two sets of sounds are the same, our perception of the distance is not. This is what makes the Mel Scale fundamental in Machine Learning applications to audio, as it mimics our own perception of sound.

Transformation from the Hertz scale to the Mel scale.

$$m = 1127 . \ln(1+f/700)$$



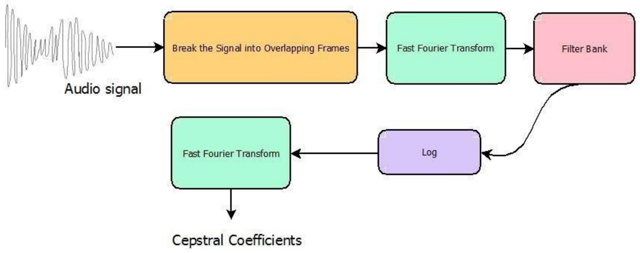


y-axis – mel scale x-axis- hertz scale

MFCC(Mel Frequency Cepstral Coefficients) : MFCC represents the short-term power spectrum of a sound. The basic procedure to develop MFCCs is the following:

1. Convert from Hertz to Mel Scale
2. Take logarithm of Mel representation of audio
3. Take logarithmic magnitude and use Discrete Cosine Transformation
4. This result creates a spectrum over Mel frequencies as opposed to time, thus creating MFCCs

Mel-Frequency Cepstrum is a representation of the short-term power spectrum of a sound by transforming the audio signal through a series of steps to mimic the human cochlea. The Mel scale is important because it better approximates human-based perception of sound as opposed to linear scales. In filter-source theory, "the source is the vocal cords and the filter represents the vocal tract." The length and shape of the vocal tract determine how sound is outputted from a human and the cepstrum ("spectrum of the log of the spectrum") can describe the filter, i.e., represent sound in a structured manner. Mel-Frequency Cepstral Coefficients (MFCC) are coefficients which capture the envelope of the short time power spectrum.



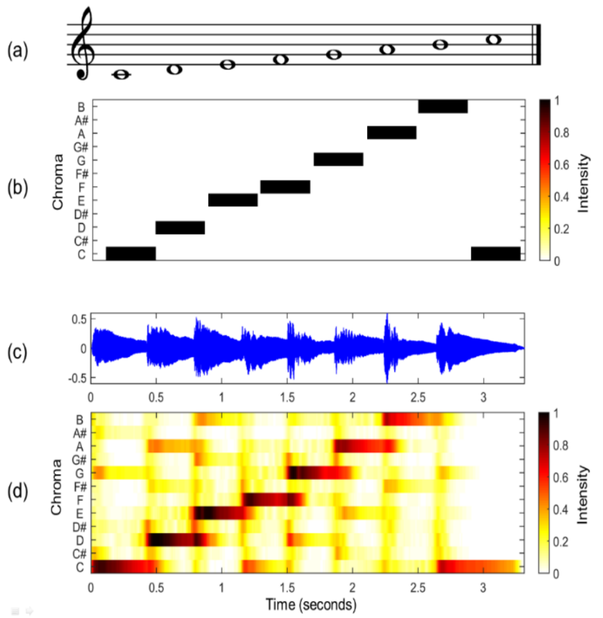
MFCC flow chart

Chroma : Chroma features are powerful representatives of the sound of music where the whole screen is displayed in 12 bins representing 12 different semitones (or Chroma) octave music. Important properties of chroma elements is that they capture harmonic and musical features, while being strong in transformation of timbre and instrumentation.

The two main features of chroma are listed below :

Chroma vector: A 12-element feature vector indicating how much energy of each pitch class is present in the signal in a standard chromatic scale.

Chroma Deviation : The normal deviation of the 12 coefficients of chroma.



Chroma figure

Features Extraction : Function defined to extract the MFCC, Chroma, and Mel features from a sound file. This function takes 4 parameters- the file name and three Boolean parameters for the three features. Opened the sound file with the Sound File library. Read from it and labeled it **X**. Also, acquired the sample rate. If chroma is True, got the **Short-Time Fourier Transform** of **X**.

The result is an empty numpy array. Now, for each feature of the three, if it existed, made a call to the corresponding function from librosa.feature and got the mean value. Called the function hstack() from numpy with result and the feature value, and stored this in the result. Then, returned the result.

ML Model : To store the emotions present in the RAVDESS dataset we create a dictionary of labels and their respective emotions to use when training the machine learning model. We also create a list of emotions that we want to focus on in this project. It's hard to do a prediction using all emotions, because the speech may sound in more than one emotion simultaneously, and that will affect our prediction scores. That's why we chose three primary emotions, which are happy, sad, and angry.

Loading Data : In this step, we are going to define a function to load our dataset. The function that takes in the relative size of the test set as parameter. x and y are empty lists. We'll use the **glob()** function from the **glob module** to get all the pathnames for the sound files in our dataset.

So, for each such path, get the base name of the file, the emotion by splitting the name around '-' and extracting the third value. Using our **emotions dictionary**, this number is turned into an emotion, and our function checks whether this emotion is in our list of **observed emotions,** if not, it continues to the next file. It makes a call to **extract features** and stores what is returned in 'feature'. Then, it appends the feature to x and the emotion to y. You can think of features as our input (x) and the labeled emotion as an output (y). This is a well-known machine learning model, also known as Supervised Learning.

Train-Test Split : After loading the data, we are going to split the labeled dataset using the train_test_split() function. It is a well-known splitting function by the Scikit-learn module. It divides the dataset into four chunks. We can define how much of the dataset we want to use for training and how much for testing. You can adjust these values to see how it affects the prediction. There is no one size fits all rule; it usually depends on the dataset. But in most cases, the 0.25 test size is applied. This means 3/4 of the dataset is used for training and 1/4 for testing.

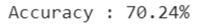
Classification Task : For our speech emotion recognition system we will be using the MLPClassifier Prediction Model . MLP stands for Multi Layer Perceptron. It uses an internal neural network model to optimize the log-loss function using Limited memory BFGS or stochastic gradient descent. This is a feedforward ANN model.

We start by running the loading_audio_data() function. This function will return four lists. That's we are going to use four different variables for each list. Then we fit our model. After our model is fit, we move to the prediction step. We are going to assign the predicted values into a new variable called y_pred. This way, we can calculate the accuracy score of the prediction. The accuracy score function checks how many of the predicted values are matching with the original label data.

To calculate the accuracy of our model, we'll call up the **accuracy_score()** function we imported from sklearn. We will also use the **confusion matrix** for a better understanding of our model predictions.

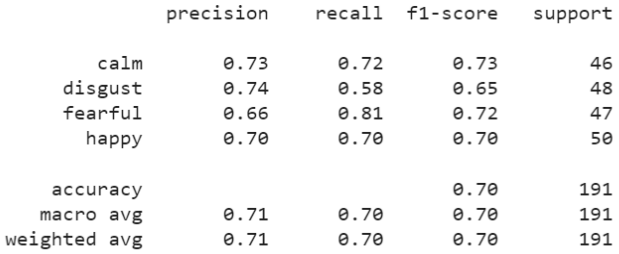
IV. PERFORMANCE ANALYSIS

Accuracy of the model :



```
Accuracy : 70.24%
```

Classification report of the model :

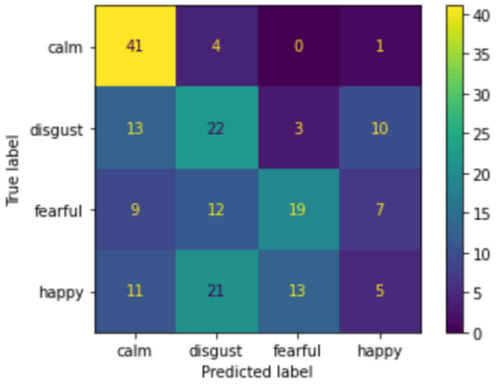


```
              precision    recall  f1-score   support

        calm       0.73      0.72      0.73        46
     disgust       0.74      0.58      0.65        48
     fearful       0.66      0.81      0.72        47
       happy       0.70      0.70      0.70        50

    accuracy                           0.70       191
   macro avg       0.71      0.70      0.70       191
weighted avg       0.71      0.70      0.70       191
```

Confusion matrix of the model :



V. CONCLUSION

In this paper, we have extracted speech features that can be used for recognition of emotional speech states, and implemented a model that confirmed the test accuracy to be approximately 70%. In future, we would like to use more features from the dataset to improve the speech recognition accuracy of this model.

REFERENCES

[1] Kyong Hee Lee; Hyun Kyun Choi; Byung Tae Jang; Do Hyun Kim," A Study on Speech Emotion Recognition Using a Deep Neural Network", 2019 International Conference on Information and Communication Technology Convergence (ICTC), 16-18 Oct. 2019

[2] Z. Li, "A study on emotional feature analysis and recognition in speech signal," Journal of China Institute of Communications, vol. 21, no. 10, pp. 18–24, 2000.

[3] scikit-learn Documentation

[4]https://www.kaggle.com/uwrfkaggler/ravdess-emotional-speech-audio - Dataset Source