# SPEECH EMOTION RECOGNITION

***A Project Report submitted in partial fulfillment of the requirements for the award of the degree of***

## Bachelor of Technology

**In**

## *Computer Science and Engineering*

By


Madhav Garg
181500356

Megha Kansal
181500382

Rishabh Tiwari
181500566

Mansi Verma
181500372


Under the Guidance of

Dr. Neeraj Gupta

Prof. Sandeep Rathore


Department of Computer Engineering & Applications

**Institute of Engineering & Technology**

**GLA University**

**Mathura- 281406, INDIA**


**Nov 2020**

**Department of Computer Engineering and Applications GLA University, 17 km. Stone NH#2, Mathura-Delhi Road, Chaumuha, Mathura – 281406 U.P (India)**

# Declaration

We hereby declare that the work which is being presented in the B.Tech. Project **"Speech Emotion Recognition"**, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the supervision of **Name & Designation of supervisor(s).**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign _____

Name of Candidate: Madhav Garg

University Roll No.: 181500356

Sign _____

Name of Candidate: Megha Kansal

University Roll No.: 181500382

Sign _____

Name of Candidate: Rishabh Tiwari

University Roll No.: 181500566

Sign _____

Name of Candidate: Mansi Verma

University Roll No.:181500372

# Certificate

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

_____

**Supervisor**

Mr. Neeraj Gupta

Designation of Supervisor

_____ _____

**Project Coordinator** **Program Coordinator**

(Mr. Mayank Srivastava) (Dr Anant Ram)

Date: Nov 21, 2020

# ACKNOWLEDGEMENT

We would like to thank all those people who graciously helped us by sharing their knowledge and valuable time. We are thankful to all those learning platforms out there on the Internet like stackoverflow, medium and many others with the people helping the community and believing in sharing knowledge. We express our sincere gratitude to our professors who keenly observed the progress of the project and guided us correctly for learning and completion in this research.

Sign _____

Name of Candidate: Madhav Garg

University Roll No.: 181500356

Sign _____

Name of Candidate: Megha Kansal

University Roll No.: 181500382

Sign _____

Name of Candidate: Rishabh Tiwari

University Roll No.: 181500566

Sign _____

Name of Candidate: Mansi Verma

University Roll No.:181500372

# ABSTRACT

*This project on **"Speech Emotion Recognition"** is based on Machine Learning. In past years a lot of research has been done on SER and as a result many applications of it have been found in medical science, robotics and other fields. One of the objectives of this research is to decrease the gap between humans and machines taking emotions into account. In this project 4 different emotions (calm, happy, sad, angry,) have been recognized and MFCC, Chroma and Mel is used for feature extraction. There are 180 extracted features from the audio files of RAVDESS dataset. It is a multi-class classification task but cannot be done with Logistic Regression (the simplest classification algorithm) as the classes and features are too much for it. There are five classifiers used namely KNN, SVM and MLPClassifier, Random Forest and Decision Tree for training the model. For better accuracy PCA (Principal Component Analysis) is also applied on extracted features before training the model. With the KNN (K-Nearest Neighbors) the model gives an accuracy of 73.38% having k=5. And by using SVM (Support Vector Machines) as a classifying model with the rbf kernel, it gives an accuracy of 76.62%. The third is Random Forest which gives an accuracy of 80.5%. And the highest accuracy is achieved with MLPClassifier having 83.12%.*

# CONTENTS

# List of Figures



# List of Tables

# CHAPTER-1: INTRODUCTION

## 1.1 OVERVIEW AND MOTIVATION

There are two parts when a person interacts with others, one is the verbal, which is used to communicate the message between the two and the other part is non-verbal communications which passes the feelings and state of speaking of the speaker. Now both parts together pass the whole conversation. The non-verbal part is as important as the verbal one. Recognizing the person's emotion by his sound and giving machines the ability to understand the emotions of a person and act according to the command plus the emotions with it is the motivation behind this project.

## 1.2 OBJECTIVE

The main aim of this project is to recognize the emotions (only 4) of a person with his sound.

## 1.3 ISSUES AND CHALLENGES

- Determining the correct emotion of a person in his speech is sometimes difficult for a human being also, especially when the speaker is suppressing his emotions.
- Identifying the features that impact the emotions of a person for feature extraction from a raw audio file.

# CHAPTER-2: LITERATURE REVIEW

In past years, different techniques were used for speech emotion recognition. Various classifiers used for the classification of speech features such as Gaussian Matrix Model (GMM), Support Vector Machine (SVM), and k-nearest neighbors (KNN), Artificial Neural Network (ANN), etc.

To recognize speech emotion, a number of classifiers have been proposed by various researchers---

1. Renjith S, et.al, (2017), have worked on Telugu and Tamil languages to detect emotions, happiness, sadness and anger using speech recordings. In their work they have pre-processed to separate disturbances from speech waveforms and raw speech signals. They have extracted Hurst and Linear Predictive Cepstral Coefficients (LPCC) features and then classification is done on the basis of statistical parameters obtained from these features. Both KNN and ANN are used to identify the reactive emotions and then accuracy, precision and recall parameter is used to compare their performance for both features individually and in combination. As compared to LPCC when use of Hurst gives better results when tested for individual features in terms of recall, precision and accuracy.

2. Atreyee Khan, et.al, (2017), has used Naive Bayes classifier along with both spectral and prosodic features for emotion detection. As spectral features a Mel-Frequency Cepstral Coefficients (MFCC) has been used and pitch is used as a prosodic feature. Naïve Bayes Classifier is used to perform classification and they have considered seven emotional classes to develop both gender independent and dependent systems. Berlin Emo-db popular speech database speech samples are used to test accuracy of the system after performing classification. The Energy spectrum of discrete cosine transform (DCT) is used to calculate MFCC in which they have considered only 1-14 coefficients of DCT and rest is discarded.

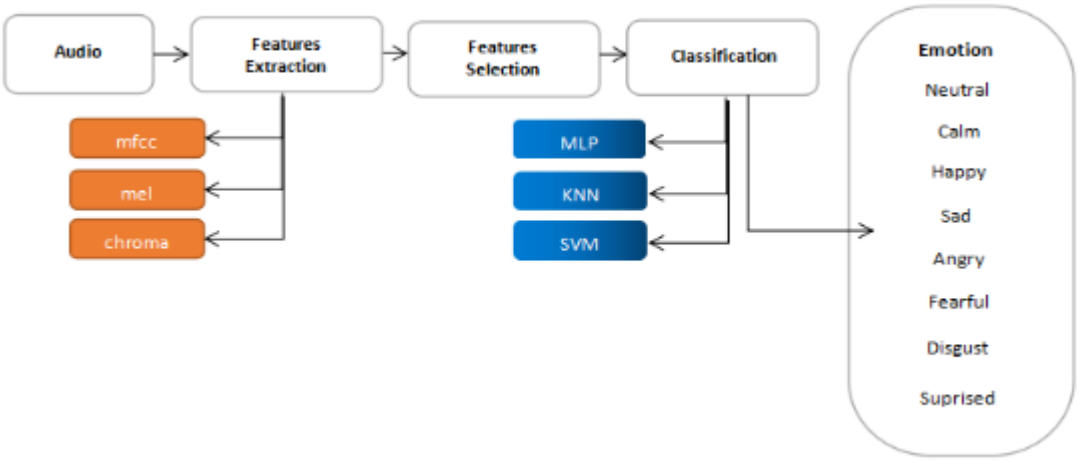
3. Ashwini Rajasekhar, et.al, (2018), have distinguished the sample using SVM and speaker utterance is detected using MFCC. In the end SVM classifier differentiates between fear, anger, sadness, happiness and updates the database accordingly. The proposed approach is evaluated for a combination of features in terms of accuracy that shows a good result for speaker independent cases as compared to individual features.

4. Li Zheng, et.al, (2018), have proposed a random forest and CNN based new network model. From normalized spectrogram a speech emotion features are extracted using CNN and then speech emotion features are classified using RF classification algorithm. From results it has been predicted that as compared to traditional CNN model use of CNN-RF model gives improved results.

**TABLE 1: COMPARISON TABLE OF DIFFERENT CLASSIFIER**

| S. No | Author name | Classifier | Database | Feature Extraction Technique | Accuracy (%) |
|---|---|---|---|---|---|
| 1 | Renjith S, et.al, (2017) | kNN and ANN | Amritaemo | LPCC | 61.29% |
| 2 | Atreyee Khan, et.al, (2017) | Naive Bayes | Berlin Emo-db | MFCC | 81% |
| 3 | Ashwini Rajasekhar, et.al, (2018) | SVM | They have used computerized voice dataset | MFCC | 87% |
| 4 | Li Zheng, et.al, (2018) | (CNN-RF) | RECOLA natural emotion database | CNN | 84.60% |

# CHAPTER-3: PROPOSED WORK




**BLOCK DIAGRAM OF SPEECH EMOTION RECOGNITION**

## 3.1 PROPOSED ALGORITHM

### 3.1.1 AUDIO

In this project, we are using the RAVDESS dataset from Kaggle. This dataset includes 24 actors, 12 male actors and 12 female actors and each actor has 60 trials that contain emotion (calm, happy, sad, angry, fearful, surprise, and disgust emotions). Every emotion has two levels of intensity, strong and normal.

### 3.1.2 FEATURES EXTRACTION

The speech or audio contains numerous features that help to uniquely identify each emotion. In this project, we have used Mel-frequency cepstrum coefficients (MFCC), Mel Spectrogram Frequency (MEL), chroma to extract features.

We have defined a function extract_feature to extract features from audio. It takes four arguments (MFCC, Mel, chroma, and the filename).

MFCC: - It is the representation of the short-term power spectrum of sound.

Chroma:-It is closely related to twelve different pitch classes.

### 3.1.3 FEATURES SELECTION

There are many features but there must be some features that would not affect the target, therefore, we need to reduce the number of features used to characterize a dataset to improve the performance of the algorithm on a given task. But on selecting a number of features only, the model performs worse on different classifiers. So, we have used all 180 features for training.

Now for increasing the accuracy of particular models, Principal Component Analysis (PCA) is applied which produces 178 new features from the dataset.

### 3.1.4 CLASSIFICATION

In this project, we are using five different classifiers to differentiate the performance as every algorithm has its pros and cons.

MLP: - Multilayer Perceptron Classifier

SVM: - Support Vector Machine is based on finding a hyper plane that separates the features. Here we are using RBF kernel (Radial Basis Function) whose value depends on the distance from some point.

KNN: - It is a lazy learner algorithm so it stores the dataset and performs an action at the time of classification. We have to use the right value of k.

Decision Tree: - A decision tree is drawn upside down with its root at the top.

Random Forest: - It is an ensemble learning model combining random decision forests together.

### 3.1.5 DECISION

Based on the accuracy of the different classifiers, we will choose the classifier with the highest accuracy that will give the best result to identify the emotion.

# CHAPTER-4: IMPLEMENTATION AND RESULT ANALYSIS

## 4.1 DATASET

The dataset used in this project is RAVDESS which has 24 professional actors (12 male and 12 female) vocalizing two lexically-matched statements. Each statement is produced at two levels of emotional intensity (normal and strong) having total 1440 audio files. The two statements used are:

- Kids are talking by the door
- Dogs are sitting by the door

## 4.2 MAKE NECESSARY IMPORTS

Firstly make all necessary imports.

```
In [2]: import os
        import librosa
        import matplotlib.pyplot as plt
        import soundfile
        import numpy as np
        import librosa.display
        import glob
        import sklearn.model_selection as model_selection
        from sklearn.model_selection import train_test_split
```

## 4.3 FEATURE EXTRACTION

Now we will define a function to extract the features from an audio file. Feature extraction is done using MFCC, Chroma and Mel. There are 180 extracted features with no noise.

In [6]:
```
def extract_feature(file_name, mfcc, chroma, mel):
    #here file_name is the full path location of file
    with soundfile.SoundFile(file_name) as sound_file:
        print("Currently running: ", file_name)
        X, sample_rate = librosa.load(file_name)
        if chroma:
            #stft is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes ove
            stft = np.abs(librosa.stft(X))
        result = np.array([])
        #mfc is a representation of the short-term power spectrum of a sound
        # mfcc collectively make a mfc
        if mfcc:
            mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate,n_mfcc=40).T, axis=0)
            result = np.hstack((result, mfccs))
        if chroma:
            chroma = np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T, axis=0)
            result = np.hstack((result, chroma))
        #mel comes from 'melodic'
        if mel:
            mel = np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T, axis=0)
            result = np.hstack((result, mel))
    return result
```

And then using another function (load_data) to extract features of each file and separating the target and training features. Also we will split the dataset into training and testing (80:20).

In [11]:
```
def load_data(test_size):
        x,y = [],[]
        for file in glob.glob("C:\\Users\\Rishabh Tiwari\\Desktop\\SER\\Actor_*\\*"):
            file_name = os.path.basename(file)
            emotion = emotions[file_name.split("-")[2]]
            feature = extract_feature(file, mfcc=True, chroma=True, mel=True)
            x.append(feature)
            y.append(emotion)
        return model_selection.train_test_split(x, y, train_size=1-test_size, test_size=test_size, random_state=101)
```

## 4.4 APPLYING PCA

To achieve better accuracy with the models, we will apply PCA (Principal Component Analysis) on the dataset with the ‘mle’ to decide the components. PCA also requires normalization of data.

In [14]:
```
# pre-processing the data
# for applying PCA data must be normalised first
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()

# fit_transform finds mean and standard deviation and then returns the transformed data
X_train_norm = sc.fit_transform(X_train)
# transform only transforms the data with previous required values
X_test_norm = sc.transform(X_test)

print(X_train_norm)
```

In [15]:
```
# apply PCA
from sklearn.decomposition import PCA

pca = PCA(n_components = 'mle')

X_train_pca = pca.fit_transform(X_train_norm)
X_test_pca = pca.transform(X_test_norm)

explained_variance = pca.explained_variance_ratio_
print(explained_variance)
```

After applying PCA, we have 178 new features (as decided by ‘mle’).

## 4.5 TRAINING MODELS

### 4.5.1 LOGISTIC REGRESSION

The simplest algorithm for a classification task is LR which can also do multi-class classification, but due to the large number of features and having 8 classes to categorize it is not sufficient.

### 4.5.2 K-NEAREST NEIGHBOUR

We will apply the KNN model with k=4 using sklearn. Firstly import the KNN classifier and then train the model and make predictions for the test dataset.

```
from sklearn.neighbors import KNeighborsClassifier as knn

knn = knn(n_neighbors=4)
#Train the model
knn.fit(X_train, y_train)
```

```
In [28]: y_pred_KNN = neigh.predict(X_test_pca)
         print(y_pred_KNN)
```

Now find the accuracy of the trained model.

```
print("Accuracy: {:.2f}%".format(accuracy*100))
```

```
Accuracy: 73.38%
```

### 4.5.3 SUPPORT VECTOR MACHINES

The second applied model is SVM using sklearn with the 'rbf' kernel which gives an accuracy of 76.62%.

```
In [30]: from sklearn.svm import SVC # "Support vector classifier"
         model_SVM = SVC(kernel='rbf', random_state=0, gamma=0.01, C=3)
         model_SVM.fit(X_train_pca, y_train)
Out[30]: SVC(C=3, gamma=0.01, random_state=0)
```

```
In [31]: #Predicting the test set result
         y_pred_SVM = model_SVM.predict(X_test_pca)
```

```
y_pred_SVM = model_SVM.predict(X_test_pca)
print ("Accuracy of SVM : ", accuracy_score(y_test, y_pred_SVM))
```

```
Accuracy of SVM :  0.7662337662337663
```

### 4.5.4 MLP CLASSIFIER

A multilayer perceptron classifier is a neural network based model. Here we have used 300 hidden layers for training the model. The accuracy is 83.12% which is the highest of all applied models.

```
In [21]: from sklearn.neural_network import MLPClassifier
         model = MLPClassifier(alpha=0.1, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(300,), learning_rate='adaptive', max_iter=500
```

```
In [22]: model.fit(X_train_pca,y_train)
Out[22]: MLPClassifier(alpha=0.1, batch_size=256, hidden_layer_sizes=(300,),
                       learning_rate='adaptive', max_iter=500)
```

```
y_pred=model.predict(X_test_pca)
accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)
print("Accuracy: {:.2f}%".format(accuracy*100))
```

```
Accuracy: 83.12%
```

### 4.5.5 DECISION TREE

A decision tree is drawn upside down with its root at the top. Every node acts as a decision point and has two leaf nodes to act according to the decision.

```
#Fitting Decision Tree classifier to the training set
from sklearn.tree import DecisionTreeClassifier
classifier= DecisionTreeClassifier(criterion='entropy', random_state=0)
classifier.fit(X_train_pca, y_train)
y_pred= classifier.predict(X_test_pca)
accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)
print("Accuracy: {:.2f}%".format(accuracy*100))
```

```
Accuracy: 53.90%
```

### 4.5.6 RANDOM FOREST

It is an ensemble model which combines random decision trees together.

```
from sklearn.ensemble import RandomForestClassifier
randomForest = RandomForestClassifier(max_depth=20, random_state=0)
randomForest.fit(X_train, y_train)
```

```
RandomForestClassifier(max_depth=20, random_state=0)
```

```
randomForest.fit(X_train_pca, y_train)
```

```
RandomForestClassifier(max_depth=20, random_state=0)
```

```
y_pred_forest = randomForest.predict(X_test_pca)
print("Accuracy: ", accuracy_score(y_test, y_pred_forest))
```

```
Accuracy:  0.8051948051948052
```

# CHAPTER-5: CONCLUSION

By working on this project we not only learned different models used in machine learning for classification tasks but also gained knowledge on how ML can be used to obtain the emotions of a person through his audio. This system of recognizing emotion can be in different places like with assistants for understanding command with emotions of the speaker can be used in audio surveillance, etc.

# REFERENCES


1. Towards Data Science: https://towardsdatascience.com/machine-learning/home
2. Intechopen: https://www.intechopen.com/books/social-media-and-machine-learning/automatic-speech-emotion-recognition-using-machine-learning
3. Medium: https://medium.com/topic/machine-learning
4. Sklearn: https://scikit-learn.org/stable/