**PROJECT REPORT**

**(Project Term January-May 2021)**

**Audio Emotion Detection**

**Submitted by**

**Name: Aditya Ray Registration Number : 11913275**

**Course Code: INT246**

**Under the Guidance of**

**Dr. Sagar Pande ( 23754 )**

# School of Computer Science and Engineering



**DECLARATION**

We hereby declare that the project work entitled “Speech Emotion Detection” is an authentic record of our own work carried out as requirements of Project for the award of B.Tech degree in Computer Science Engineering from Lovely Professional University, Phagwara, under the guidance of Dr. Sagar Pande ( 23754 ), during August to November 2020. All the information furnished in this project report is based on our own intensive work and is genuine.

Name of Student: Aditya Ray

Registration Number: 11913275

Signature of Student

Date: 20th Nov 2021

**CERTIFICATE**

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science And Engineering from Lovely Professional University, Phagwara.

Signature and Name of the Mentor

Designation

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab.

Date :

**ACKNOWLEDGEMENT**

I would like to thank everyone who helped me in completing this project directly or indirectly. This Project would not have been possible without help of my mentor and teacher, Dr. Sagar Pande. His guidance was utmost helpful in completion of this project.

**TABLE OF CONTENTS**

**Title**

**Declaration**

**Certificate**

**Acknowledgement**

**Table of Contents**

**INTRODUCTION**

**1. Problem Statement**

**2. Existing System**

**3. Requirement Analysis**

**4. Implementation**

**4.1 How to handle Audio Input**

**4.2 Decision Tree Model**

**4.3 CNN Model**

**4.4 MLP Model**

**4.5 Interface for Testing**

**5. Project Legacy**

**Bibliography**

**Introduction**

Emotion recognition is a good feature to integrate into various different applications like voice-based assistant systems like Google Assistant, Alexa, Siri, etc., crime detection in call monitoring, etc. The only limitation is the human mind. It can even be used at public places for monitoring criminal activities beforehand using hidden mics. Many organizations FBI (Federal Bureau of Investigation), CBI (Central Bureau of Investigation), IB (Intelligence Bureau) which are government agencies with the rights to track phone calls, can use such a technique to ease up the tracking process.

I tried to implement Decision Tree, CNN, and MLP deep learning models to classify audio under different emotion tags. Among these Decision Tree performed worst while the MLP performed the best. The CNN model can be further improved with some hyperparameter tuning.

1. **Problem Statement**

**Problem Description**

The chain linking all Human beings with one another is communication. Be it day to day personal life or even professional life humans are dependent on their effective communication skills. Many different sectors like Call Centers, Banks, Policy Business, Sales Department, etc. are a bit more dependent on their communication skills, as their work is solely based on their ability to communicate with clients. In such sectors, the emotions of the client also play a very important role.

The proposed project can be used to analyse customers’ emotions to make appropriate decisions. The same thing when said with different emotions can be considered differently, and appropriate actions can be taken accordingly.

1. **Existing System**

In this field, many different models are proposed through the years. A popular one was given by Iqbal et al. He used a different dataset named RAVDESS dataset. He used SVM, KNN, and Gradient Boosting to classify the audio samples. He observed that different models gave better accuracies for some particular models like SVM and KNN gave 100% for anger and neutral class. While Gradient Boosting did better in sad class.

I decided to try Decision Tree, CNN, and MLP for this Multiclass Classification Problem.

1. **Requirement Analysis**

I made the project with machine learning using Python. There are some software requirements to implement the project. They are given under:

* Python: It can be downloaded from its official site [link](https://www.python.org/downloads/).
* Some additional libraries that are required to do different operations are:
  + tensorflow
  + keras
  + librosa
  + matplotlib
  + pandas
  + numpy
  + pickle
  + tkinter
* Software to run Jupyter notebook files like Visual Studio Code or some online platform like Google Colab.

Since it is a tricky interface for non-tech savvy people so I made an interface.py file which is a GUI based interface for easy use.

Since machine learning is a heavily hardware-oriented task below given as some recommended hardware recommendations:

CPU: i5 8th gen or above

Ram: Min 8GB recommended

Graphics Card: Highly Recommended as it will reduce the model training time by a lot.

**4 Implementation**

The biggest problem with Audio Input is that it is not computable by default. Therefore I concverted audio signals into array of integers representing the amplitude.

**4.1 How to handle Audio Input**

I used MFCC to extract features from audio samples. I converted the audio samples to amplitudes using librosa library and then passed them to the MFCC feature extraction function. Mel frequency Cepstral coefficients algorithm is a complex algorithm with applies a number of functions to the list of amplitudes and divide the audio sample into a set number of features. I called the function to divide the audio into a list of 60 features.

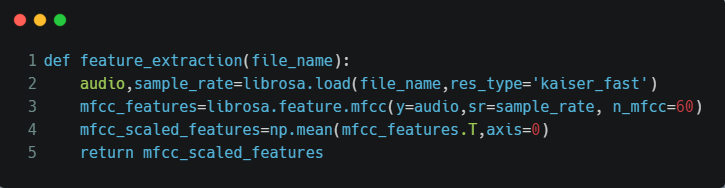


Fig1: Function to Extract Features from Audio File

These extracted features are used to train different models.

**4.2 Decision Tree Model**

Decision Tree is a Supervised Machine Learning Algorithm. I selected this as my first algorithm for project. It gave 48% accuracy. It is not a suitable to be selected as the final model.

Its confusion matrix is as follows.

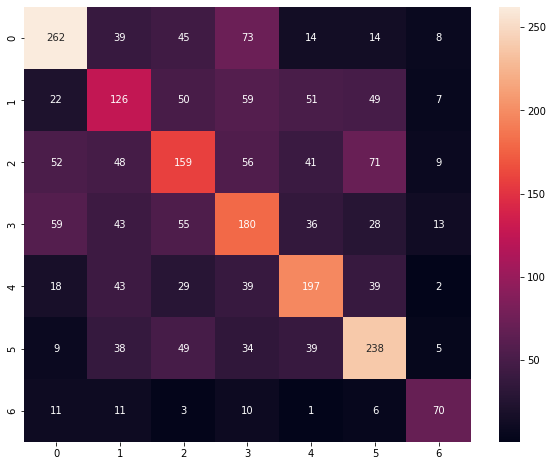


Figure 2: Confusion Matrix for Discission Tree Model

**4.3 CNN Model**

I used CNN as my next model for classification as CNN is a popular multiclass classification algorithm. I used to following model.

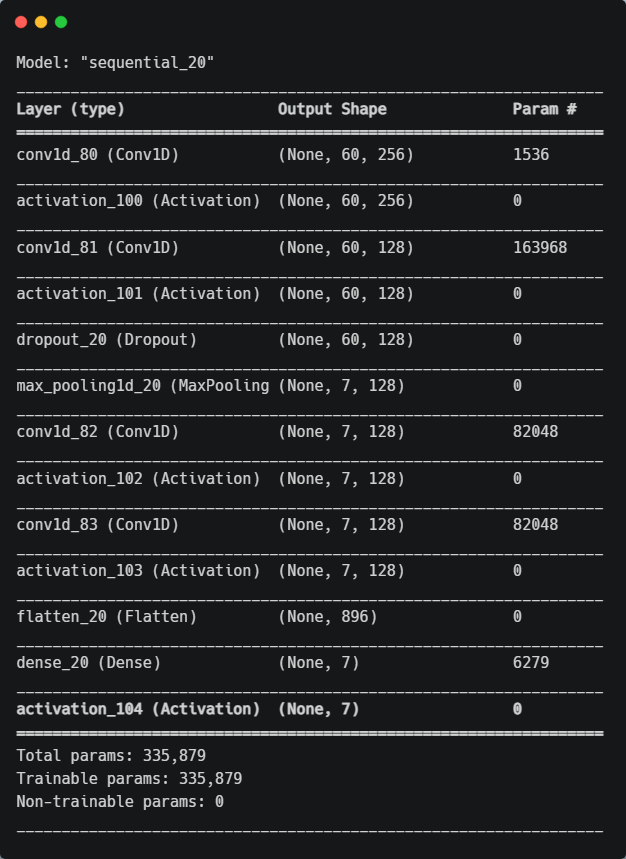
****

Figure 3: CNN Layers

The CNN Model gave ~60% accuracy. This model is acceptable but still nut good as this model was a bit underfit.

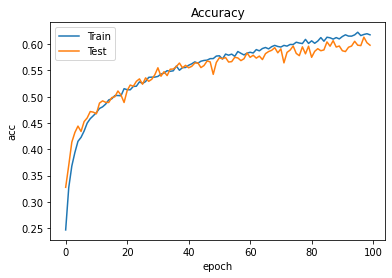


Figure 4: Accuracy Graph for CNN Model

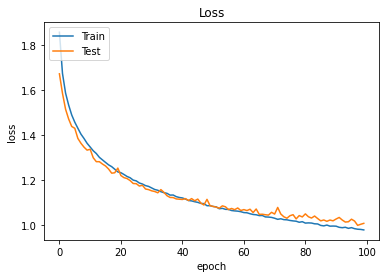


Figure 4: Loss Graph for CNN Model

Since both the graphs are almost same the model is not overfit. But since both the accuracy values are low the model can be said to be a little underfit. This can be improved by hyperparameter tuning.



Figure 5: Confusion Matrix for CNN Model

As we can see the class 0 ( Angry ) is showing unexplainable results the model cannot be said to be good. But the model gives good result for other class. But still it can’t be considered the best one.

**4.4 MLP Model**

MLP or Multilayer Perceptron is a very good multiclass classifier algorithm. It gave 91% accuracy, which is very good for any deep learning model.

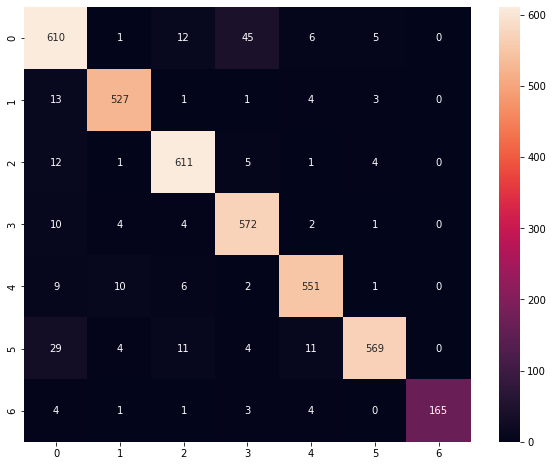


Figure 6: Confusion Matrix for MLP Model

**4.5 Interface for Testing**

Since the jupyter notebook can be hard to operate for someone who is unfamiliar with it. I made a tkinter GUI to ease the Checking process. I allowed the user to choose an emotion from the given list or set a random one. Then select a model to test that file. This UI is far better and easy to use as compared to jupyter notebook

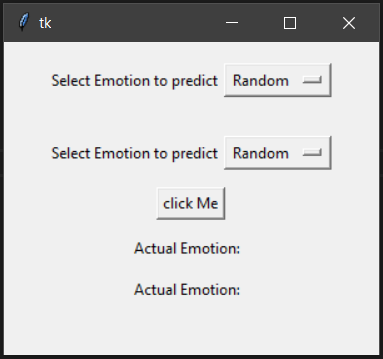


Figure 7: UI of Testing Interface

**5. Project Legacy**

**Remaining Areas Of Concern**

One of the biggest concerns of the project is that different models gives different accuracy scores for every emotion. Some model performs better for 1-2 emotion but gives very low accuracy on some other emotions. One of the possible solutions for the same could be voting. For each file be can run all the models are get percentage chances for every emotion. Then for each emotion take average of all the values for each model. And output the emotion class with highest average score of all the models.

**Bibliography**

* **towardsdatascience.com**
* **stackoverflow.com**
* **wikipedia.com**
* **betterprogramming.pub**
* **Python Documnetation: docs.python.org/3/**
* **keras.io**
* **www.tensorflow.org/guide**