Emotion Detection From Speech

A Comprehensive Project report Submitted in Partial Fulfillment of the Requirements for Award
of
the Degree of Bachelor of Technology in Information and Communication Technology

Final CP Report

Submitted by
Jay Joshi
18BIT037
Jai Lohana
18BIT035

Group 34

Submitted to
Department of Information and Communication Technology
School of Technology
Pandit Deendayal Petroleum University (PDPU)
Gandhinagar, INDIA, 382007

Declaration

I hereby declare that the project work entitled **"Emotion Detection From Speech"** is an authentic record of my own work carried out in Pandit Deendayal Energy University as requirement of B. Tech dissertation for the award of **Bachelor of Technology in Information and Communication Technology**. I have duly acknowledged all the sources from which the ideas and extracts have been taken. The project is free from any plagiarism and has not been submitted elsewhere for any degree, diploma and certificate.

Signature:.....
Jay Joshi
(18BIT037)

Signature:.....
Jai Lohana
(18BIT035)

Certificate of approval by HoD

Information and Communication Technology

Certificate

This is to certify that the project entitled **"Emotion Detection From Speech"** submitted by **Jay Joshi**, Roll No. 18BIT037 and **Jai Lohana**, Roll No. 18BIT035 to the Department of Information and Communication Technology under School of Technology, PDEU in partial fulfillment of the requirements for award of the degree of **Bachelor of Technology in Information and Communication Technology** embodies work carried out under the guidance and supervision of Dr. Mohendra Roy, Assistant Professor, Dept. of ICT.

.....

Dr. Ganga Prasad Pandey
(HOD, I.C.T. Department, PDEU.)

Dept of ICT, PDEU

Raisan Village, PDPU Rd, Gandhinagar, Gujarat 382007

Dr Mohendra Roy
Assistant Professor, Dept. of
ICT, PDEU

Certificate

This is to certify that the project entitled **"Emotion Detection From Speech"** submitted to the Department of Information and Communincation Technology under School of Technology, Pandit Deendayal Petroleum University in partial fulfillment of the requirements for award of the degree of **Bachelor of Technology in Information and Communication Technology** is a record of work carried out by **Jay Joshi**, Roll No. 18BIT037, **Jai Lohana**, 18BIT035 under my supervision and guidance in "Pandit Deendayal Energy University", "Raisan Village, PDPU Rd, Gandhinagar, Gujarat", "382007".

Signature .....
Dr Mohendra Roy(PDEU):
Assistant Professor, Dept. of ICT
Email ID: Mohendra.Roy@sot.pdpu.ac.in

Certificate of approval by evaluators

The forgoing project entitled **"Emotion Detection From Speech"** submitted by **Jay Joshi**, Roll No. 18BIT037, **Jai Lohana**, 18BIT035 to the Information and Communication Technology under School of Technology, PDEU is hereby approved as project work carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite of **Bachelor of Technology in Information and Communication Technology** degree for which it has been submitted. It has been understood that by this approval of the undersigned do not necessarily endorse or approve every statement made, opinion expressed or conclusion drawn therein but approve only for the purpose for which it is being submitted.

.....

Signature of Panel Members

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people. It is my privilege to pledge the following few lines of dedication to those who helped us directly or indirectly in completing our project.

First of all we would like to thank Dr Mohendra Roy who guided us throughout the project and let us explore various aspects of it and helped us in any way possible.

Furthermore, I am thankful to and fortunate enough to the faculties of ICT. Department for all the knowledge they have imparted, which we could put to use in this project.

Abstract

Emotion detection from speech refers to classification of emotions based on speech signal. In this project, we extract mel-frequency cepstral coefficients and zero crossing rate from the input speech signal and use these features for our new proposed classification model which consists of a stack of CNN and Bi-LSTM for the identification of emotions. Samples from various datasets are used which include RAVDESS, CREMA-D, SAVEE and TESS. Along with the base dataset, audio augmentation is also introduced to train our model to become more robust and finally optimization is performed. We achieved an accuracy of 80% which is comparable to the state-of-the art results.

Contents

Chapter 1

Introduction

1.1 Objective

Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and affective states from speech.

1.2 Solution approach

1. Literature Survey

2. Dataset selection: https://www.kaggle.com/uwrfkaggler/ravdess-emotional-speech-audio

3. Feature Engineering: To extract features from the audio files so that it can be used as an input feature for the model to be built. Through some literature review, we have identified Mel-frequency cepstral coefficients (MFCC) to be a candidate.

4. Model Building: A classification model needs to be built which would be able to classify the speech into different emotions.

1.3 Tentative Roadmap

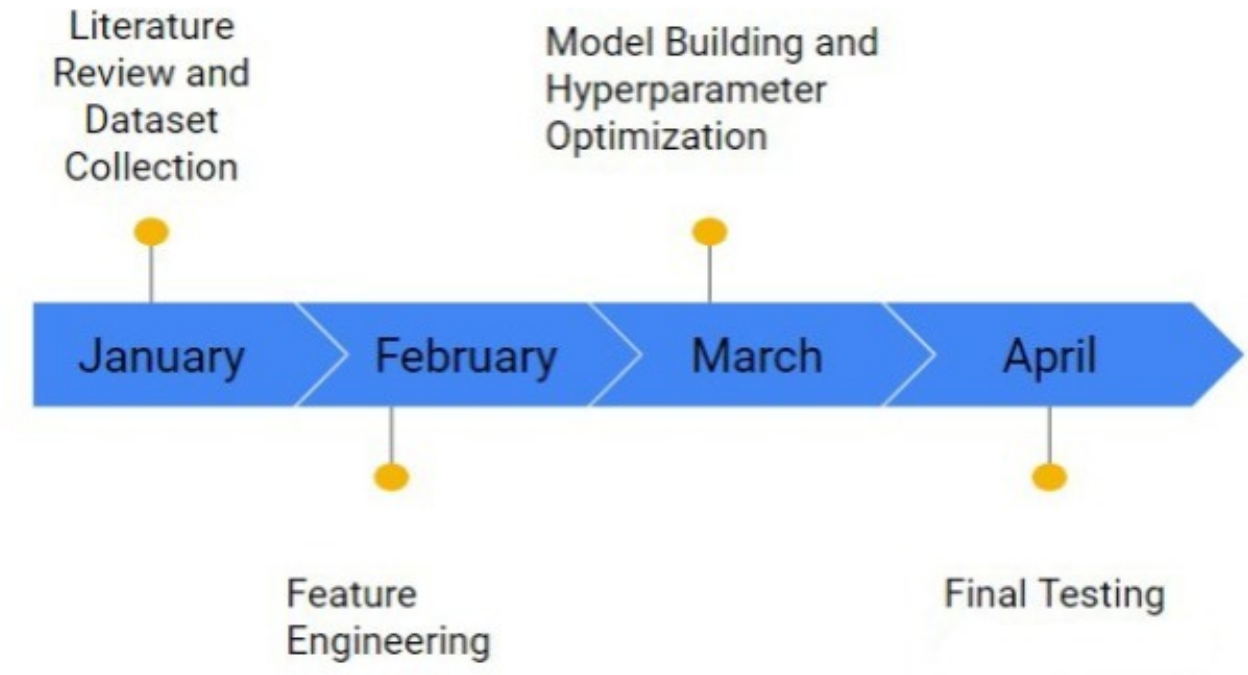


Figure 1.1: Roadmap

1.4 Problem Formulation

1. To find out the features which will help train the model.
2. To find out a dataset which can provide sufficient samples to train the model and perform data preprocessing on it.
3. To build a model capable of detecting speech signal and compare it to existing models.
4. Performing optimization on the model.

Chapter 2

Initial Research

2.1 Deciding on the Pipeline

This week, along with continuing our Literature review, we have developed a pipeline which would be followed for this project.

It describes the process of starting with the input, which is the RAVDESS dataset, performing the feature engineering required to extract the Mel-frequency cepstral coefficients and finally building the classification model required to classify the emotions.

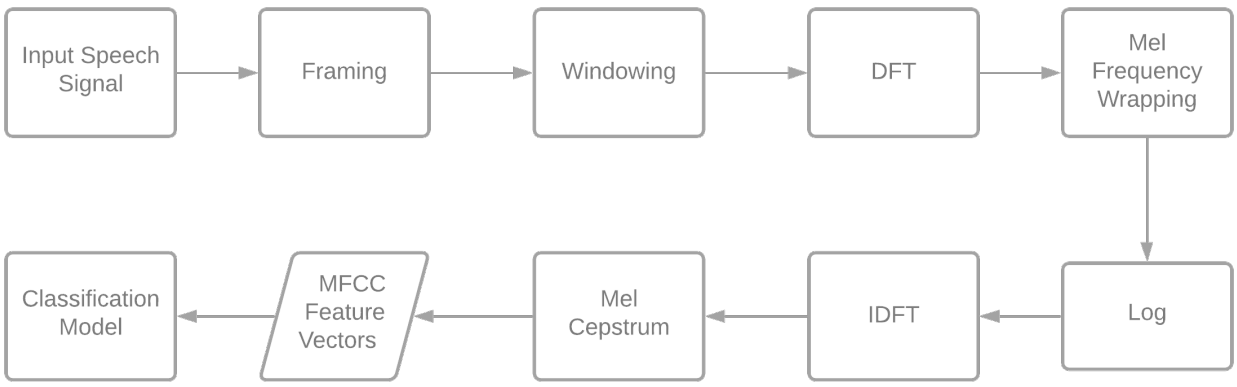


Figure 2.1: Proposed Pipeline

2.2 Selecting the Dataset

A dataset consisting of multiple different voices with all spectrum of emotion is required for the model to be trained and for it to become sufficiently generalized.

For this reason, the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) was chosen. The details of this dataset are provided below:

1. Emotions: 01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised
2. Emotional Intensity: 01 = normal, 02 = strong
3. Voice Actors: 24 - Odd numbered actors are male, even numbered actors are female

Chapter 3

Dataset Preprocessing

3.1 Dataset Collection - Continued

We found out that a single RAVDESS dataset would not be sufficient for the model to be trained. We researched and included three more datasets similar to the previous one. Overall, all the datasets included are:

1. Crowd Sourced Emotional Multimodal Actors

2. Ravdess

3. Surrey - Audiovisual expressed emotion

4. Toronto Emotional Speech Set

3.2 Data Preprocessing

Since all the four datasets are from different sources, a standard should be created which would point to source of the audio as well as label it according to the actual emotion of the speech so that we can have a proper supervised dataset. This was done for all four of the datasets and stored in a csv file. This file would be accessed first, which would then point to the audio file and also identify the label.

	A	B	C	D	E	F	G	H	I	J	K	L
1	labels	source	path									
2	male_disgust	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_d03.wav									
3	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a13.wav									
4	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a10.wav									
5	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a09.wav									
6	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a08.wav									
7	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a12.wav									
8	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a07.wav									
9	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a06.wav									
10	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a04.wav									
11	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a05.wav									
12	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a03.wav									
13	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a11.wav									
14	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a02.wav									
15	male_angry	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_a01.wav									
16	male_neutral	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_n09.wav									
17	male_neutral	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_n02.wav									
18	male_neutral	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_n08.wav									
19	male_neutral	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_n11.wav									
20	male_neutral	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_n05.wav									
21	male_neutral	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_n04.wav									
22	male_neutral	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_n03.wav									
23	male_neutral	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_n01.wav									
24	male_happy	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_h11.wav									
25	male_happy	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_h12.wav									
26	male_happy	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_h15.wav									
27	male_happy	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_h09.wav									
28	male_happy	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_h05.wav									
29	male_happy	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_h10.wav									
30	male_happy	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_h07.wav									
31	male_happy	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_h06.wav									
32	male_happy	SAVEE	/content/drive/My Drive/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/DC_h13.wav									

Figure 3.1: Path Data

Chapter 4

Literature Review and Feature Engineering

4.1 Literature Review

1. This paper selects Mel-frequency Cepstral Coefficients, Mel-scaled spectrogram, Chromagram, Spectral contrast feature and Tonnetz representation as the features. They note that MFCC and Mel spectrum are generally used for this purpose however, chromagram is also used to tackle the problem of distinguishing pitch classes. The model used is a 1 Dimensional (Kernel moves in 1 direction) CNN with dropout, batch normalization and activation layers (Relu and Softmax). They later modify this model a few times and take the ensemble of all their models. This ensemble model outperforms those they referenced, however these models performed on raw audio inputs rather than feature engineered. They note that order of stacking the sound features also plays an important role.

2. Here, they use raw audio features which is a spectrogram instead of processed features like Mel Spectogram or MFCC. The model is a CNN with three fully connected layers. The accuracy achieved is 84.3 percent, which is less than the one in the first paper.

3. In this paper, four features are used. These are MFCC, Mel Energy Spectrum Dynamic Coefficients (MEDC), Pitch and Energy. They use three layers which are fully connected layer, Pooling/subsampling layer and a convolution layer. They then try a combination of features such as just MFCC, MFCC+MEDC etc and find that providing all the selected features provides the best result for the model of 93.8931 percent and also a greater precision. However, other models also achieve over 92 percent accuracy.

4. In this paper, they have used MARYAS to extract twenty-six MFCC features. They have used Probabilistic neural networks, extreme learning machines, standard back propagation neural networks and SVM as the identifiers for emotion classification. They found that SVM had the best results.

5. In this they have employed MFCC Rather than taking the simple spectrogram of the input speech signal, so that it will represent the signal better than the spectrogram. CNN accept the two dimensional input therefore they have converted the input speech signal in to the MFCC spectrum. For the classification they have used CNN having 3 layers and by using MFCC+CNN(LAYER 3) they have achieved highest accuracy.

6. In this paper, they use all thirty-nine of the MFCCs instead of the regular 13. These are used in a broad learning system and then identification of emotions is done and a testing accuracy of around seventy-five% is achieved.

4.2 Feature Extraction

Mel-frequency cepstral coefficients: Generalized formula:

$$C(x(t)) = F^{-}1(log(F(x(t))))$$

Where x(t) is sound signal

4.2.1 Mel Spectrogram Steps

1. Short-Term Fourier Transform

2. Amplitude to decibels

3. Conversion of scale from frequency to mel.

    - Select number of mel bands first.
    - Create the mel filter bands.
    - Then, use the mel filter bands on the spectrogram.

4.2.2 Short-Term Fourier Transform

Formula for STFT is:

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n\frac{k}{N}}$$

Where, k=frequency, m=Frame Number, N=Frame Size, w(n)=Window Function

STFT can be used to find out the Spectrogram, which gives us information about both time and frequency domain. It can be found out by:

$$Y(m,k) = |S(m,k)|^2$$

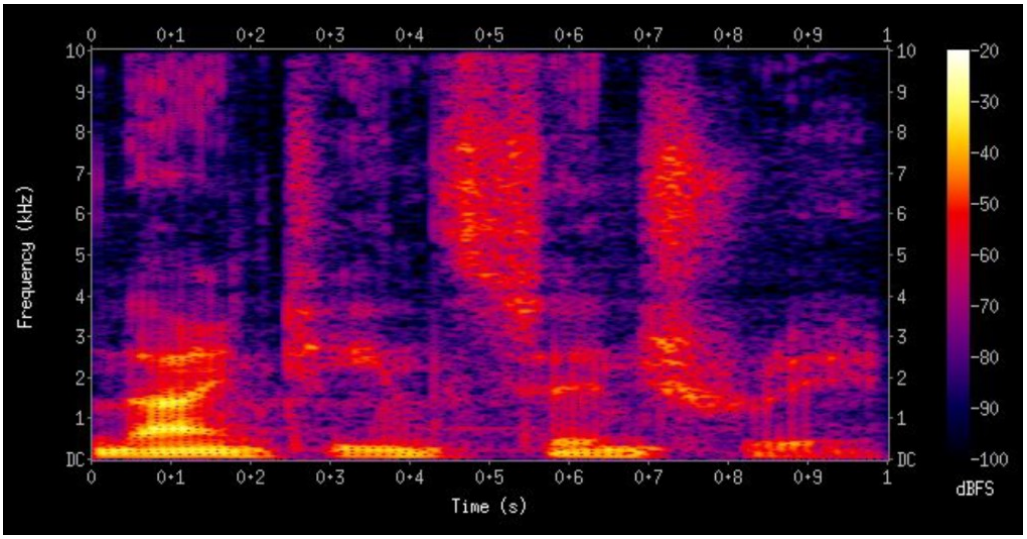


Figure 4.1: Spectrogram

4.2.3 Steps to create Mel filter banks

Step 1: Converting lowest/highest frequency to Mel

$$m = 2595 \cdot log(1 + \frac{f}{500})$$

Step 2: Creating some bands using equally spaced points
Step 3: Converting points back to Hertz

$$f = 700(10^{m/2595} - 1)$$

Step 4: Creating triangular filters

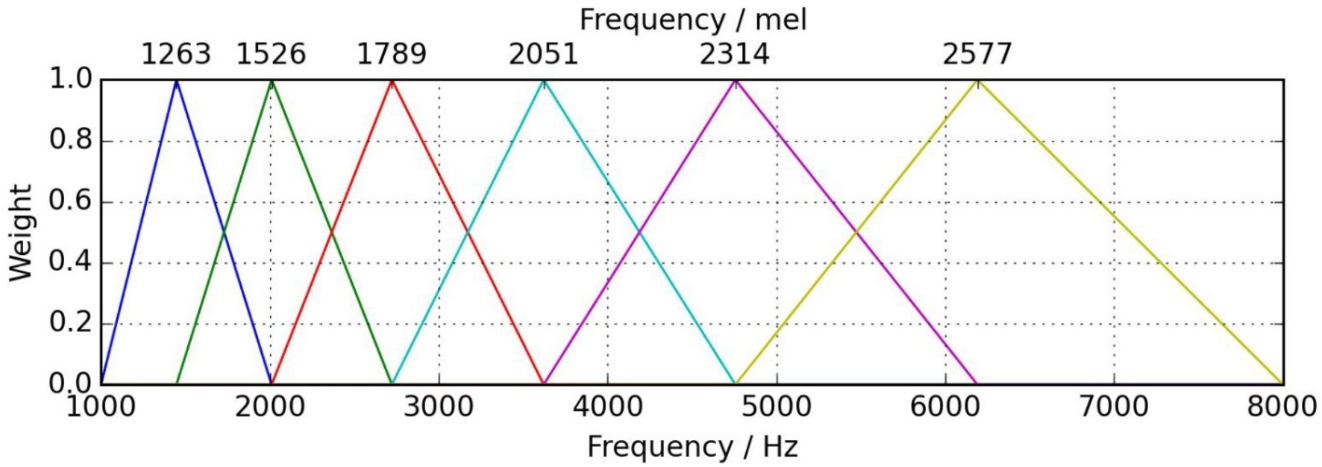


Figure 4.2: Mel Filter Bands

4.2.4 Mel Spectrogram

Mel Filter Bank's shape (M) = (No. of bands, Framesize/2 + 1)

We already know our spectrogram's shape,

Y = (Framesize/2 + 1, No. of frames)

Therefore, we get Mel Spectrogram = MY (No. of bands, No. of Frames)

Using either Inverse Discrete Fourier Transform or Discrete Cosine Transform provides us with MFCC.

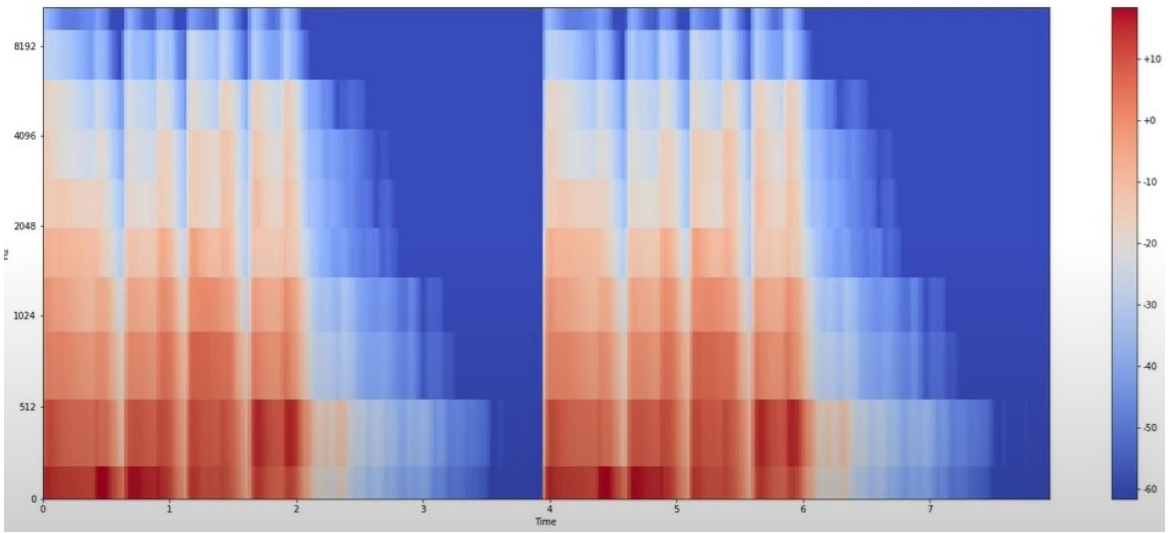


Figure 4.3: Mel Spectrogram with 10 Mel Bands

Chapter 5

Initial Model Building

5.1 Building the Model

This week we started working on the model architecture. First, we had to make sure that all the different examples in the input were of same length. This was done by neglecting the 0.6s of start of the signal and a total duration of 3s. After all the features were extracted from these signals, the input was normalized and then fed into the model,

5.2 Architecture of the model

We are using the Conv1D from tensorflow. Conv1D takes input of a 2D matrix and the kernel moves along in a single dimension. It consists of 5 convolution modules, which are made up of a convolution layer, batch normalization layer and a max pooling layer. After that we have used flatten layer which feeds its output to a dense layer. We have used the Rmsprop optimizing algorithm and loss function is categorical cross entropy.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_5 (Conv1D)            (None, 216, 512)          3072

batch_normalization_6 (Batc  (None, 216, 512)          2048
hNormalization)

max_pooling1d_5 (MaxPooling  (None, 44, 512)           0
1D)

conv1d_6 (Conv1D)            (None, 44, 512)           1311232

batch_normalization_7 (Batc  (None, 44, 512)           2048
hNormalization)

max_pooling1d_6 (MaxPooling  (None, 9, 512)            0
1D)

conv1d_7 (Conv1D)            (None, 9, 256)            655616

batch_normalization_8 (Batc  (None, 9, 256)            1024
hNormalization)

max_pooling1d_7 (MaxPooling  (None, 2, 256)            0
1D)

conv1d_8 (Conv1D)            (None, 2, 256)            196864

batch_normalization_9 (Batc  (None, 2, 256)            1024
hNormalization)

max_pooling1d_8 (MaxPooling  (None, 1, 256)            0
1D)

conv1d_9 (Conv1D)            (None, 1, 128)            98432

batch_normalization_10 (Bat  (None, 1, 128)            512
chNormalization)

max_pooling1d_9 (MaxPooling  (None, 1, 128)            0
1D)

flatten_1 (Flatten)          (None, 128)               0

dense_2 (Dense)              (None, 512)               66048

batch_normalization_11 (Bat  (None, 512)               2048
chNormalization)

dense_3 (Dense)              (None, 14)                7182

_________________________________________________________________
```

Figure 5.1: Model Summary

Chapter 6

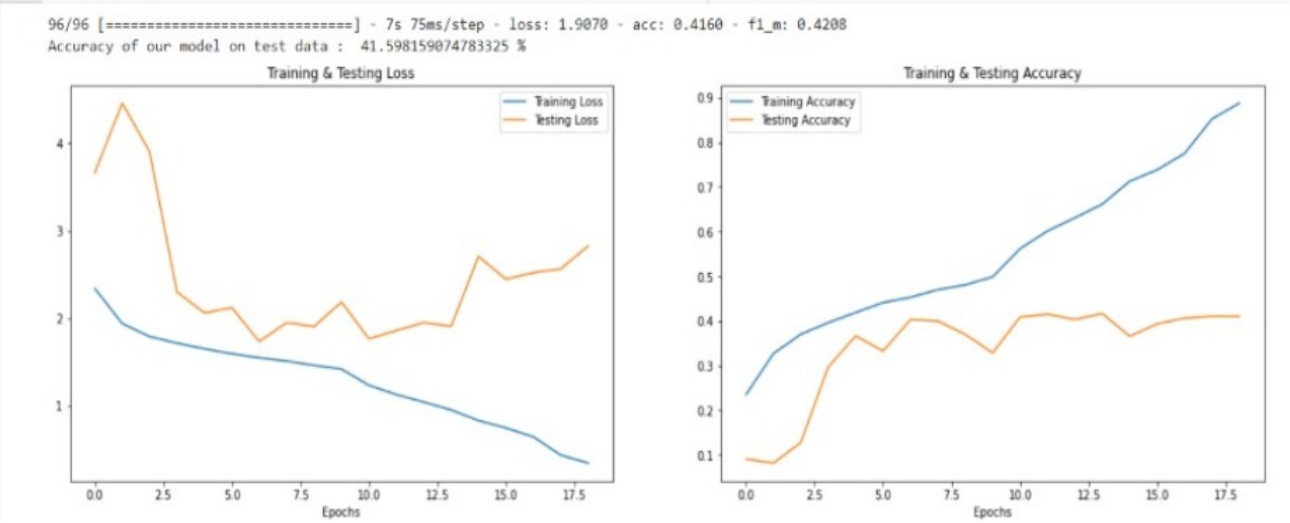
Initial Results

6.1 Implementing the model

we have trained the model on three different combinations of datasets and these are the results which we got

1. Model evaluated on datasets - Ravdess, Cremad and Savee.

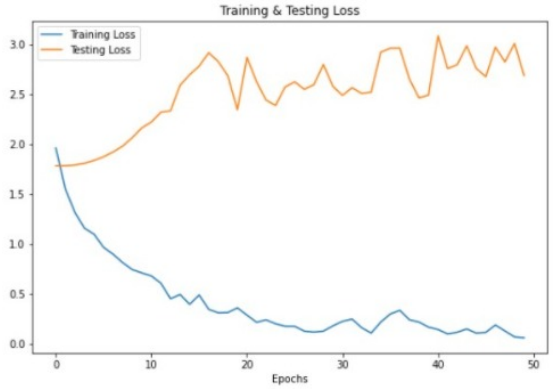
   Epochs: 50 (Earlystopping: 19), Batches: 64 Testing accuracy: 41



2. Model evaluated on datasets -Ravdess and Savee (Male Labels only).

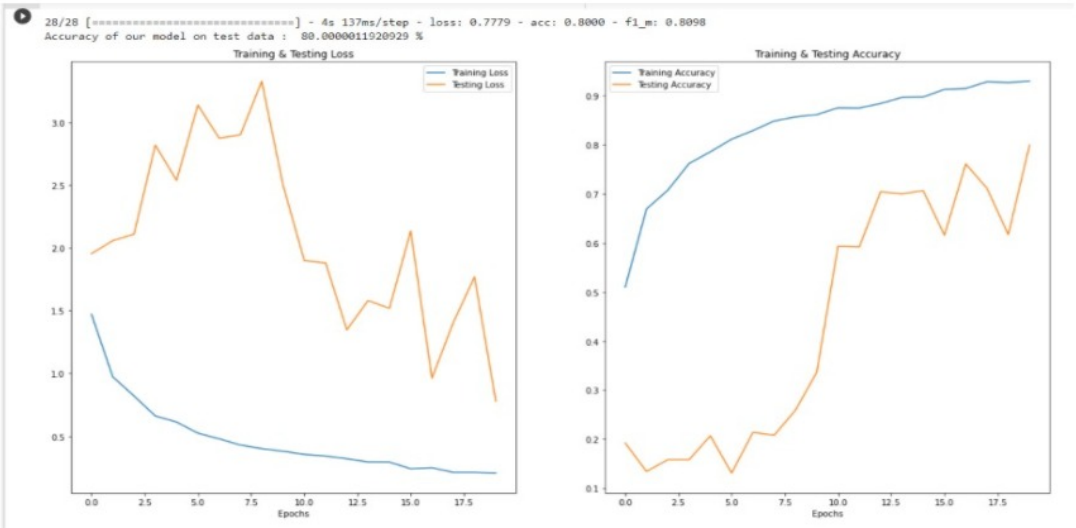
   Epochs: 50, Batches: 32 Testing accuracy: 45.9

```
9/9 [==============================] - 0s 5ms/step - loss: 2.6867 - acc: 0.4598 - f1_m: 0.4265
Accuracy of our model on test data :  45.97701132297516 %
```



3. Model evaluated on datasets - Ravdess, Cremad and Savee(Female labels only).

   Epochs: 20, Batches: 64 Testing accuracy: 80



Chapter 7

Introducing Regularization On The Model

7.1 Regularization

Regularization is a method used to reduce the value of loss function and also avoid over or under fitting the model. We have used L2 regularization in our model.

7.2 L2 Regularization

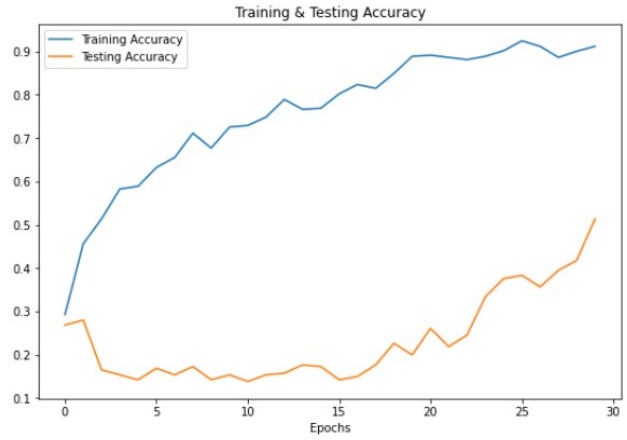
At first, we used to minimize only the loss function. However, in L2 Regularization we now minimize the complexity along with the loss. It can be written as:

Min(Loss(Data—Model) + Lambda * Complexity(Model))

Let the complexity be a function of weight. Then it is taken that a high or larger absolute valued weight is more complex than when compared to one that has a lower absolute value. In L2 regularization, the complexity is found out by squaring all of the weights. Lambda is an added hyperparameter which is used to control the extent of regularization. Higher the lambda, higher the effect of regularization.

Applying this regularization in our model, with the value of lambda being 0.005, we achieved a higher accuracy on a dataset consisting of male labels, which had an accuracy of 45% before. Now, we achieved an accuracy of 51%.

```
9/9 [==============================] - 0s 28ms/step - loss: 2.1639 - acc: 0.5134 - f1_m: 0.4869
Accuracy of our model on test data :  51.34099721908569 %
```



Chapter 8

Addition of ZCR

8.1 Zero-Crossing Rate

We decided to use ZCR along with MFCC as features for our model. We are processing the audio signals to perform the feature engineering required to extract ZCR from the audio files.

8.1.1 What is Zero-Crossing Rate?

Zero Crossing Rate or ZCR of a signal is the frequency at which the signal changes its sign. That is, when the signal's values go from negative to positive or when it goes positive to negative. This rate is then divided by the size or length of the frame. The formula of ZCR is as follows:

$$Z(i) = \frac{1}{2W_L} \sum_{n=1}^{W_L} \left| \, sgn[x_i(n)] - sgn[x_i(n-1)] \right|,$$

where $sgn(\cdot)$ is the sign function, i.e.

$$sgn[x_i(n)] = \begin{cases} 1, & x_i(n) \geq 0, \\ -1, & x_i(n) < 0. \end{cases}$$

ZCR can be used to measure the noisiness of a speech signal. It has also been used in simple algorithms that detect pitch. It is because of these properties; we have used it as an input feature. They are easy to compute as well, which reduces the time complexity of feature extraction part.

ZCR is extracted from our dataset using the following function from the library Librosa: librosa.feature.zero_crossing_rate(y=Data)

Chapter 9

New Model Implementation

9.1 Stacking CNN and Bi-directional LSTM

In order to improve the previously obtained results, we constructed a new model. This model consists of a stack of CNN model first to extract the local features and then Bidirectional LSTM to work on the extracted features while also looking at the long-term reliance of the data.

Other parameters include: Epochs = 75, Batch Size = 16, l2 Regularize hyper-parameter (lambda) = 0.01

```
#New Model - 2
model =Sequential()
model.add(Conv1D(128, 5, padding='same',
                         input_shape=(X_train.shape[1], 1),
                         kernel_regularizer=regularizers.l2(l2=0.01),
                       bias_regularizer=regularizers.l2(l2=0.01)))
model.add(Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPool1D(pool_size=5, padding="same"))
model.add(Dropout(0.2))
model.add(Conv1D(64, 5, padding='same',
                         input_shape=(X_train.shape[1], 1),
                         kernel_regularizer=regularizers.l2(l2=0.01),
                       bias_regularizer=regularizers.l2(l2=0.01)))
model.add(Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPool1D(pool_size=5, padding="same"))
model.add(Bidirectional(LSTM(units=64)))
model.add(Flatten())
model.add(Dense(6, kernel_regularizer=regularizers.l2(l2=0.01),
                       bias_regularizer=regularizers.l2(l2=0.01)))
model.add(Activation('softmax'))
model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["acc", f1_m])
```

9.2 Results

Testing this model on the previous datasets and providing the features as MFCCs and ZCR, we acheived an accuracy of 76.47% and a F1-score of 0.7630

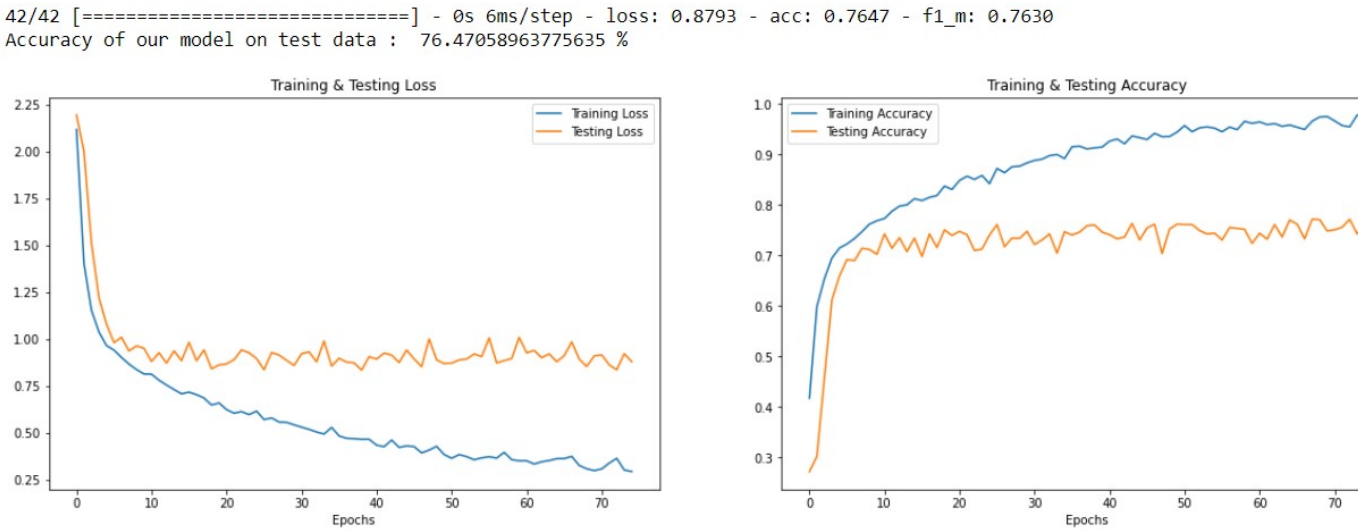


Figure 9.1: Loss and Accuracy Graphs

9.2.1 Confusion Matrix

As seen in the confusion matrix below, this model is able to better classify each class irrespective of genders, which was not achieved before.

```
Confusion matrix, without normalization
[[157   9  25  21   5   2]
 [  5 154   5   5  13  15]
 [  6   4 163  15  12  12]
 [ 13   3  15 163  20   6]
 [  3  12   0  17 222  23]
 [  3   9   5   6  27 168]]
```

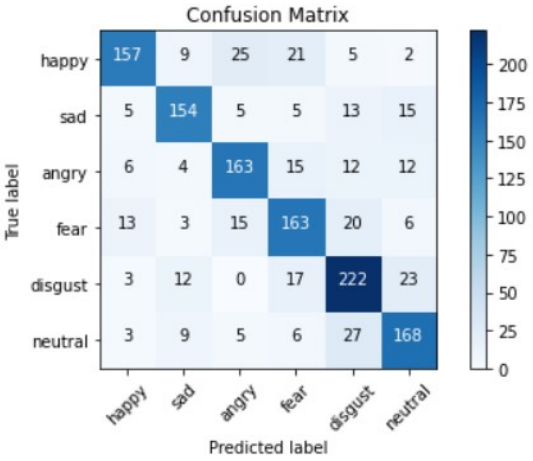


Figure 9.2: Confusion Matrix

Chapter 10

Data Augmentation

10.1 Addition of Noise in the data

Since in real-life scenarios, the audio signal to be processed would be consisting of some noise, we need to train our model along with noisy data to make it more robust. For this purpose, we use Gaussian Noise

10.1.1 Gaussian Noise

Gaussian or white noise has zero as its mean and the standard deviation is adjusted according the use-case. Adding less noise is not useful as it doesn't have a noticeable difference on the training data and adding lot of noise makes the data difficult to learn. This is controlled by the standard deviation.

The function used is as follows:

```
#Function to augment data with noise
def noise(data,rate=0.035, threshold=0.075):
    noise_amp = rate*np.random.uniform()*np.amax(data)
    data = data + noise_amp*np.random.normal(size=data.
    return data
```

Figure 10.1: Noise function

10.1.2 Result

```
84/84 [==============================] - 1s 7ms/step - loss: 0.9385 - acc: 0.7423 - f1_m: 0.7439
Accuracy of our model on test data :  74.2271900177002 %
```

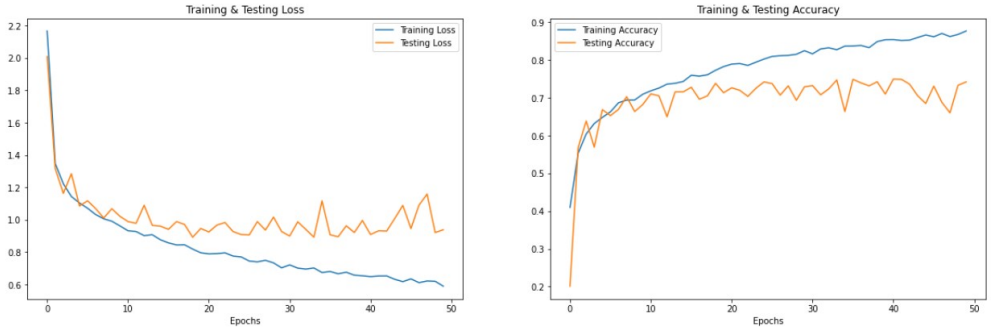


Figure 10.2: Loss and Accuracy

Chapter 11

Optimization

11.1 Optimizing Hyperparameters

To optimize the hyperparameters, we have used the Bayesian Optimization method. It is an approach that uses Bayes Theorem to direct the search in order to find the minimum or maximum of an objective function.

11.1.1 Bayesian Optimization

Posterior probability: The probability we need to find out is known as this.

Surrogate function: It is a function which is a Bayesian approximation of the objective function.

Activation function: This is used to select the next sample.

The Bayesian Optimization algorithm can be summarized as follows:

1. Select a Sample by Optimizing the Acquisition Function.
2. Evaluate the Sample With the Objective Function.
3. Update the Data and, in turn, the Surrogate Function.
4. Go To 1.

11.1.2 Results

By applying Bayesian Optimization, we can get an increase in accuracy by about 5%.

The parameters to be evaluated:

```
params = {
    'optimizer': ['adam','rmsprop','sgd'],
    'activation': ['relu','tanh'],
    'batch_size': [16,32,64],
    'neurons':Integer(100,600),
    'epochs':[50,100],
}
```

Figure 11.1: Parameters

Accuracy and Loss Graphs

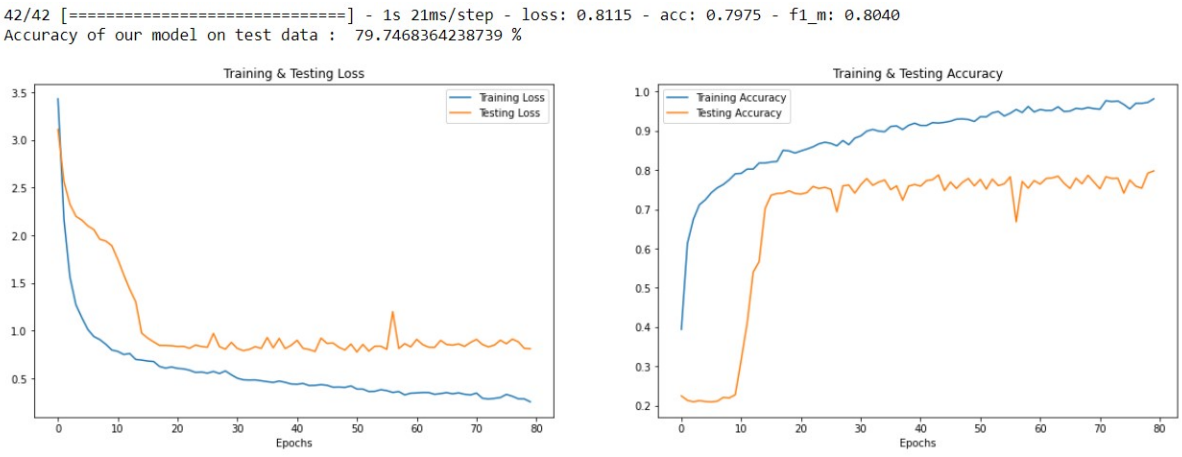


Figure 11.2: Loss and Accuracy

Confusion Matrix

```
Confusion matrix, without normalization
[[157   9  25  21   5   2]
 [  5 154   5   5  13  15]
 [  6   4 163  15  12  12]
 [ 13   3  15 163  20   6]
 [  3  12   0  17 222  23]
 [  3   9   5   6  27 168]]
```

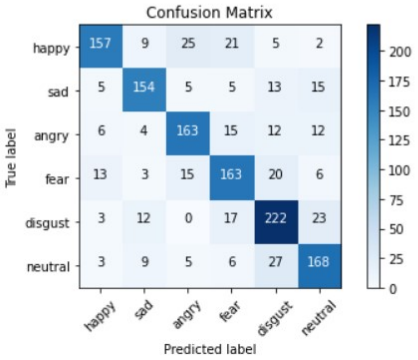


Figure 11.3: Confusion Matrix

Chapter 12

Conclusion and Future Work

12.1 Conclusion

From the previously shown results we can see that an accuracy of approximately 80% can be achieved using only the MFCC and ZCR features. This proves to be helpful in scenarios where emotions can't be extracted using facial features. We can successfully conclude that relying on speech for emotion detection is possible and can be used extensively in many applications.

12.2 Future Work

1. Since the process of extraction of audio features from speech is a heavily engineered process, we can experiment on using simple Mel Spectrograms as features to check for any biases introduced.

2. Using a different architecture of model for detection can be used instead, such as the transformer networks that makes use of attention, which allows the model to focus on relevant part of the inputs.

Bibliography

[1] Issa, Dias; Fatih Demirci, M.; Yazici, Adnan (2020). Speech emotion recognition with deep convolutional neural networks. Biomedical Signal Processing and Control, 59, 101894–. doi:10.1016/j.bspc.2020.101894

[2] Badshah, Abdul Malik; Ahmad, Jamil; Rahim, Nasir; Baik, Sung Wook (2017). [IEEE 2017 International Conference on Platform Technology and Service (PlatCon) - Busan, South Korea (2017.2.13-2017.2.15)] 2017 International Conference on Platform Technology and Service (PlatCon) - Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network., 1–5. doi:10.1109/PlatCon.2017.7883728

[3] Pawar, M.D., Kokate, R.D. Convolution neural network based automatic speech emotion recognition using Mel-frequency Cepstrum coefficients. Multimed Tools Appl 80, 15563–15587 (2021). https://doi.org/10.1007/s11042-020-10329-2

[4] Yang, Ningning; Dey, Nilanjan; Sherratt, R. Simon; Shi, Fuqian (2020). Recognize basic emotional statesin speech by machine learning techniques using mel-frequency cepstral coefficient features. Journal of Intelligent and Fuzzy Systems, 1–12. doi:10.3233/JIFS-179963

[5] H. Cheng and X. Tang, "Speech Emotion Recognition based on Interactive Convolutional Neural Network," 2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP), 2020, pp. 163-167, doi: 10.1109/ICICSP50920.2020.9232071.

[6] Yang, Z., Huang, Y. Algorithm for speech emotion recognition classification based on Mel-frequency Cepstral coefficients and broad learning system. Evol. Intel. (2021). https://doi.org/10.1007/s12065-020-00532-3