**Sentiment Classification Using Speech Recognition**

**Data Used:** Dataset contain 1440 audio files which were in the wav format from the following websites: <http://neuron.arts.ryerson.ca/ravdess/?f=3>

1440 audio files are split into Train, Validation and Test datasets.

There two kind of dataset :

1. Audio

2. Video

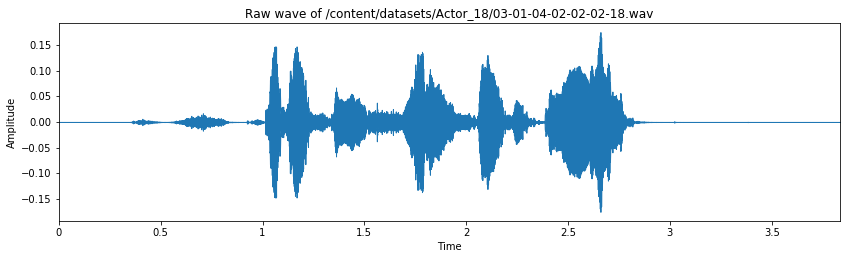
I used only audio.

*Filename identifiers*

* Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
* Vocal channel (01 = speech, 02 = song).
* Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
* Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.
* Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
* Repetition (01 = 1st repetition, 02 = 2nd repetition).
* Actor (01 to 24. Odd numbered actors are male, even numbered actors are female).

*Filename example: 01-01-06-01-02-01-12.wav*

1. audio-only (0)
2. Speech (01)
3. Fearful (06)
4. Normal intensity (01)
5. Statement "dogs" (02)
6. 1st Repetition (01)
7. 12th Actor (12)
8. Female, as the actor ID number is even



The next step involves organizing the audio files. Each audio file has a unique identifier at the 6th position of the file name which can be used to determine the sentiment the audio file consists. We have 8 different emotions in our dataset.

1. Neutral
2. Calm
3. Happy
4. Sad
5. Angry
6. Fearful
7. Disgust
8. Surprised

**Feature Extraction :**

The next steps involve shuffling the data, splitting into train and test and then building a model to train data.

For audio analysis and feature extraction I used Librosa library. Then I have used two methods for.

. feature extraction

1. **MFCC — Mel-Frequency Cepstral Coefficients**
2. **Spectral Rolloff**

For both feature extraction method the artificial neural network model was same. Also, data argumentation was added. Data was shifted, pitch, added noise and speed and pitch. Then the data were feeded into the model.

**Model:**

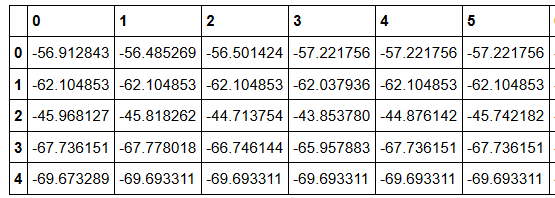
I built a simple CNN model. Where there are total 8\_canv layer corresponding every conv\_layer activation, dropout, batch normalization is used. Here valid padding us used for pooling layer. There are total 8 different emotion and 2 gender class. So final dense layer has 16 neurons with a softmax activation. Categorical\_crossentropy is used as loss function. Adam with learning rate 0.0001 is used for optimization.

**Model Summary:**

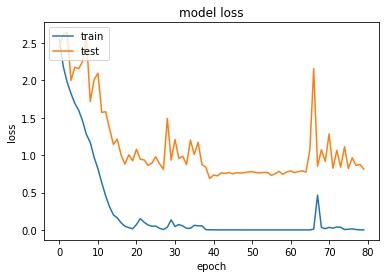


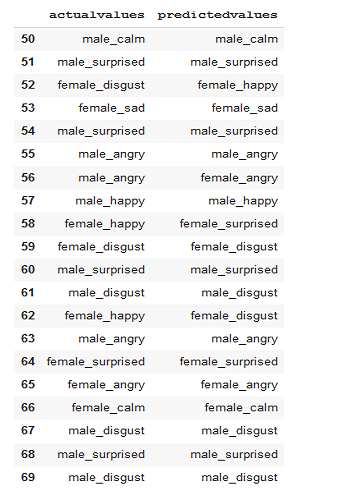
**!.Mel-Frequency Cepstral Coefficients:**

**1.1.Feature map:**



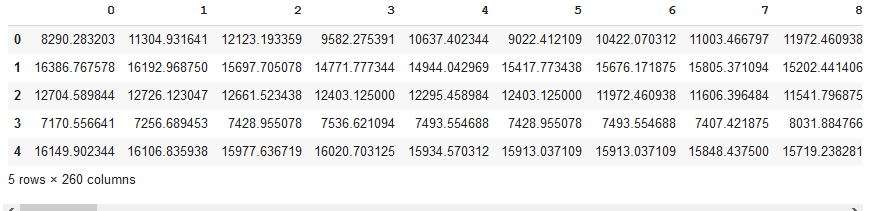
**1.2.Training Accuracy: 78%**



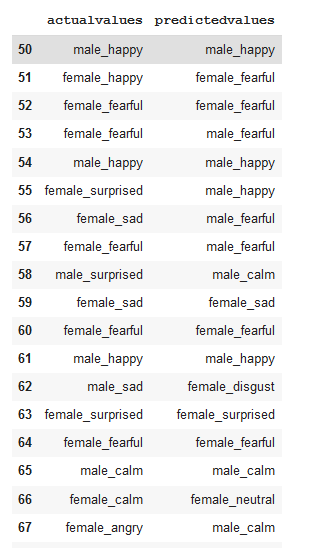
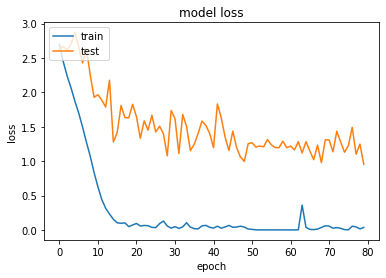


**2.Spectral Rolloff:**

**2.1.Feature map:**



2.**2.Training Accuracy: 73%**

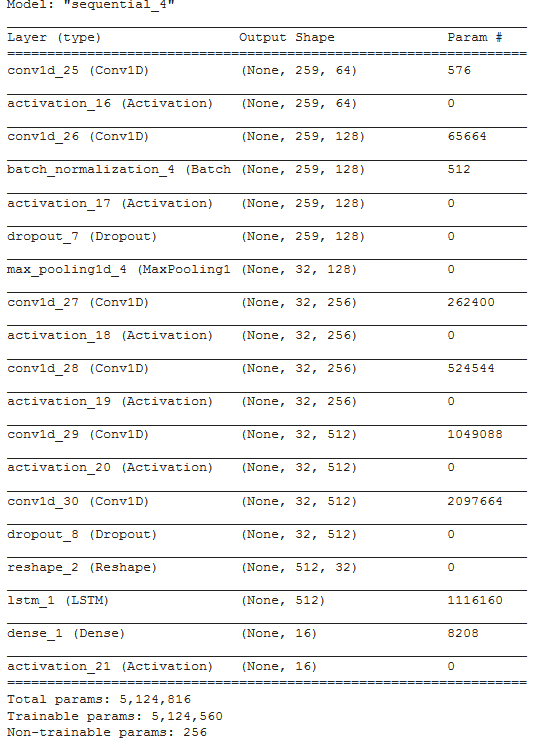


**3.Hybrid Model:**

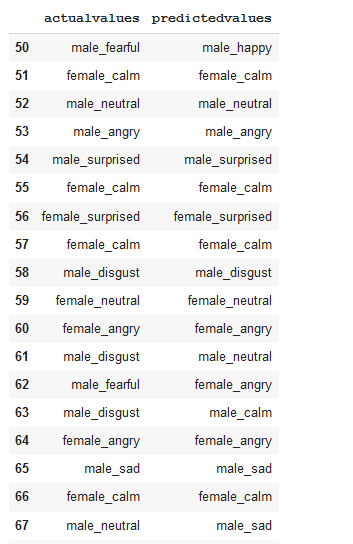
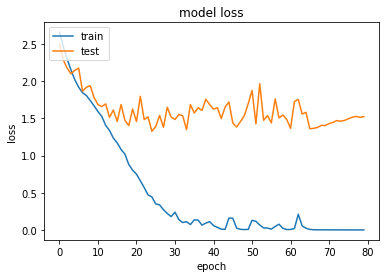
Here , we can see the model performed well with **Mel-Frequency Cepstral Coefficients(MFCC) feature map then Spectral Rolloff. So, I have used a simple hybrid model. In, hybrid model it contain conv\_6th layer of the cnn model as it was in CNN model. Other layer are cutoff. After that a LSTM (Long-Short Term Memory) cell is added. Before adding the LSTM cell we need to reshape (512,-1). After LSTM cell a dense layer is add with 16 neurons for classification with a softmax activation. In this model there are total 5,124,560 trainable parameters.**

**As Cnn model here also adam optimizer is used with learning rate 0.0001.** Categorical\_crossentropy is used as loss function

**Model summary:**



**3.2 Training Accuracy: 69.54**



**4. Conclusion:**

|  |  |
| --- | --- |
| Model | Accuracy |
| CNN + MFCC | 78.15 |
| CNN +  **Spectral Roll off** | 73.65 |
| Hybrid Mode (CNN + LSTM) | 69.54 |

It’s a basically a simple speech recognition model. Where the model can detect the sentiment within the speech. There are total 16 classes of sentiment and both male and female are equal distribution. Adam is used for optimization. CNN model has outperformed every other model where MFCC was used for audio feature extraction. It has an accuracy of 78.15%. Because it’s train on little dataset that why its accuracy is low. But the result is promising.