VOICE EMOTION ANALYSIS

VOICE EMOTION ANALYSIS

A project report submitted in partial
fulfillment of the requirements for the degree of
Master of Science

By

Ranjan Nandkumar Raut
Master of Science, University of Colorado Denver, 2020
Bachelor of Engineering in Computer Engineering, University of Mumbai, 2012

12/2019
University of Colorado Denver

ABSTRACT

As of now, many methods have been established to recognize sentiment in the sentence using only text keyword's analysis. But, sometimes pitch of the vocal sentence projection makes difference in the emotion behind the sentence. In other words, voice modulation does bring change in the emotion.

This project predicts emotion from Speech. Emotion recognition from speech is difficult task. Voice modulation differ from person to person, since it differs in speaking styles, language etc. Therefore, it is very important to extract correct feature from speech. I have used Mel-frequency cepstral coefficients (MFCCs) from speech to capture such features.

I have tried two approaches in order to achieve this task. In first approach, I have used Convolution Neural Network Architecture of Artificial Neural Network, whereas in Second Approach I have used Long Short-term Memory Architecture of Artificial Neural Network. Both approaches work fine in achieving objective of Emotion Recognition.

This Project Report is approved for recommendation to the Graduate Committee.

Project Advisor:

_____

Dr. Ashis Kumer Biswas

MS Project Committee:

_____

_____

TABLE OF CONTENTS

1. Introduction 1

1.1 Problem 1

1.2 Project Statement 1

1.3 Approach 2

1.4 Organization of this Project Report 2

2. Background 3

2.1 Key Concepts 3

2.1.1 Mel-Frequency Cepstral Coefficients (MFCCs) 3

2.1.2 Convolutional Neural Network (CNN) 4

2.1.3 Long Short-Term Memory (LSTM) 5

2.1.4 Adam Optimizer 6

2.1.5 Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) Dataset 7

2.2 Related Work 8

2.2.1 Sentiment Analysis on Speaker Specific Speech Data 8

3. Architecture 9

3.1 High Level Design 9

3.2 Initial setup and Preprocessing 10

3.3 Implementation 11

3.3.1 CNN 12

3.3.2 LSTM 13

4. Methodology, Results and Analysis 15

4.1 Methodology 15

4.2 Results 16

4.3 Analysis 17

5. Conclusions 18

5.1 Summary 18

5.2 Potential Impact 18

5.3 Future Work 19

References 20

LIST OF FIGURES

Figure 1: Waveform for Audio Clip 3

Figure 2: Spectrogram of MFCC values 4

Figure 3: CNN Architecture 5

Figure 4: LSTM Unit Architecture 6

Figure 5: Proposed structure of the Sentiment Analysis System by Authors 8

Figure 6: System Architecture 10

Figure 7: CNN Architecture 13

Figure 8: LSTM Architecture 14

Figure 9: Normalized Confusion Matrix using CNN model 15

Figure 10: Normalized Confusion Matrix for LSTM model 15

Figure 11: Epoch vs Loss 16

Figure 12: Epoch vs Accuracy 17

1. INTRODUCTION

1.1 Problem

Voice Modulation is one of the important factors in doing sentence projection. Without voice modulation, sentence sounds very monotonous or neutral. In such cases, listeners either get bored or distract easily. Voice modulation not only shows confidence of speaker but also helps in convincing ideas/beliefs to the listeners. It helps in communicating speaker's messages more effectively. Very important factor behind convicing the listener is emotion in sentences. In voice modulation, sometimes pitch goes high, sometimes it goes low. Sometimes, it is required to strech a word or to put some pauses.

Now a days, advancement in technology has brought Machine Learning algorithms in the world. One of the popular use case of machine learning is Sentiment Analysis using text parsing. Words are powerful factor in determing Sentiment of statement. Such as sequence of words, emotion related to individual words are few of many factors in determing Sentiment of overall sentence.

But Text parsing doesn't capture most important factor. Sentence projection while delivering speech, i.e. Voice Modulation. This project captures Emotion behind any sentence based only on Voice Modulation i.e. Sentence Projection using voice.

1.2 Project Statement

Goal of this project is to build artificial neural network, which will classify emotion behind a sentence based on modulation in sentence projection.

1.3 Approach

Every speech sentence has highs and lows, stretch on the words or pause in between. In order to capture such details, Mel-frequency cepstral coefficients (MFCCs) are used. In short, we can visualize it as spectrogram for a sentence. These MFCC values can be then feed to Artificial Neural Networks.

In this project, I have used 2 approaches. In First approach, I have used Covolution Neural Network Architecture, whereas in Second approach, I have used Long short-term Memory Architecture. Both these methods are used to determine classification of emotion for a given speech/voice sample.

1.4 Organization of this Project Report

In Chater 3, Architecture, High Level Design and Implemntation Details are explained. Chaper 4 describes Methodology, Results using both approaches and detailed analysis.

2. BACKGROUND

2.1 Key Concepts

2.1.1 Mel-Frequency Cepstral Coefficients (MFCCs)

MFCCs are typically used in sound/music processing. Mel Frequency Ceptral Coefficients (MFCCs) altogether form Mel-Frequency Ceptral (MFC). Short-term power spectrum can be represented by MFC. Power Spectrum shows frequency distribution of power over time series. MFCCs are retrieved from a type of ceptral form/representation of sound. In MFC, frequency bands are equally spaced on mel scale, which mimics human auditory system's response. [1]

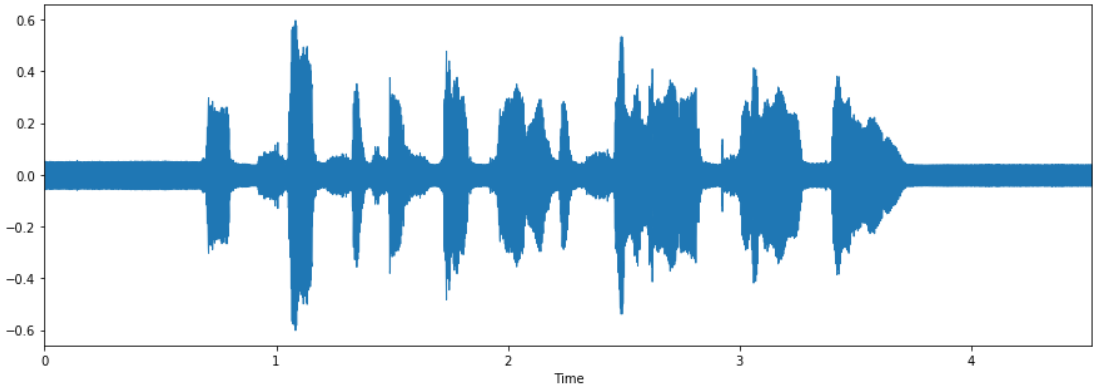

Figure 1: Waveform for Audio Clip

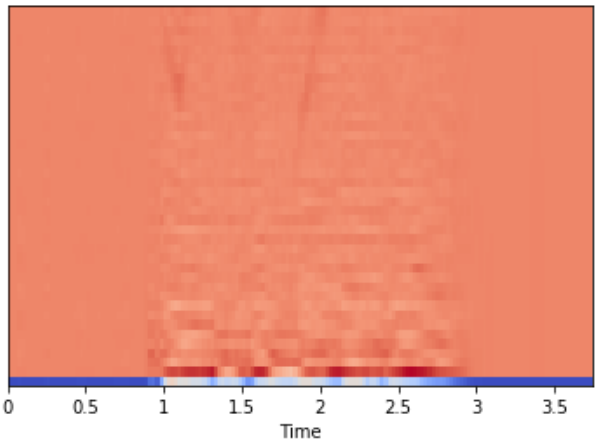

Figure 2: Spectrogram of MFCC values

2.1.2 Convolutional Neural Network (CNN)

CNN is one of the architectures of Deep Neural Network, mostly used for image processing. It has got its name because of the type of hidden layers it got. Hidden layers consist of Convolution Layer, Pooling Layer, Fully Connected Layers. Very first step is Convolution Step. In this step, different filters also know as kernels are applied over input image. Depending on size of input image and size of kernel, output of this step produces data in lesser dimention than input dimention. One can visualize this step as sliding many different small image frames by certain fixed intervals over complete image. Next step is Pooling. Pooling layer reduce dimention data of neuron clusters in single layer into on neuron for the next layer. One of the famous pooling method is max pooling. The next step to flatten the output of pooling layer into single dimention. After flattening, input is then feed to fully connected Artificial Neural Network, which have weight associated to each perceptron. Also, each perceptron has activation function associcated at the end.

Many times, CNN is used for image classification. In such cases, output layer has softmax activation function, which generates probabilities for all categories which sums upto 1.

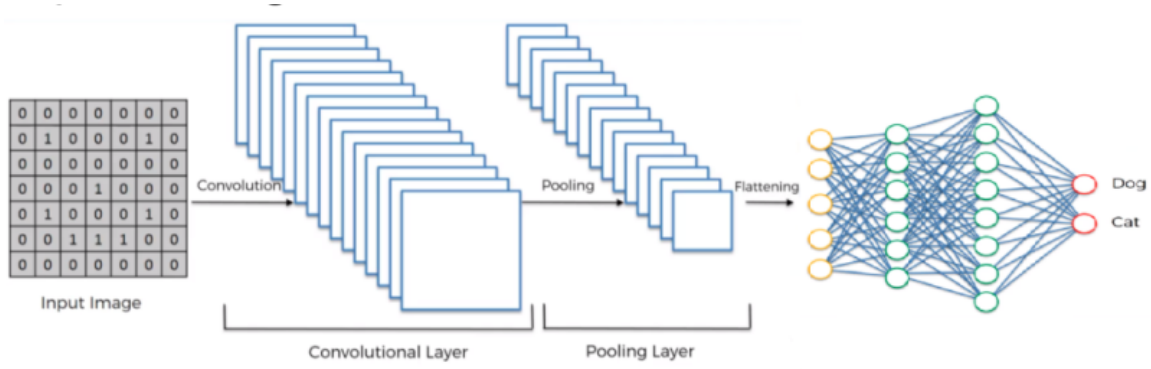

Figure 3: CNN Architecture

2.1.3 Long Short-Term Memory (LSTM)

LSTM is a type of Recurrent Neural Network architecture, which helps in sequence prediction problems. LSTM has feedback connections. Not only it can predict single data points, but also sequential data. LSTMs were developed in order to resolve occurrence of vanishing gradient problem and exploding gradient problem in traditional Recurrent Neural Network. LSTMs are ideal for classification, intermediate processing or making prediction in time series.Many LSTM units form a LSTM architecture.

A single LSTM unit consists of [2]:

1. Cell : Stores values over arbitrary time interval.
2. Input Gate : Input gate controls, how much of new value should flow into new cell
3. Forget Gate : Forget Gate helps in forgetting unncecessary information, thus filtering information to store in cell.

4. Output Gate : Output Gate controls, how much of value stored in cell should be used to compute output of LSTM unit.

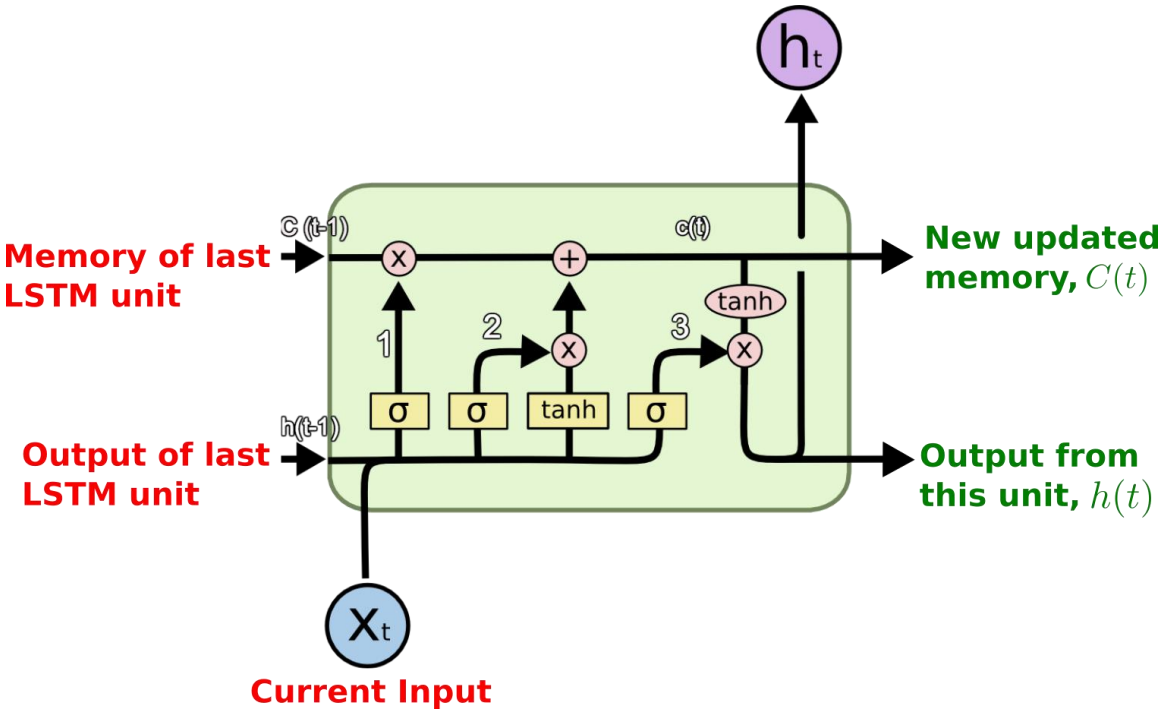

Figure 4: LSTM Unit Architecture

2.1.4 Adam Optimizer

Adam Optimizer can be seen as best of Stochastic Gradient Descent and RMSProp, but with momentum. It has moving average of gradient descend and it also uses squared gradients to scale learning rate, just like RMSProp. Adam optimizer calculates learning rate individually for every parameter; thus, it is one of the adaptive learning rate method. Name Adam is derived from Adaptive Moment Estimation.[3]

2.1.5 Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) Dataset

RAVDESS dataset is released under Creative Commons Attribution license and made available from Zenodo. RAVDESS contains three types of files. Full-Audio Video, Video Only, Audio Only. Dataset contains recordings of 24 actors speaking 2 statements in North American Accent. Actors have portrayed emotions along with intensity in two modes. First mode is Speech, whereas Second Mode is Song. Altogether, there are 1440 files present just for Audio Speech. Each file name has many identifiers suggesting below specifications. E.g. Filename: '03-01-06-01-02-01-04.wav' [4]

- 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).

2.2 Related Work

2.2.1 Sentiment Analysis on Speaker Specific Speech Data

In this paper, authors have proposed model for Speech Emotion Recognition. Firstly, voices are stored in the form of chunks into database, and those are passed to Speech Recognition System and Speaker Discrimination System. In Speaker Discimination System, firstly features are extracted from Speech using Mel-Frequency Cepstral Coefficients. Then Speaker Ids gets assigned to chunks by Speaker Discriminator after comparision. Those chunks are then converted to text by Speech recognition System. Text output from speech which is tied to individual speaker is then served as feature in order to identiy Sentiment [5]. Below figure explains Architecture.

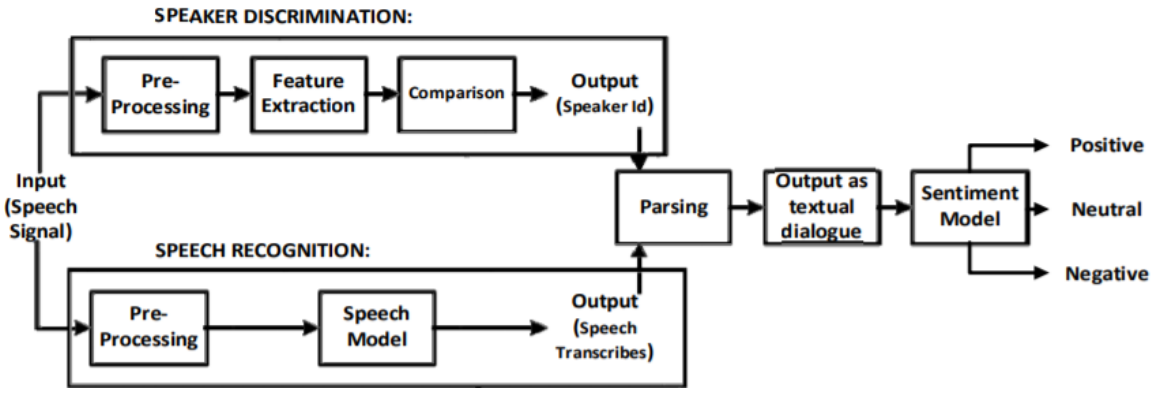

Figure 5: Proposed structure of the Sentiment Analysis System by Authors

3. ARCHITECTURE

3.1 High Level Design

I have built classifier model using below system architecture. Overall, architecture can be divided into two stages. First stage is training stage. In this stage, Speech Dataset (RAVDESS) is read from local file system. Then it is preprocessed. Pre-processed data is then split into training and validation dataset. Training Dataset is used to train model, whearas Validation Dataset is used to tune model. Both datasets are then feed to Artificial Neural Network Model. Once Training and Tuning is done, model is then saved into local file system. In Second stage, we can do classification on the fly. Firstly, we record audio clip using python utility. Then we process that audio clip in the format that model can accept. After that, we load Model from local file system into memory and load reformatted audio clip data into model. Output of model is nothing but detected emotion.

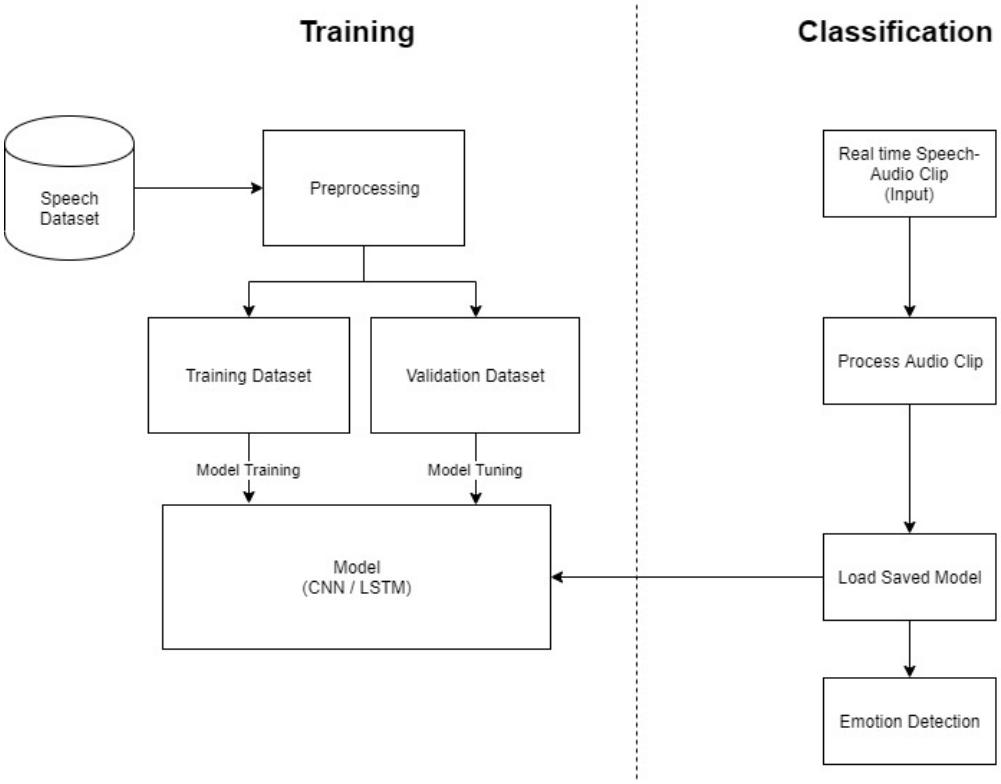

Figure 6: System Architecture

3.2 Initial setup and Preprocessing

This project was run on Inspiron 7570 Laptop, which has Intel Core i5-8250U CPU @1.6GHz processor, DDR4 16GB RAM and NVIDIA GeForce 940MX GPU. PyCharm IDE is used for code development, whereas Anaconda Environment is used for Python environment. In External Python Libraries, NumPy, LibROSA, Sounddevice, Tensorflow with Keras, Matplotlib and scikit-learn are used.

I used RAVDESS dataset in this project. I have converted Video-Audio Speech Data to Audio Format and used it along with Audio Speech Data provided in Dataset. File format in dataset is in form of audio clips (‘.wav’ format). Since, this format can not

be feed to any ANN model directly, I had to preprocess it. Firstly, I convered audio clip to audio time series and then obtained 40 MFCCs per audio time series. These MFCCs can be now feed to ANN model. I figured that, ‘Neutral’ and ‘Calm’ audio samples had similar Voice Modulations, whereas many samples from ‘Disgust’ has resemblance to samples of ‘Angry’. For that reason, I merged those categories. Audio Samples from categories ‘Fearful’ and ‘Surprised’ have lot of similarities with mixed samples of ‘Angry’, ‘Happy’, ‘Sad’, which was causing confusion for model to recognize. For that reasom I excluded categories ‘Fearful’ and ‘Surprised’. In total, after preprocessing, I have data labeled into 4 Categories:- Neutral, Sad, Happy, Angry. I have also included my own voice samples in Dataset, so that Model will be able to recognize my voice accent (test purpose).

Furthermore, dataset is divided into Training Dataset, Validation and Test Dataset.

3.3 Implementation

The goal of this project to accurately recognize Emotion/Sentiment of Speaker through Voice Modulations in real time. In order to achieve this, I did experiment using two approaches. First approach is using CNN Model and Second approach is using LSTM Model, which I have described in subsections below. Once required model is trained, then it is saved on local file system.

To detect emotion on demand, I have written another script in python. In the beginning, required type of trained model is loaded from file system. Then, audio is recorded and saved on local file system. In order to feed this audio clip to model, I had to

process it. I convered audio clip to audio time series and then obtained 40 MFCCs per audio time series. These MFCCs are then feed to trained model. Output of Model is nothing, but emotion type recognized from audio sample.

3.3.1 CNN

First approach is using CNN architecture. Preprocessed Data is first fed to Convolution Layer, which has 128 feature maps and Kernel Size of 5x5. Each of the output fo through ReLU activation function. Then, output goes through Max Pooling Layer, which has pool size of 8. Output of MaxPooling Layer is then fed to another Convolution Layer, which again has 128 feature maps and Kernel Size of 5x5. Each of the output again has ‘ReLU’ activation function. Output is then fed to Max pooling layer, which has pool size of 2. It is then followed by Flattening layer, which then distributes data into 256 Neurons. Final layer is output layer, having 4 Neurons. Each having ‘Softmax’ activation function. This layer gives classification for a given Voice Sample.

Since, this model has a purpose of classification, I have used ‘categorical_crossentropy’ as loss function. I have used ‘adam’ as an optimizer function. Once model is trained and optimized, it is then saved on local file system.

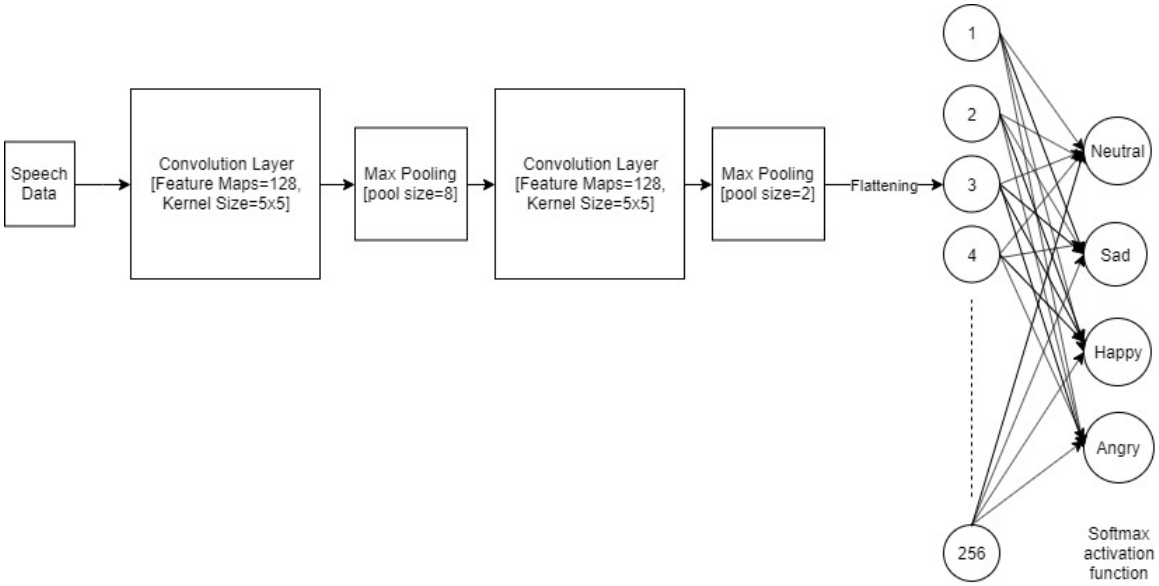

Figure 7: CNN Architecture

3.3.2 LSTM

Second Approach is using LSTM (Long short-term Memory) architecture. Data is fed into 100 LSTM units. Output of LSTM unit is fed to next LSTM unit and given to neuron in next layer. Next Layer consist of 100 Neurons, each having 'ReLU' activation function. Outcome of this layer is then feed to final Output layer, which has 'Softmax' activation function. This layer gives classification for a given Voice Sample.

Since, this model has a purpose of classification, I have used 'categorical_crossentropy' as loss function. I have used 'adam' as an optimizer function. Once model is trained and optimized, it is then saved on local file system.

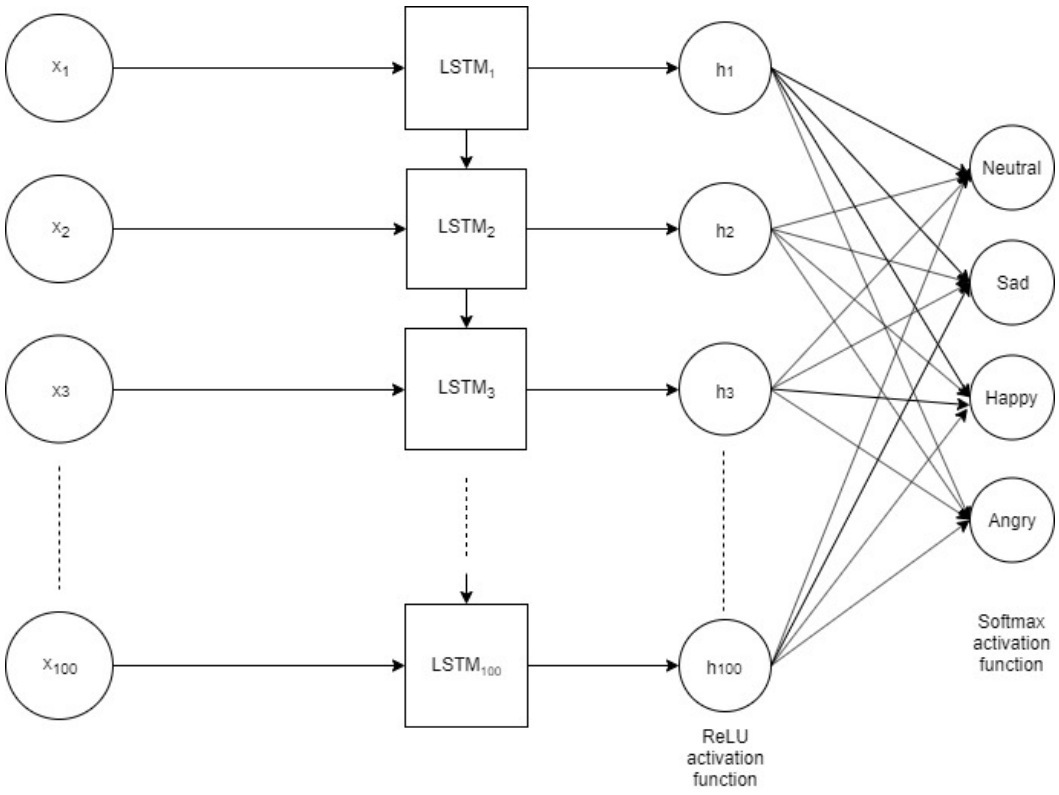

Figure 8: LSTM Architecture

4. METHODOLOGY, RESULTS AND ANALYSIS

4.1 Methodology

To evaluate both models, I have used confusion matrix. This allows more detailed analysis than mere proportion of correct classifications. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. A confusion matrix is a summary of prediction results on a classification problem [6].

Predicted Value

True Value	Neutral	Sad	Happy	Angry
Neutral	0.98	0.01	0.01	0.0
Sad	0.03	0.84	0.04	0.09
Happy	0.06	0.07	0.79	0.07
Angry	0.02	0.0	0.05	0.93

Figure 9: Normalized Confusion Matrix using CNN model

Predicted Value

True Value	Neutral	Sad	Happy	Angry
Neutral	0.94	0.02	0.0	0.04
Sad	0.06	0.8	0.06	0.09
Happy	0.03	0.06	0.73	0.18
Angry	0.02	0.02	0.03	0.92

Figure 10: Normalized Confusion Matrix for LSTM model

4.2 Results

Both models can recognize emotion in real time. After preprocessing, CNN model took approximately 17 minutes for training, whereas LSTM model took approximately 42 minutes for training. Figure 11 shows loss value per epoch for both models. At the end of 1000th epoch, CNN model had loss value of 0.0908 for training data and 0.8390 for validation data, whereas LSTM model had loss value of 0.0056 for training data and 0.9466. Figure 12 shows accuracy value per epoch for both models. At the end of 1000th epoch, CNN model had accuracy of 0.9736 for training and 0.9011 for validation data whereas LSTM model had accuracy of 1 for training data and 0.8681 for validation data. For testing data, CNN model got accuracy of 90%, whereas LSTM model got accuracy of 87%.

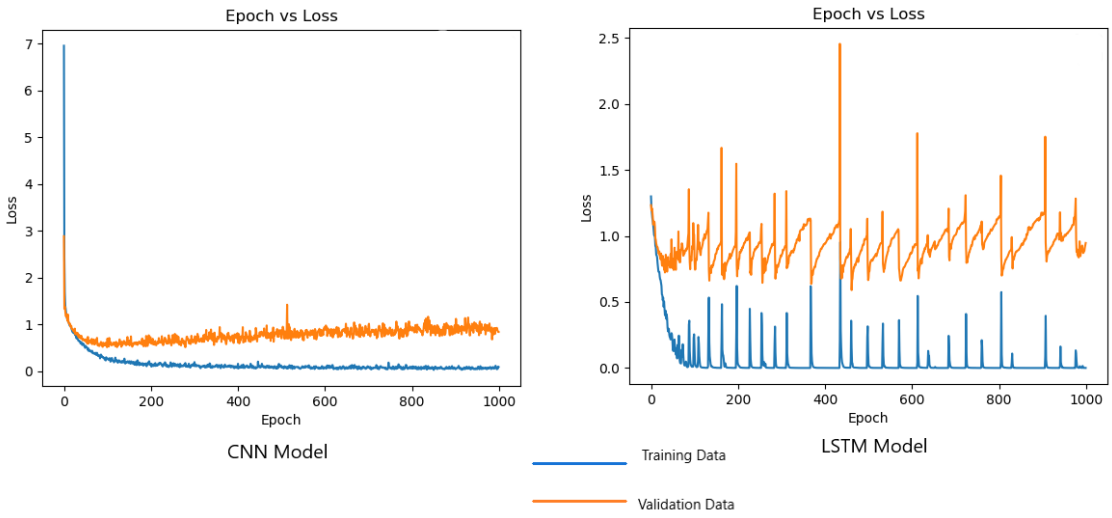

Figure 11: Epoch vs Loss

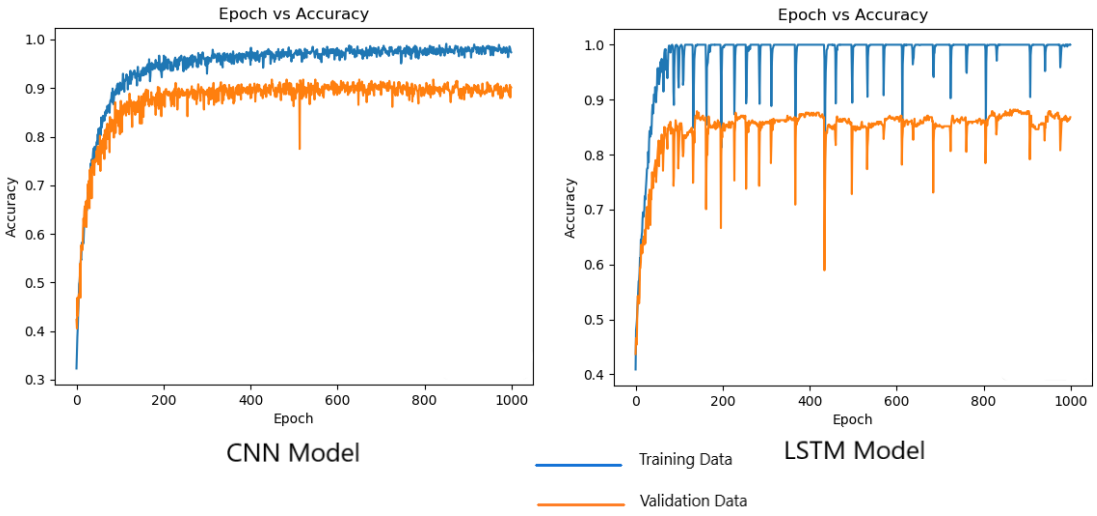

Figure 12: Epoch vs Accuracy

4.3 Analysis

From figures 12, we can see that CNN model has accuracy little higher than LSTM model for validation and testing dataset. We can also see that loss values are stable for CNN model than LSTM. Thus, CNN model appears more accurate than LSTM model.

Also, while doing real time emotion recognition I realize that Voice Modulation differs from person to person in terms of accent, language etc. In order to recognize emotion from my speech, I had to include my voice samples in training data.

5. CONCLUSIONS

5.1 Summary

Speech Emotion Recognition is one of the popular and challenging topics in the field of Machine learning. To accomplish this task, Firstly, I acquired MFCCs from audio samples. Those values were then feed to CNN and LSTM models. Although, both of those models got more than 85% accuracy, CNN model slightly outperformed LSTM model. This explains that both models can recognize features from voice samples, and both are able to classify emotion from it. Another standalone utility can load these models and recognize emotion from an audio sample on demand.

5.2 Potential Impact

With arising voice assistant services such as Amazon's Alexa, Google Assistant, Siri, Cortana, machines now have power to take actions based on commands. Yet, machines don't have capability to analyze emotion of speaker's voice modulation. We can set a specific set of target action for such services, if they can recognize emotion when user speaks with them. E.g. Play songs on the mood, play a joke, make coffee in case of connected home etc.

In cars, if we can figure out emotional state of the driver correctly, then we can ask driver to take precautionary steps such as slowing down, more focus, calming down to prevent accidents.

In Call Centers, we can determine customer satisfaction from sentiment analysis of their recorded calls.

5.3 Future Work

This project recognizes only emotions. In future, more classificatios can be done as an extension this project. For example, Gender Recognition, Language Recognition only based upon Voice Sample. This project will improvise more, if it uses speech-to-text service and recognize overall emotion from both models (Sentiment Analysis from Text and Sentiment Analysis from voice sample).

REFERENCES

[1] https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

[2] https://en.wikipedia.org/wiki/Long_short-term_memory

[3] https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

[4] https://smartlaboratory.org/ravdess/

[5] Maghilnan S, Rajesh Kumar M, "Sentiment Analysis on Speaker Specific Speech Data", 2017 International Conference on Intelligent Computing and Control (I2C2), VIT University, Tamil Nadu, India.

[6] https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

This is the final page of a Project Report and should be a blank page