BANGLA SPEECH RECOGNITION USING 1D CNN AND LSTM WITH DIFFERENT DIMENSION REDUCTION TECHNIQUES

by

S M Mahsanul Islam Nirjhor

14201031

Mohammad Abidur Rahman Chowdhury

15201049

Md. Nazmus Sabab

16101135

This thesis report is submitted to the Department of Computer Science and Engineering in order to fulfill partial requirements for the degree of B.Sc. in Computer Science

Department of Computer Science and Engineering

Brac University

August, 2019

Declaration

We hereby stating that

1. The thesis that we submitted in order to complete bachelor degree at Brac University-is our own original work.

2. This thesis does not contain any material which has been published or written by a third party, except accurate referencing and appropriately cited.

3. This thesis also does not contain any material that has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources which we took for help.

Student's Full Name & Signature:

Mohammad Abidur Rahman Chowdhury

15201049

S M Mahsanul Islam Nirjhor

14201031

Md. Nazmus Sabab

16101135

Approval

The thesis titled "BANGLA SPEECH RECOGNITION USING 1D CNN AND LSTM WITH DIFFERENT DIMENSION REDUCTION TECHNIQUES" submitted by

1. S M Mahsanul Islam Nirjhor (14201031)
2. Mohammad Abidur Rahman Chowdhury (15201049)
3. Md.Nazmus Sabab (16101135)

Of Summer, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science & Engineering on August 28, 2019.

Examining Committee:

Supervisor:
(Member)

Jia Uddin, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Mahbubul Alam Majumdar, PhD
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

In the area of machine learning, speech recognition was always a hot topic but as world's 8th most widely spoken language Bangla hasn't got the focus as much as she deserved. This research will be on speech recognition using Bangla language dataset. The training model to recognize consists of 1 dimensional Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM). For feature extraction Mel-frequency Cepstral Coefficient (MFCC) and Mel Spectrogram has been used as the key features for the recognition task. MFCC alone gave an accuracy of 98% for 1d CNN. MFCC when used with LSTM gave an accuracy of 82.35%. Next dimensionality reduction technique was implemented Principal Component Analysis (PCA), Kernel-PCA (k-PCA) and T-distributed Stochastic Neighbor Embedding (t-SNE) transformation on MFCC and Mel Spectrogram for dimensionality reduction technique in a hope to obtain better as efficiency as possible. This is the first attempt to implement these feature reduction methods on Bengali speech. Dimensionality reduction is a technique that is used to reduce large number of features into fewer factors which holds several advantages like reducing time and required storage space. After transformation using PCA a high consistent accuracy was obtained compared to k-PCA and t-SNE transformation (lowest in t-SNE). With PCA applied on MFCC coefficient the accuracy obtained was 94.54% for 1D CNN and 82.35% for LSTM. With t-SNE the accuracy obtained was 49% with 1D CNN and 50% with LSTM. We have also computed the Mel Spectrogram of the audio data after feeding it to model we obtain an accuracy of 90.74% for 1D CNN and 91.6% for LSTM. With k-PCA applied on Mel Spectrogram coefficient the accuracy obtained was 73.95% for 1D CNN and 72.27% for LSTM.

Keywords: MFCC, PCA, Kernel PCA, t-SNE, 1D CNN, RNN, LSTM

Dedicated to

This thesis is dedicated to our honorable Supervisor Dr.Jia Uddin and our beloved parents Our beloved Parents and honorable Supervisor Dr.Jia Uddin for their endless support and patience.

Acknowledgement

First of all, we are very much grateful to the Almighty Allah for keeping us in a good health which was necessary to complete our thesis.

Next,We would like to thank our mentor Dr.Jia Uddin for guiding us and helping us anytime when we needed.

In the end, we have to express our gratitude to our parents and siblings for continuous support and encouragement throughout all these years.

This accomplishment would not have been achieved without their help and support.

Thank you.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CNN Convolutional Neural Network

DNN Deep Neural Network

k-PCA Kernel Principal Component Analysis

MFCC Mel-Frequency Cepstral Coefficient

PCA Principal Component Analysis

RNN Recurrent Neural Network

t-SNE t-Distributed Stochastic Neighbor Embedding

Chapter 1

Introduction

With 261 million speakers, Bangla language ranks 8th biggest language in the world. Literacy rate is 72.89% in Bangladesh and 76.26% in west Bengal. But it is a matter of regret that research of Bangla language detection is very insufficient compare to other languages with less population such French, German. One of the main difficult tasks is to gain dataset. Compare to other major languages, Bangla dataset is hard to come by. In fact, we have to make our own dataset.

Through utilizing different types of feature extraction methods, speech recognition process have been successfully developed. Among them extracting MFCC coefficients from the audio which gives a compress representation of a cosine transform of the real logarithm of the short-term energy spectrum on a Mel frequency scale. Davis and Mermelstein (1980) has done a research by comparing parametric representations for recognizing word from continuously spoken sentences. Linear Frequency Cepstrum, parametric representation based on Mel Frequency Cepstrum, Linear Prediction Spectrum and Linear Prediction Cepstrum were used in this research. They concluded based on the result that MFCC gives better accuracy than other methods. Later on, based on MFCC we also used PCA, t-SNE for dimension reduction in an attempt to improve our accuracy with CNN, LSTM. In Addition to that we computed Mel spectrogram value to apply it in Kernel PCA for the same purpose.

1.1 Motivation

Main thought of this thesis work is to detect words based on MFCC coefficients and Mel spectrogram. Moreover, we have suggested improvements of the detection work. For improvement work, feature reduction techniques PCA, Kernel PCA, t-SNE has been used. The hope of this thesis work was to find a new way to detect Bangla words and using algorithms for feature selection or reduction, enhancing the model performance.

1.2 Objectives

Main objectives of this thesis are summarized as follows:

- Design a system based on MFCC and Mel Spectrogram that can detect Bangla words.
- Use various dimensionality reduction techniques to transform features from higher dimension to lower dimension.
- Design two Neural Network models as the recognizer. One being 1d CNN and another being LSTM variant of RNN.
- Compare efficiency between all these different models.

1.3 Thesis Orientation

The rest of the thesis is organized as follows:

Chapter 2 discusses the literature review.

Chapter 3 discusses background study about different artificial neural network and dimension reduction techniques.

Chapter 4 presents proposed methodology for Bangla speech recognition system using 1D CNN and LSTM with different dimension reduction techniques.

Chapter 5 demonstrates the results for different feature extraction algorithms, neural

network models and dimension reduction techniques used in our proposed model for Bangla speech recognition and also a brief discussion has been done.

Chapter 2

Literature Review

There are various ways to recognize human speech using audio signal processing. Since last decade, different speech recognition models have been developed by using different feature extraction methods and artificial neural network models as well as a lot of research works have been done [16]. Most ongoing models for speech recognition system are mainly focused on reducing the rate of error while recognizing the speeches. Although speech recognition in many other languages is improving day by day, speech recognition for Bangla language could not draw much interest [34]. The reason behind that is the complexity of Bangla spoken words and the scarcity of comprehensive Bangla audio data to train any model. Various speech recognition methods have been proposed since last decade. Thiang, et al. (2011) presented a speech recognition model which uses Linear Predictive Coding (LPC) with Artificial Neural Network (ANN) that controls the movement of mobile robot. Here, the inputs were given by a microphone and the audio feature was extracted by LPC and trained with ANN [17]. Ms.Vimala.C and Dr.V.Radha (2012) proposed a speech recognition method for Tamil language which is speaker independent and for isolated speech only. Implementing Hidden Markov Model (HMM), this system produced 88% accuracy with a data set of 2500 words [20]. Kannan Balakrishnan and Cini Kurian (2012) developed a model for continuous speech recognition system which also uses Hidden Markov Model (HMM) in order to compare among the Context Dependent, Context Independent and Context Dependent tied mod-

els. There were 21 speakers (11 females and 10 males) in the data set [18]. Suma Swamy et al. (2013) presented a more efficient speech recognition system that uses Mel Frequency Cepstral Coefficients (MFCC) and HMM which has 98% accuracy. There were five words, ten times each from 4 different speakers in the data set [25]. Annu Choudhary (2013) introduced a Hindi speech recognition system which detects both isolated and connected words. On the process, Hidden Markov Model Toolkit (HTK) was used and Hindi audio dataset was extracted by Mel Frequency Cepstral Coefficients (MFCC). The accuracy of the system was 95% and 90% for isolated and connected words respectively [22]. Preeti Saini (2013) presented an automatic speech recognition system for Hindi isolated words with Hidden Markov Model Toolkit (HTK). This speech recognition process was done by 10 states in HMM topology and the accuracy was 96.61% [24]. Md. Akkas Ali (2013) developed an automatic speech recognition system for Bengali words using Gaussian Mixture Model (GMM) and Linear Predictive Coding (LPC). In the dataset, there were 100 Bengali words for 1000 times each. The accuracy was 84% [21]. Maya Money Kumar (2014) proposed a speech recognition system that recognizes Malayalam words. The recognition process was done by syllable based segmentation with Hidden Markov Model (HMM) and the features were extracted by Mel Frequency Cepstral Coefficients (MFCC) [26]. Dr. Sanjay Mathur and Jitendra Singh Pokhariya (2014) proposed a speech recognition system for Sanskrit language by HTK. 2 states of HMM and MFCC were used for feature extraction. The accuracy varied from 95.2% to 97.2% [28]. Geeta Nijhawan (2014) presented a real time Hindi speech recognition system using MFCC and QuantizationLinde, Buzo and Gray (VQLBG) algorithm. In order to remove the silence part, Voice Activity Detector (VAC) was used [27]. The best speech recognition performance is observed for a system which uses MFCC and Kernel PCA for dimension reduction technique with 117 features dimensions while Support Vector Machine (SVM) was used as classifier. Here, it was proven that feature reduction technique can decrease verification time dramatically and improve the performance of the system also [29]. Haque and Hussain (2014) devel-

oped a system with 1D Convolutional Neural Network (CNN) which is very fast and lightweight. It uses MFCC features which are suitable to add in the front of the audio processing pipeline. This model achieved high overall accuracy from 93% to 99% in various tasks for the clip length of 0.5-2s [38].

In our work we have chosen MFCC as primary feature set and applied Dimension reduction algorithms to reduce the number of components. Without any dimension reduction, MFCC alone gave an accuracy of 98% for the 1-dimension convolutional neural network or 1d CNN. MFCC when used with LSTM gave an accuracy of 82.35%. With PCA applied on MFCC coefficient the accuracy obtained was 94.54% for 1D CNN and 82.35% for LSTM. With t-SNE the accuracy obtained was 49% with 1D CNN and 50% with LSTM. We have also computed the Mel Spectrogram of the audio data after feeding it to model we obtain an accuracy of 90.74% for 1D CNN and 91.6% for LSTM. With k-PCA applied on Mel Spectrogram coefficient the accuracy obtained was 73.95% for 1D CNN and 72.27% for LSTM.In this study, we discovered that PCA provides surprisingly good result after transformation as compared other feature reduction techniques. Also feature reduction techniques on Bengali Speech audios have never been implemented yet. The data set we used is also developed by us which can be used for other Bengali language research.

Chapter 3

Background

3.1 Artificial Neural Network

Artificial Neural Networks or ANN for short belongs to group of information processing techniques which finds pattern or answers questions from the large amount of data provided [31]. It is an approach that mimics the operation of the brain and is modeled after the structure of the brain [2]. It ANN is a collection of algorithms developed in order to recognize patterns for learning. Learning is progressive which cannot be programmed just like how a child learns new things. The neurons work by modifying the internal parameters to learn to look for relationships for building a mathematical model. They are designed to have processing units that may be hardware or algorithms that work in conjunction.

Deep Neural Network (DNN) is a Neural Network with more than two layers and may contain several hidden layers. The hidden layers help build more a complex model that can detect raw shapes, such as cat or dog faces from image dataset for example [2]. This process of learning is called Deep Learning. Essentially deep learning is a machine learning technique which trains computers to do what comes naturally to humans. The term "deep" generally refers to the number of hidden layers in the deep neural network. A Deep learning model can achieve state-of-the-art accuracy, which sometimes exceeds human-level performance. To train a deep neural network, a large set of data is required. Unlike most traditional machine-learning algorithm,

Deep-learning networks can perform automatic feature extraction without human intervention which requires extensive computing power.

3.1.1 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a specific type of neural network which can work with one two or even three dimensional data [33]. CNN consists of convolutional layers where a convolution process happens and hence the name convolutional. Convolution is a linear operation which takes place through multiplication of the data matrix and a kernel where the size of the kernel determines the type of CNN [33]. The major benefit of CNN is that it is computationally effective with it performing pooling layers and parameter sharing [33].

3.1.2 Recurrent Neural Network

A Recurrent Neural Network (RNN) is a specific type of artificial neural network that can feed information from the nodes of previous layer [14]. This makes it popular for their use in tasks such as speech recognition, handwriting recognition and natural language processing. RNNs are used to identify a data's sequential characteristics and to predict the next likely scenario with the help of the patterns [23].

RNNs use feedback loops where output from previous step are fed as input to current step in order to process a sequence of data that provides the final output, which can also be a sequence of data [23]. These feedback loops allow information to persist as memory which remembers all information about a sequence and about what has been calculated so far [23].

3.2 Dimensionality Reduction

Dimensionality Reduction is the process of transforming feature set from higher to dimensional space to less dimensional space where it still retains its meaningfulness

[9]. There are many techniques and some popular ones are Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA), Generalized Discriminant Analysis (GDA) and T-distributed Stochastic Neighbor Embedding (t-SNE). Kernel Principle Component Analysis (k-PCA) is a variant of PCA that uses kernel techniques which is better at dealing with complicated spatial relationships than PCA.

3.2.1 Principle Component Analysis (PCA)

Principal Component Analysis (PCA) is a linear transformation technique that produces linear uncorrelated variables called principal components through statistical procedure using an orthogonal transformation [13]. The main objective of PCA is to transforms vectors from higher dimension to lower dimension such that the variance of data in that dimension is higher. This ensures the variability is as maximum as possible and the data points retain the significance of being part of the feature set.

3.2.2 Kernel PCA

Kernel PCA is an extension of PCA which is a non-linear technique and uses kernel methods [12]. It accomplishes non-linear dimensionality reduction through the use of kernels while retaining the computational benefits of PCA [12]. K-PCA performs PCA in a new space.

3.2.3 T-Distributed Stochastic Neighboring Entities (t-SNE)

t-Distributed Stochastic Neighbor Embedding (t-SNE) is another technique for dimensionality reduction by giving each data points in a high dimensional data a location in two- or three-dimensional map [11]. Where PCA is a mathematical technique, t-SNE follows the probability technique. The algorithms calculate the probability of the similarity of points in high-dimensional space as well as the probability of the similarity of points in the corresponding low-dimensional space [11]. The similarity of points is calculated as the conditional probability of two points in

proportion to their probability density. Then it minimizes the difference between these conditional probabilities in higher-dimensional and lower-dimensional space for a perfect representation of data points in lower-dimensional space [35]. To calculate the minimization of total difference of conditional probability, t-SNE minimizes the sum of Kullback-Leibler divergence of overall data points using a gradient descent method [35].

Chapter 4

Proposed Method

Most of the early research done on Bengali Language focused on building a recognizer with the number of features that could be obtained through the feature extraction techniques of their audio dataset. However, in our model we have considered using feature transformation techniques to reduce the number of features significantly without hampering accuracy by great margin. Different techniques perform differently for which we will provide a comparative analysis and also compare results against results without using any feature reduction.

The features we have chosen to work on are Mel Frequency Cepstral Coefficient and Mel Spectrogram. These features are very commonly extracted and used for tasks involving speech recognition or emotion recognition or anything that involves speech or audio data. For feature reduction we have used principal component analysis (PCA), kernel principal component analysis(k-PCA) and t-distributed stochastic neighbor embedding (t-SNE). The advantages of feature reduction is that it reduces space required as that data is more compressed and also reduce time complexity of the learning the neural network has to learn from less features. Also if there are any redundant features originally, the transformation process can reduce to completely eliminate their influence. Our methodology is illustrated in the Figure 4.1.

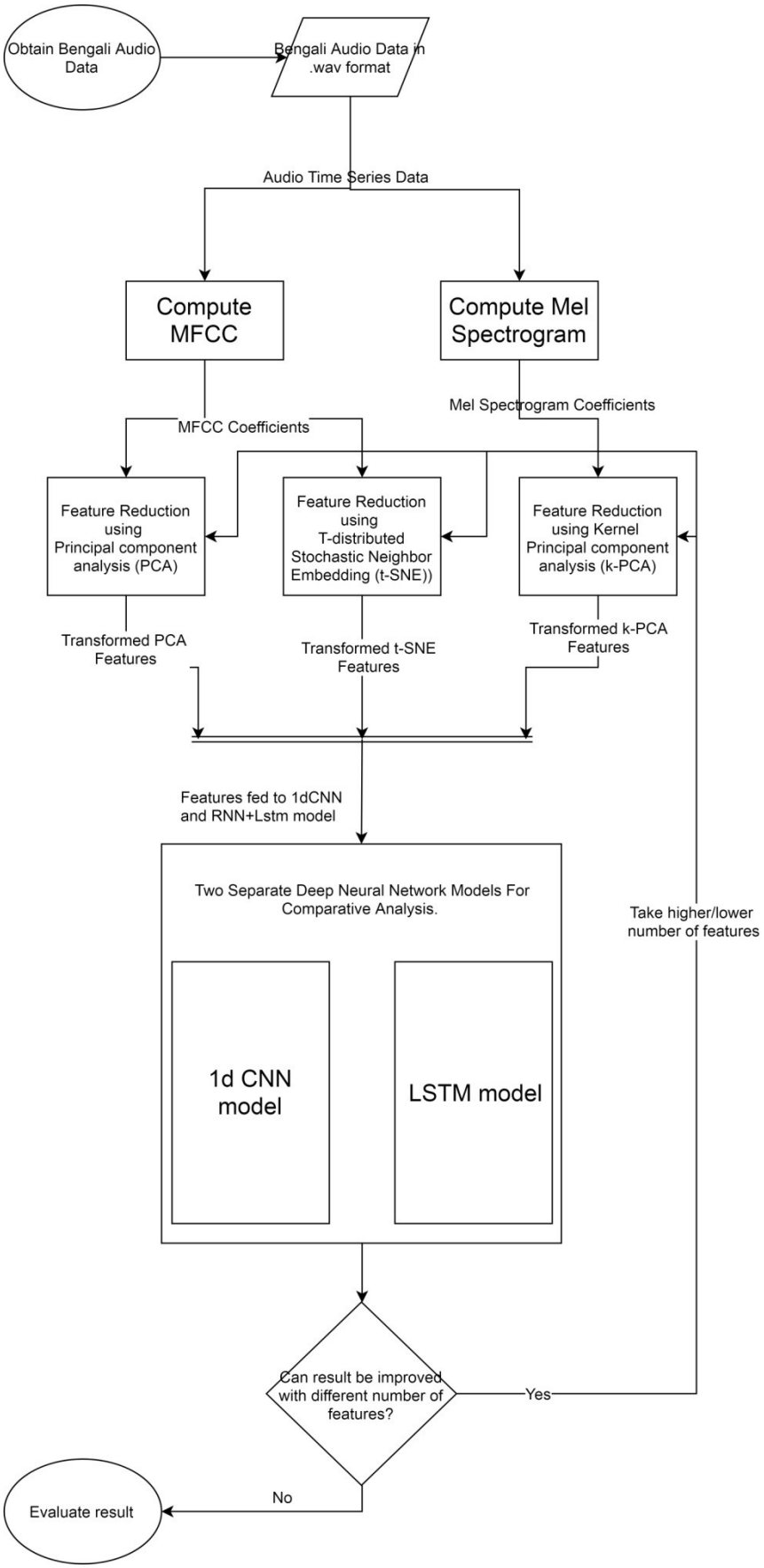

Figure 4.1: Proposed Methodology

4.1 DATA DESCRIPTION

In our thesis, most challenging part was data collection. Although Bangla is one of the most spoken languages, there have been very few data available. We had to make our own data for research purpose. First, we collected audio voice recordings from some random Bangla speakers. Later on, we have trimmed those audio samples eliminating silent and other noisy part from the collected data with Audacity software. Then, we have converted those audio files to WAV format. Multiple speakers are introduced in our dataset. We tried to minimize noise in our data as much as possible.

4.1.1 Dataset Properties

For our dataset we chose two Bangla words which are "Three" and "Seven" in Bangla. Each words were put inside folder named "Seven", which contains 284 wav files and another folder named "Three" containing 307 wav files. We have six different speakers in our data.

A single speaker uttering the same word "Seven" on two different occasions may also sound different which is illustrated by Figure 4.2 and Figure 4.3 below.

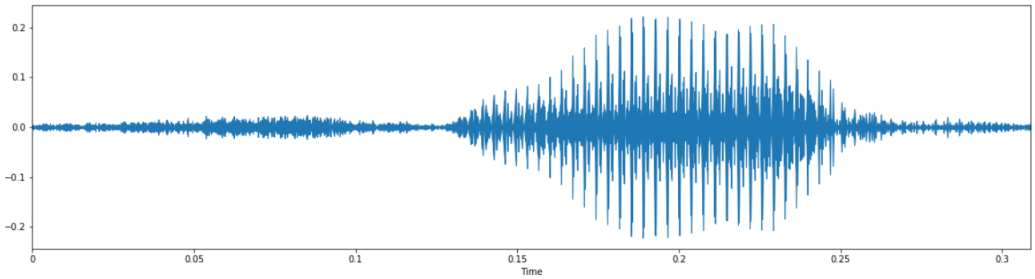

Figure 4.2: "Seven" for Speaker 1

Figure 4.4 and 4.5 illustrates the difference between the utterances of the same word by different speakers.

The Figure 4.6, 4.7 and 4.8 below represents waveform of utterances of the word "Three" all by different speaker.

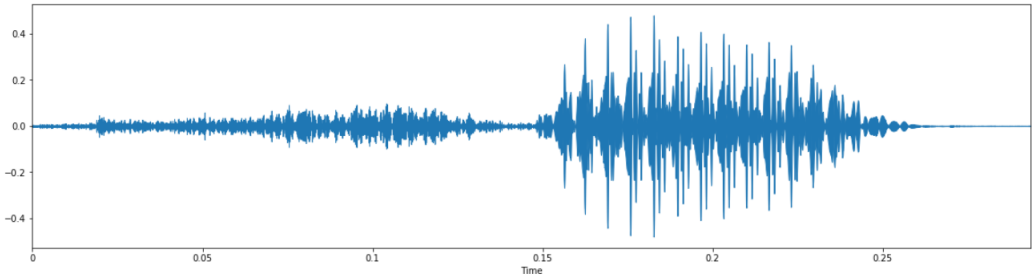

Figure 4.3: "Seven" for Speaker 1

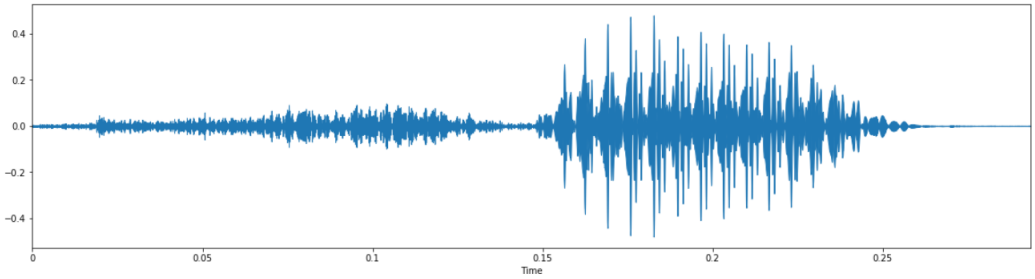

Figure 4.4: "Seven" for Speaker 3

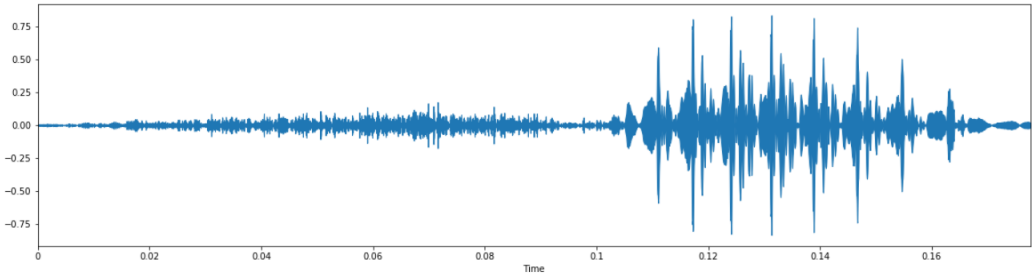

Figure 4.5: "Seven" for Speaker 4

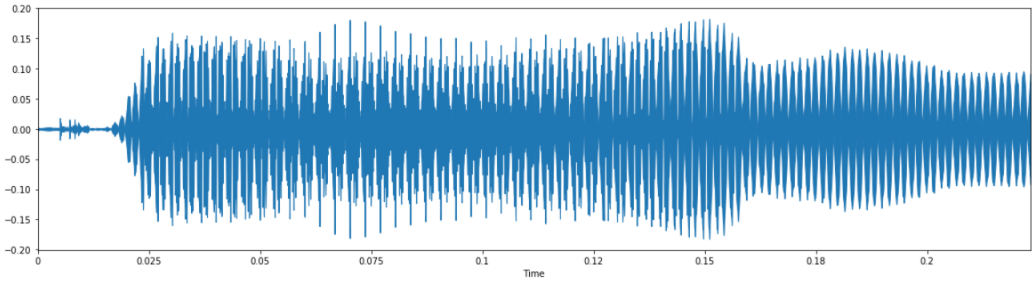

Figure 4.6: "Three" for Speaker 1

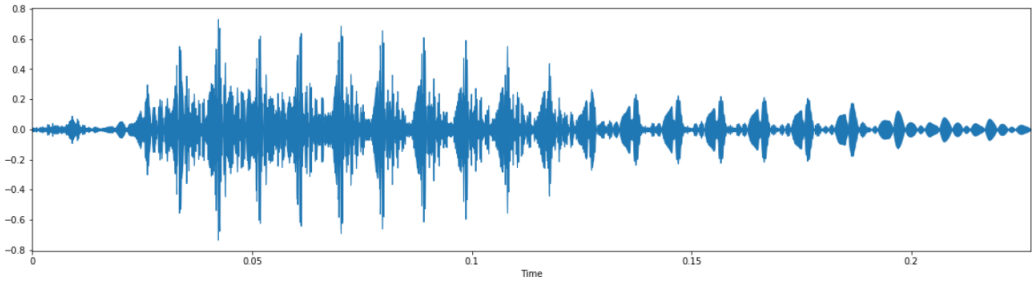

Figure 4.8: "Three" for Speaker 3

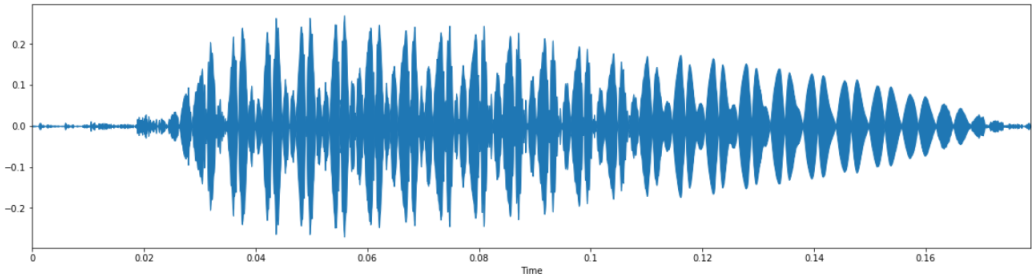

Figure 4.7: "Three" for Speaker 2

4.2 FEATURE EXTRACTION

Feature Extraction is a process of obtaining information from data that can be used to classify the dataset based on it. Features can be treated as attributes of the data, synonymous to common characteristics of all birds vs. mammals. There is various feature extraction techniques that are used based on its application. For signal data such as audio, features can be MFCC values which mimic human auditory system as required for audio. In the case of emotion recognition, these MFCC values are taken and their mean, variance, standard deviation, mode etc. are also being calculated. Not only MFCC are being used for audio data for the purpose of speech recognition, other features such as time series data and Mel spectrogram of the audio sample is also used. There are other techniques applied on these features to get new features and sometimes of different dimensions that can also be called as features.

The features we chose for our model are MFCC and Mel spectrogram and after computing these features we have applied dimensionality reduction algorithm like PCA, t-SNE, Kernel PCA that give us new feature values.

Both MFCC and Mel spectrogram give us coefficient values for the time window. For each window we have taken 128 MFCC coefficients and 128 Mel Spectrogram coefficients. We then reduced the number of coefficients to six, three, six respectively for PCA, t-SNE and Kernel PCA building three different models. For comparison we have made another model but only this time we have used raw MFCC values for the model.

4.2.1 MFCC

MFCC are feature of Audio signals very popular for Automatic Speech Recognition and Speech based Emotion Detection models. MFCC is the abbreviation for Mel Frequency Cepstral Coefficient. The concept behind MFCC is that it converts audio being represented as a function of time to representing it as a function of frequency. Moreover, during conversion Mel filter is used that mimics the Human cochlea. This ensures many uses of MFCC. L. Muda [15] built a voice recognition algorithm where MFCC features were extracted and used. F Zheng [4] made a comparative analysis of the factors that may affect the performance of MFCC. The factors he stated were, the number of filters, the shape of filters, the way in which filters are spaced and also the way in which the power spectrum is warped. For speaker and voice recognition MFCC features were used [8] [7]. MFCC features were used in research works for detection of emotionally abused women [36]. MFCC features are also being in used in Music industry where things like karaoke or vocal file generation, identifying instruments and many other tasks performed. M. Muller [10] used MFCC features for music information retrieval for tasks like genre classification.

A major advantage of MFC over Cepstral is that it mimics the Human Auditory System much better and thus much more preferable in Speech Recognition Systems [19]. The idea is that, since human auditory system is excellent at picking up human speech why not build an artificial system mimicking it. The reason why MFCC is better is due the frequency bands being equally spaced vs in Cepstral Coefficient the frequency bands are linearly spaced [19].

In our work we have utilized a Python package called librosa. Which has a sub module called feature.mfcc where get the MFCCs of an audio file. By default the function returns 20 coefficients for each window, however we have taken the maximum number of 128 coefficients. We have kept the number of time frames to 21. We chose 21 because that is the maximum number of frames any of the sample might have and we did not want to chop off any coefficients. As for the samples which have lower number of frames, we have zero padded to complete 21 frames. Thus our train and test data is in three dimensions with a shape of (number of samples, 128, 21). In our work, we have utilized a Python package called librosa. Librosa has a sub module called feature.mfcc where we get the MFCC's of an audio file. By default the function returns 20 coefficients for each window, however we have taken the maximum number of 128 coefficients. We have kept the number of time frames to 28. We chose 28 because that is the maximum number of frames any of the sample might have and we did not want to chop off any coefficients. As for the samples which have lower number of frames, we have zero padded to complete 28 frames. Thus our train and test data is in three dimensions with a shape of (number of samples, 128, 28).

4.2.2 Mel Spectrogram

Mel spectrogram is a lower level acoustic representation of audio signal [39]. Mel Spectrogram is another feature set we have used. Like MFCC, Mel spectrogram holds the advantage of using a Mel scale which biologically inspired to the human auditory system.

In our model we used python's librosa package has functions that will calculate mel frequency of an audio file directly by taking the audio file time series and frequency in hertz.

We have taken128 Mel coefficient of our audio data in order for the transformation to lower feature set be as smooth as possible. The audio is divided into 28 windows of the time series to keep all the audio files of equal length. This is crucial because

the CNN recognizer model build later cannot work with variable length dataset.

4.2.3 PCA

The MFCC coefficients are transformed from 128 values per window to 6 values per window. This significantly reduce the number of data from 3584 (128x28) to 168 (6x28). For implementing PCA in our code we used python's skicitlearn library. The data has to be scaled for normalization in order for kPCA to perform its best.

4.2.4 k-PCA

Here we chose Mel Spectrogram coefficients because it does not generate any negative eigenvalues when applied on large sets. Here 128 mel spectrogram per window transformed to 6 values per window. This significantly reduce the number of data from 3584 (128x28) to 168 (6x28). For implementing k-PCA in our code we used python's skicitlearn library. The data has to be scaled for normalization in order for k-PCA to perform its best.

4.2.5 t-SNE

We applied t-SNE on MFCC coefficients and converted 128 feature dimensions into 3 dimensional space. In python scikitlearn also has a module for implementing t-SNE. Thus, after transformation, we get a number of data from 3584 (128x28) to 84 (3x28).

Chapter 5

RESULTS AND DISCUSSION

This chapter will discuss about the result obtained from recognition model on the dataset. The various approaches used to extract features such as MFCC and Mel Spectrogram and dimensionality reduction techniques which are PCA, k-PCA and t-SNE applied to those features. Finally, a comparative analysis is presented based on all the approaches discussed previously.

5.1 Classification Results

5.1.1 MFCC with 1D CNN

Our 128 MFCC coefficients is fed into a 1d convolutional layer with 64 neurons with a kernel size of 3 the output is entered to another 1d convolutional layer also comprising of 64 neurons. Both of these layers have a kernel size of three. The layer follows is a max pooling layer of size 3 which is followed by a 1-dimensional global average pooling. The output is fed into a dropout layer and final layer is a dense layer with the activation function set to sigmoid.

The loss function of the overall model is binary_crossentropy and optimizing function is rmsprop. Our model obtained an accuracy of 98.81% in 50 runs which is illustrated in Figure 5.1. Figure 5.2 shows the loss function if this model.

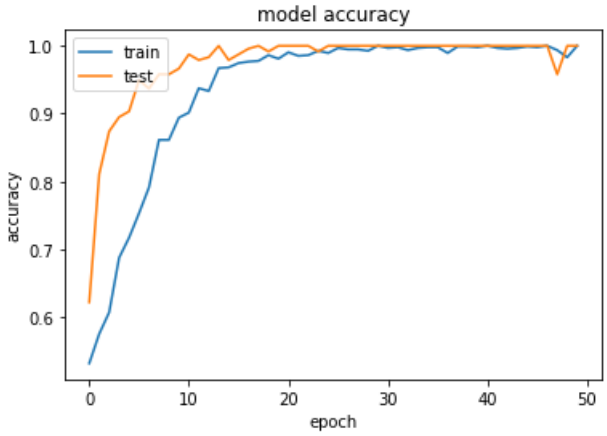

Figure 5.1: Accuracy of 1D CNN using MFCC

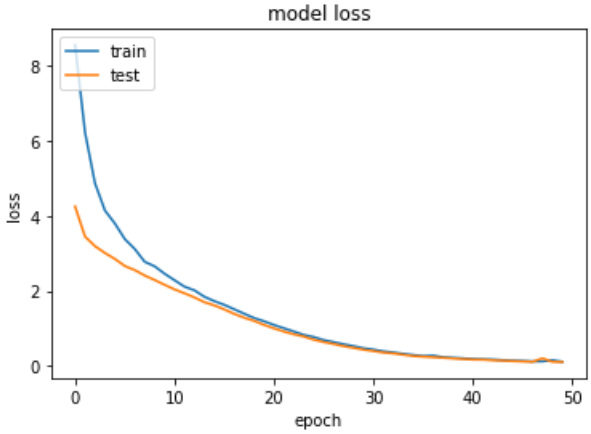

Figure 5.2: Loss function of 1D CNN using MFCC

5.1.2 MFCC with LSTM

Our Recurrent LSTM also has two LSTM layers with 128 neutrons in the first followed by a dropout layer and recurrent dropout layer. And another LSTM layer of 64 neurons also followed by a dropout layer and recurrent dropout layer. For our regularizer function we have used kernel, recurrent and bias regularizers with various values. The final layer is a dense layer configured with the activation function of softmax. The loss function here for this model is categorical_crossentropy and optimizing algorithm we used is adamax. Our model obtained an accuracy of 96.49% for 50 epochs. Figure 5.3 represents accuracy and loss graph.

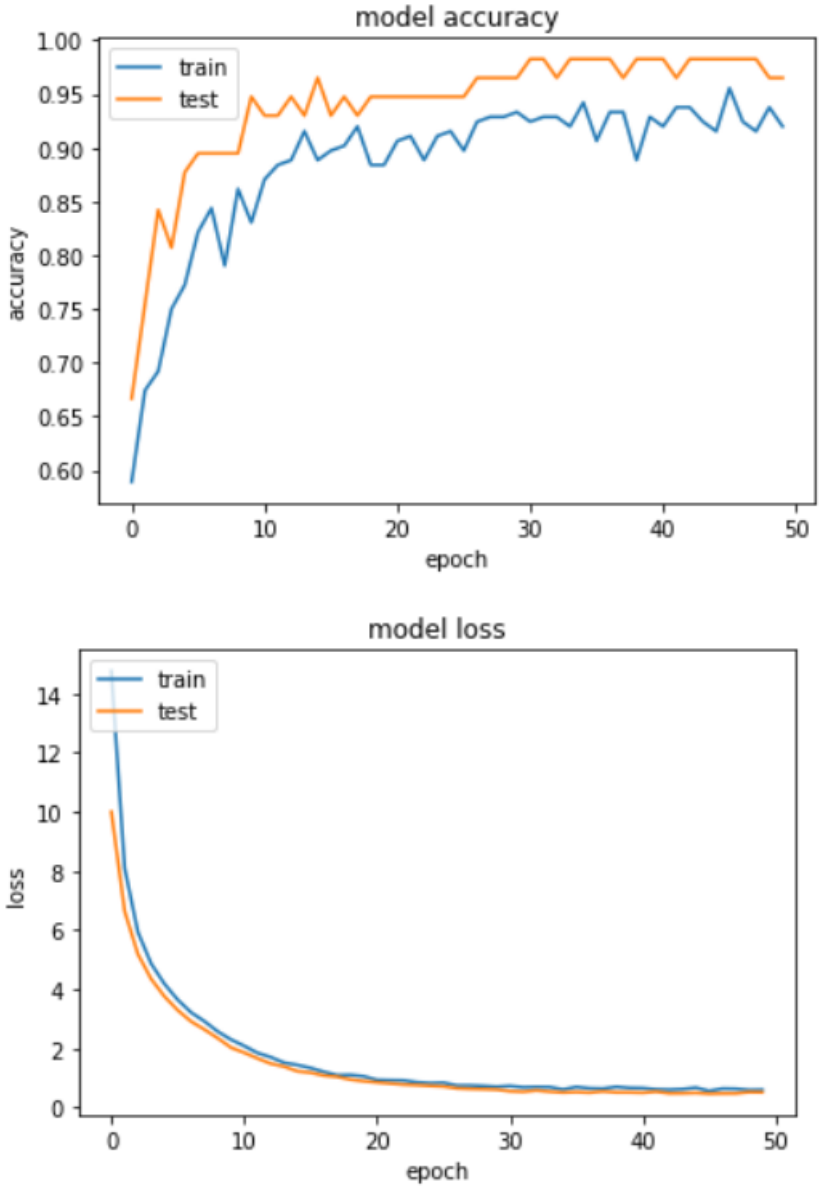

Figure 5.3: Accuracy and Loss function of LSTM using MFCC

5.1.3 PCA with 1D CNN

For our project we have used Principal Component Analysis as a Dimensionality Reduction Technique. We fed our MFCC features into PCA technique and reduced the component number from 128 to just 6. Then these PCA reduced features is fed into 1d CNN.

For our Neural Network model, the first layer is a 1d convolutional layer with 64 neurons followed by another 1d convolutional layer also comprising of 64 neurons. Both of these layers have a kernel size of three. The output from each layer has to go through an activation layer of the function of relu also known as Rectified Linear Unit. This particular activation has the capacity to learn much faster than 'tanh or sigmoid functions.

$$f(x) = m(0, x)$$

This is the general formula for relu, where the function return x when positive otherwise, it returns 0.

Next, we added a 1d Max Pooling layer with the kernel size of 3 followed by a 1d Global Average Pooling layer. Both of these layers do not have any learnable parameters and their function is merely to reduce the output significantly. The output from Global Average Pooling is fed into Dropout layer. The purpose of these layers is to switch off certain nodes at random to reduce the complexity of the network and thus improve the network's validation accuracy. The final layer is the dense layer, which is a fully connected layer. Fully connected meaning each neuron in the dense layer receives input provided by the previous layer. The output of this layer the general output that are looking for. In our case whether the audio file says "Three" and "Seven".

The loss function that is used by our model is binary_crossentropy and optimizing algorithm we used is rmsprop.

Our model obtained an accuracy of 94.12% for the test dates after 50 runs which is

depicted in Figure 5.4 and 5.5.

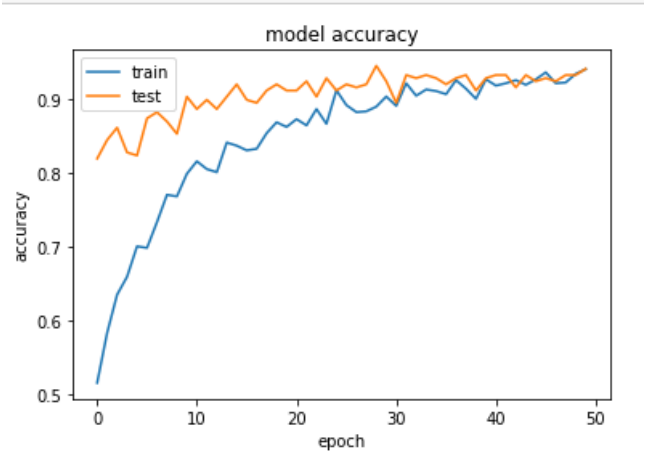

Figure 5.4: Accuracy function of 1D CNN using PCA

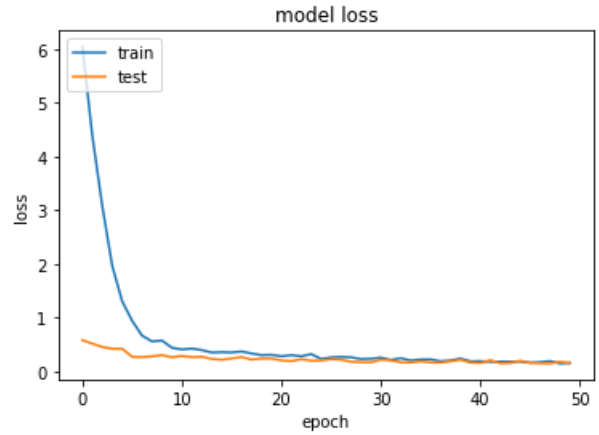

Figure 5.5: Loss function of 1D CNN using PCA

5.1.4 PCA with LSTM

Our Recurrent LSTM also has two LSTM layers with 32 neutrons in the first and 32 neurons in the second layer. The successive output from these layers goes through a dropout out layer with and the output which is then passed through recurrent dropout layer. For our regularizer function we have used kernel, recurrent and bias regularizers with various values. The final layer is a dense layer configured with the activation function of softmax. The loss function here for this model is categorical_crossentropy and optimizing algorithm we used is adamax. Our model obtained an accuracy of 82.35% after 50 epochs as illustrated in Figure 5.6 and 5.7.

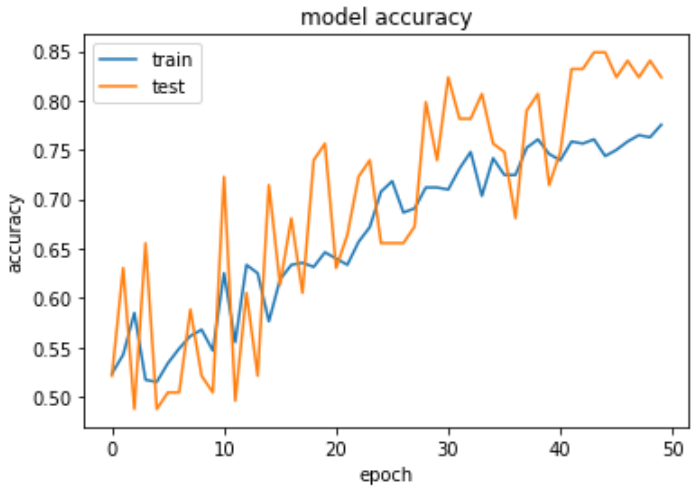

Figure 5.6: Accuracy function of LSTM using PCA

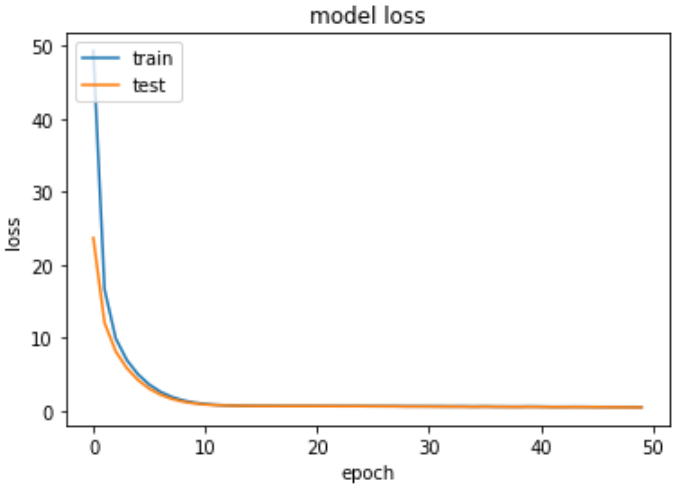

Figure 5.7: Loss function of LSTM using PCA

5.1.5 t-SNE with 1D CNN

t-SNE used for reducing MFCC features gave very poor 1-dimensional convolutional neural network with an accuracy of 49% only. Here we have experimented with building network model of various depth however, an ideal network structure was never seemed to be found. We have tried both shallow and deep network model but the accuracy remained at around 49% -50% with high loss when accuracy is 50%. Thus, we have concluded the features provided by t-SNE dimensionality reduction contains very poor information for classification.

The shallow network we attempted was a 4-layer CNN with neuron numbers 64, 64, 128 and 256 respectively. There is a 1d max pooling layer after the second layer

and each of the CNN layers are followed by an activation layer of the function relu. Next is a global average pooling layer followed by a dropout layer. Finally, a dense layer gives the output.

The accuracy function is seen in Figure 5.8.

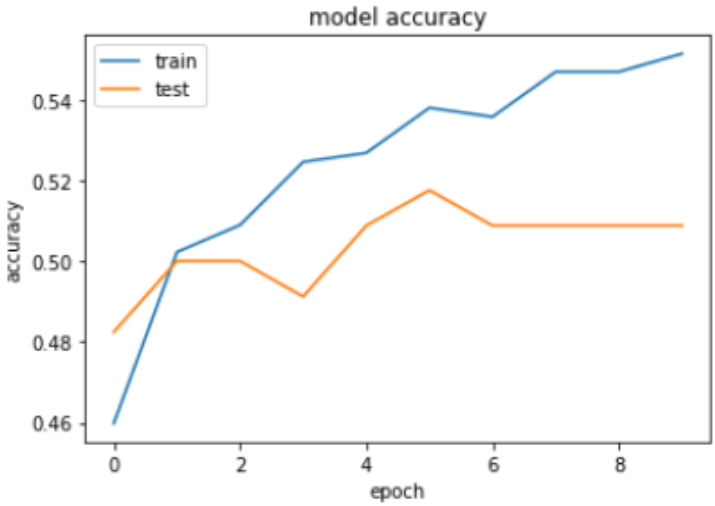

Figure 5.8: Accuracy 1D CNN using t-SNE

5.1.6 t-SNE with LSTM

Like the CNN part our LSTM also performed poorly. This is the model we settled with after deeper models not giving any better accuracy. We have 2 LSTM layers of 64 and 32 units successively. We have used dropout and recurrent dropout layers for each model. The final layer is a dense layer.

As we can see in fig 18, the model provided 50% accuracy with no room for improvement even when increasing epoch. This system also had a very high loss. Figure 5.9 and 5.10 illustrates the accuracy and loss.

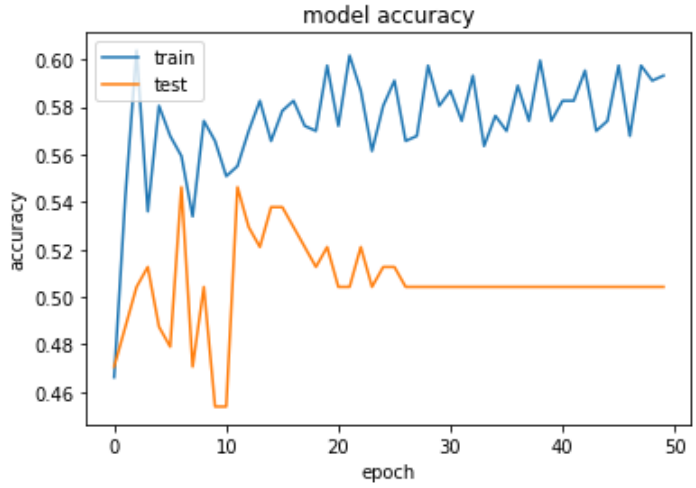

Figure 5.9: Accuracy function of LSTM using t-SNE

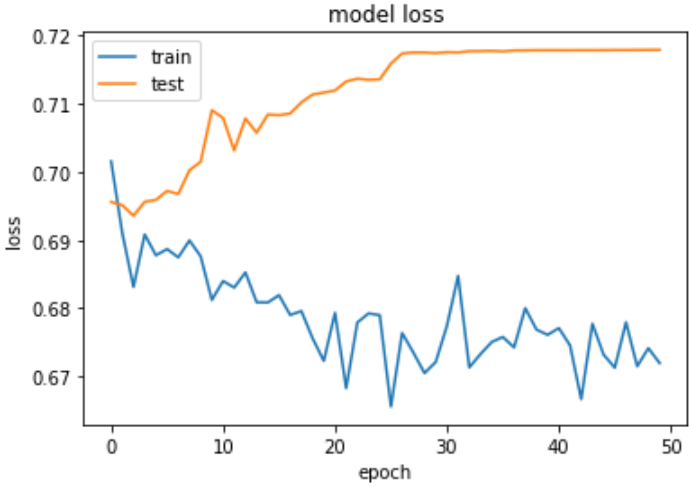

Figure 5.10: Loss function of LSTM using t-SNE

5.1.7 Kernel-PCA with 1D CNN

For our project we have used Kernel Principal Component Analysis as a Dimensionality Reduction Technique. We fed our Mel spectrogram features into K-PCA technique and reduced the component number from 128 to just 6. The reason we used mel-spectrogram over MFCC here because the matrix of the MFCC produces negative eigenvalues, which when the square root is computed produces nun values. For our Neural Network model, the first layer is a 1d convolutional layer with 64 neurons with an activation layer following configured to relu. This is Followed by

another 1d convolutional layer also comprising of 64 neurons also followed by an activation layer configured to relu. Both of these convolutional layers have a kernel size of three. After this there is a 1d max pooling layer followed by 1d global average pooling layer .The output from Global Average Pooling is fed into Dropout layer. The purpose of these layers is to switch off certain nodes at random to reduce the complexity of the network and thus improve the network's validation accuracy. The final layer is a fully connected dense layer.

The loss function that is used by our model is "binary_crossentropy" and optimizing algorithm we used is "rmsprop". Our model obtained an accuracy of 73.95% for the test dates after 50 runs as illustrated in Figure 5.11 and 5.12.

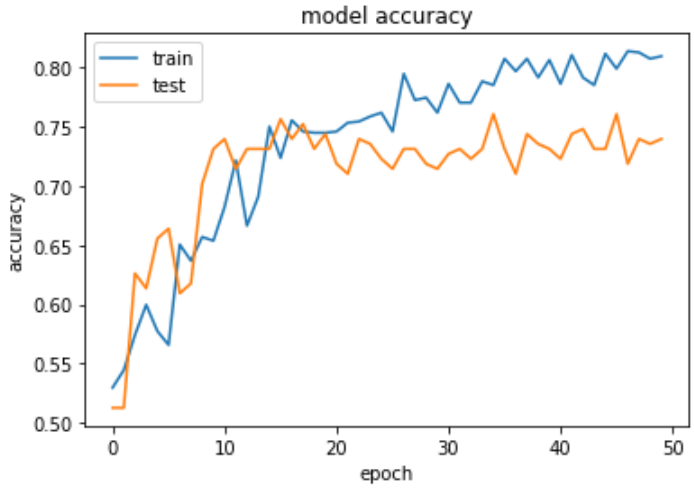

Figure 5.11: Accuracy function of 1D CNN using Kernel PCA

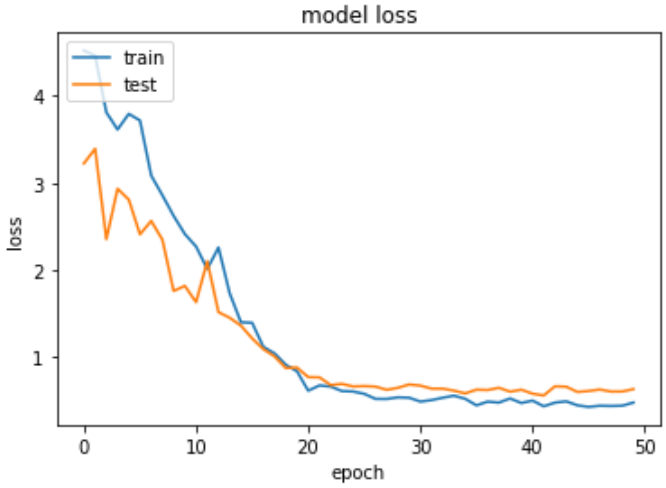

Figure 5.12: Loss function of D CNN using Kernel PCA

5.1.8 Kernel-PCA with LSTM

Our Recurrent LSTM also has two LSTM layers with 64 neutrons in the first and 32 neurons in the second layer. The successive output from these layers goes through a dropout out layer with and the output which is then passed through recurrent dropout layer. For our regularizer function we have used kernel, recurrent and bias regularizers with various values. The final layer is a dense layer configured with the activation function of softmax. The loss function here for this model is categorical_crossentropy and optimizing algorithm we used is adamax. Our model obtained an accuracy of 72.27% for the test dates after 50 runs as shown in Figure 5.13 and 5.14.

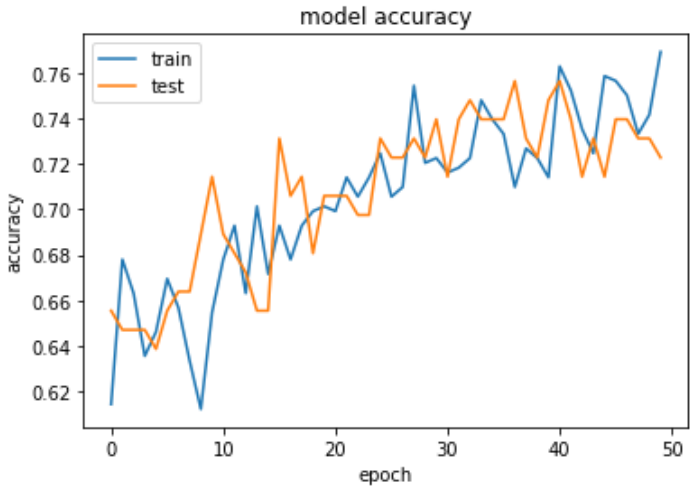

Figure 5.13: Accuracy function of LSTM using Kernel PCA

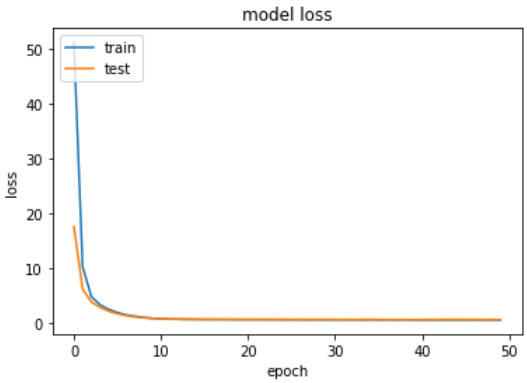

Figure 5.14: Loss function of LSTM using Kernel PCA

5.2 Result

We have compiled our result based on the accuracy of the model vs number of features used. For the CNN model using MFCC as features gave us a very high accuracy for all number of features. As illustrated in Figure 5.15.

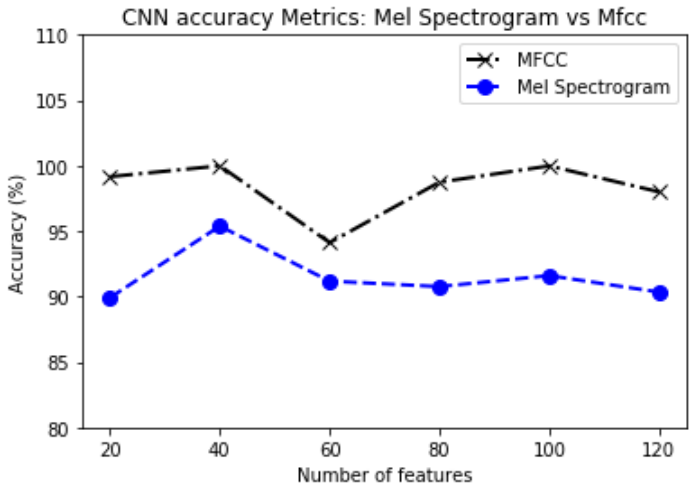

Figure 5.15: CNN accuracy for Mel Spectrogram vs MFCC

For the LSTM model Mel Spectrogram features provided average higher accuracy for any number of features set as illustrated in Figure 5.16.

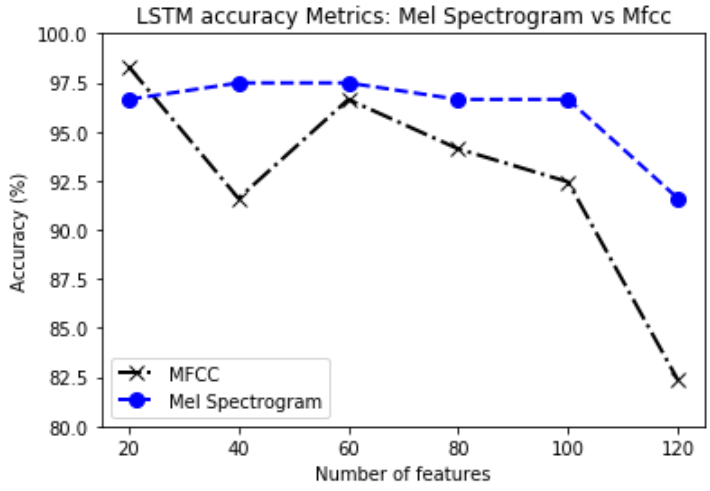

Figure 5.16: LSTM accuracy for Mel Spectrogram vs MFCC

After applying feature reduction algorithm and training it on 1d CNN model, we can see using PCA coefficients holds a very accuracy among all the other techniques which does not change a lot by changing the number of features being transformed.

T-SNE transformation gave the least accuracy around 49%-50%. And k-PCA gave a respectable accuracy somewhere around 72%-75% for the features illustrated in the Figure 5.17.

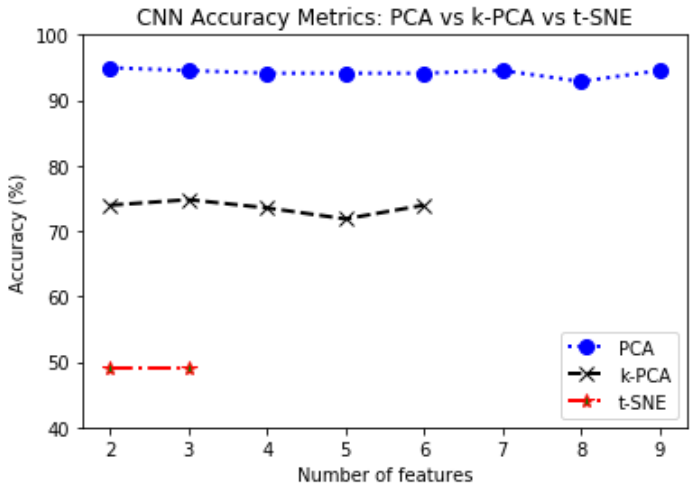

Figure 5.17: CNN Accuracy Metrics for PCA vs k-PCA vs t-SNE

After applying feature reduction algorithm and training it on LSTM model, we can see using PCA coefficients again holds a very accuracy among all the other techniques which changes slightly when changing the number of features being transformed to. PCA features give their peak accuracy with LSTM model when transformed to 5 features per window. T-SNE transformation again gave the least accuracy around 45%-50%. And k-PCA gave peak accuracy when transformed to 4 features per window as illustrated in the Figure 5.18.

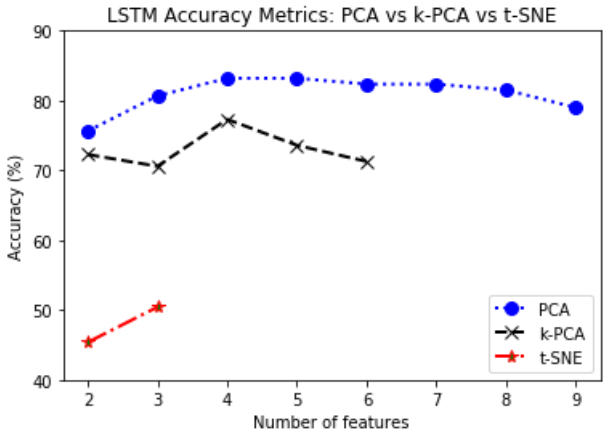

Figure 5.18: LSTM Accuracy Metrics for PCA vs k-PCA vs t-SNE

5.3 Discussion

There were a few but significant issues that we encountered in this project. The primary issue was over fitting. Slight over fitting may be ignored if the dataset sample is too high. However, in our case it affects the result significantly. The dataset brings us to the second issue and that is scarcity of it. There are very few speech dataset corpuses for Bengali Speech and thus we had to build one ourselves. The limited dataset we built was increased to a decent amount by Audio Augmentation. These issues are discussed below and how was the problems tackled or at least mitigated.

5.3.1 Reducing Over fitting

Over fitting occurs when the model performs really well at classifying or predicting data from the training sample but performs poorly on testing sample [5]. How can we know if our model is over fit? Well, during the training process the model calculates and shows the matrices for the training and validation data showing us accuracy and loss values for both training and validation data. If the matrices of validation set are considerably worse than the training set, then we can say the model has over fit.

Our model initially had high over fitting and we have subsequently applied a few techniques to mitigate the issue. We have used Regularization layers, Dropout layers and Audio Augmentation to reduce over fitting.

Regularization

It is a technique that helps reduce over fitting or reduce variance in our network. As mentioned earlier, complex algorithms when used for training can give amazing results on the training sample but May not do so on the testing sample. Also discussed about the notion that an ideal training algorithm has to be somewhere between a simple algorithm where it oversimplifies the network vs. a complex algorithm where it starts to incorporate noise into the network [37]. Here the idea is that certain complexity may make our model to generalize poorly despite performing well on the training data. So the model starts to tone down the complexity of the algorithm

based on the loss function if the network is performing too well on training data. As a result, we are trading in some of the ability of the model to classify training data well for the model to better generalize on unseen data. In a Neural Network, Regularization technique works by turning off certain nodes thus toning down or eliminating their contribution. Sometimes in Machine Learning the best models are the large models which have restrictions in using its full potential i.e they are regularized.

The most common Regularization techniques are L1 and L2. These update the general cost function by adding another term known as the regularization term [6].

Cost function = Loss function + Regularization term

• L1 Regularization In L2 Regularization, the Regularize term is the summation of the squares of all the weights. L2 regularization forces the weights to decay towards but not exactly zero.

Here the lambda value is the hyper parameter that we tune to obtain an optimum result.

In our model we have used Regularization layer in our LSTM network. We chose L2 Regularization for its advantages discussed already. We implemented low regularization coefficients on the initial layers with high coefficient for later layers.

Dropout

This is also a technique to reduce over fitting. Also a kind of Regularization technique, It also produces respectable results and is used often to reduce over fitting. This technique is very popular in the field of deep learning. As the name suggests it drops nodes to reduce the complexity of the model [30]. The way the nodes are dropped is random in nature and the probability of how many nodes to drop can be set in within a hyper parameter.

Dropout layer has been used in our model, with high dropout coefficients used where over fitting is high.

Early Stopping

It is a cross-validation technique where when the accuracy and loss metrics for the

validation set gets worse than the training set we stop the process.

Bias vs. Variance

• Bias

The algorithm we use to train our model may not be able to capture the true relationship between the data and the label. The higher the bias, the more oversimplified the model gets [1]. To give an example, linear function will always figure out a linear relationship if used in a model where the data and label have a nonlinear correlation. Squiggly Lines on the other hand tries to capture as many data points as it can in relation to its label and fits all the data much better. This means squiggly lines have a lower bias.

• Variance

Any algorithm let that be Squiggly Lines or Linear Function, however it may perform on training sample may perform differently on testing sample. If the algorithm fits the testing data poorly then it has very high variability [3]. Because, the sums of distance between the dataset and algorithms are much higher. If an algorithm fits the training set really well but not the testing set then we say model is over fit. An ideal algorithm as low bias fits well on the training set and low variance fits well on the testing set. To do that, we need to look for the ideal spot between a simple and complex algorithm. Some techniques are Regularization, Boosting and Bagging.

Audio Augmentation

Audio augmentation is a data augmentation method where the data available is modified to create new data and increase the quantity of the data reduce over fitting and make model more robust [32]. The Augmented data should be such that it still retains the usefulness for the system that it is training on. We need data augmentation because data are very hard to come by. So, to ensure we make the most use of the data we have, we change the data ever so slightly that it is counted as a unique sample without making it too different.

Our audio data example is speaker saying the word "Tin" in Bengali which means "Three" in English.

Figure 5.19 is the visualization of the audio file:

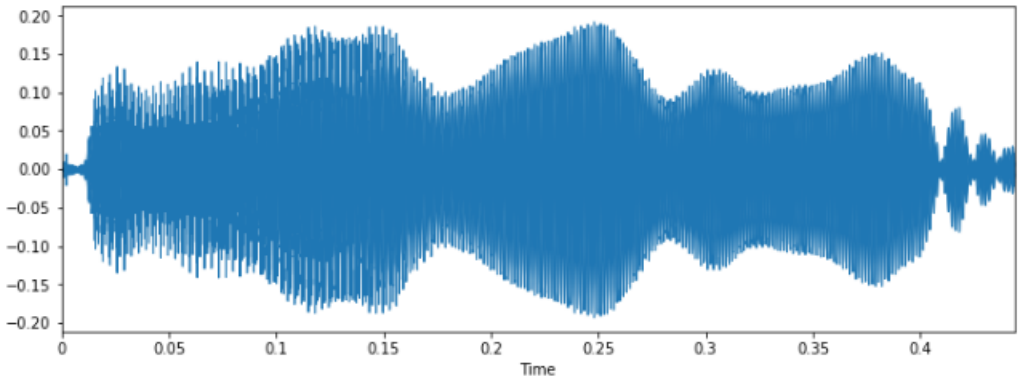

Figure 5.19: Normal "Three" wav file

- Adding Noise

Using numpy it simply adds some random values into data. The random value is generated through numpy's random number generator function and multiplied with a factor. The result is then added to the original data. In Figure 5.20 we can see how adding noise added additional spikes to the waveform. The data is still recognizable as "Tin".

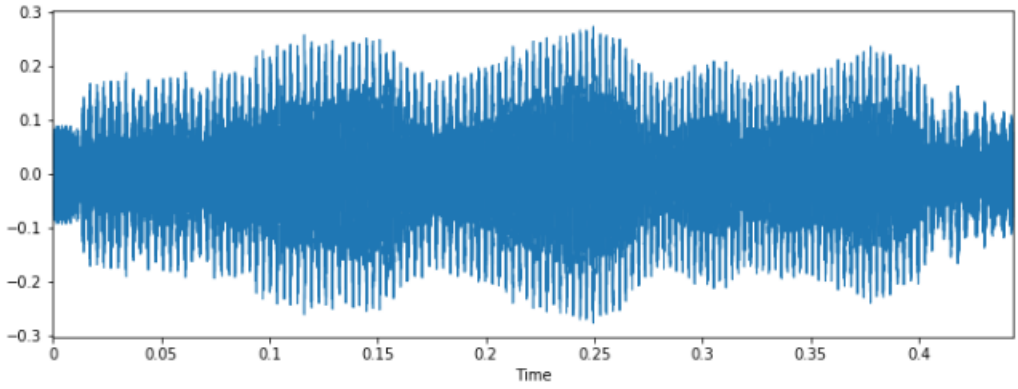

Figure 5.20: Noise added "Three" wav file

- Shifting left or right

The idea of shifting time is very simple. It just shifts audio to left or right with a random time unit, half a second or quarter of a second for example. Since numpy's roll functions re-introduces the falling values to the beginning it is a good idea to zero pad at the end as illustrated in Figure 5.21.

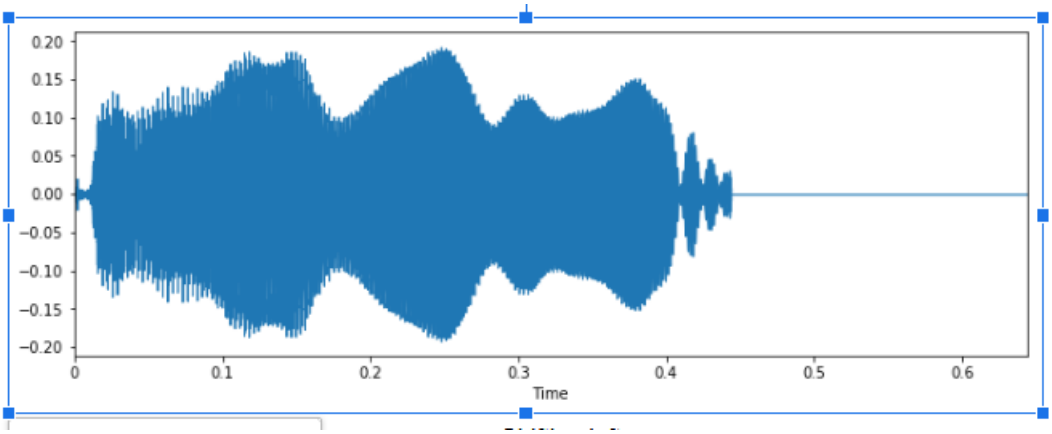

Figure 5.21: Left shift "Three" wav file

- Pitch Shifting

Pitch corresponds to the frequency of the sound wave. Thus a higher frequency means higher pitch and vice versa. Here the pitch has been increased by 1.5 times to give the output below. Pitch-shifting keeps the time duration of the signal same. In Figure 5.22 we can see how shape changes as corresponding to figure 5.19 without any change in the length of the signal.

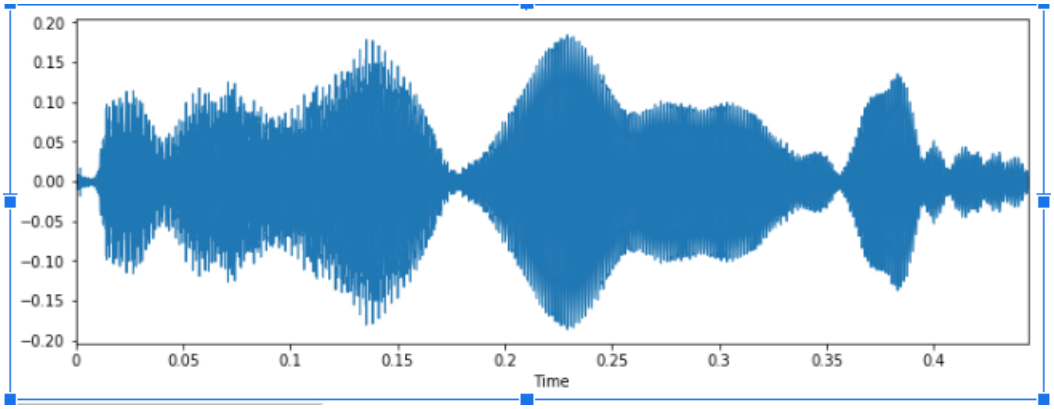

Figure 5.22: Pitch shift "Three" wav file

- Time stretching

It is opposite to pitch shifting where it changes the time duration of the audio signal without changing the pitch. Here the time is shortened as can be observed in Figure 5.23.

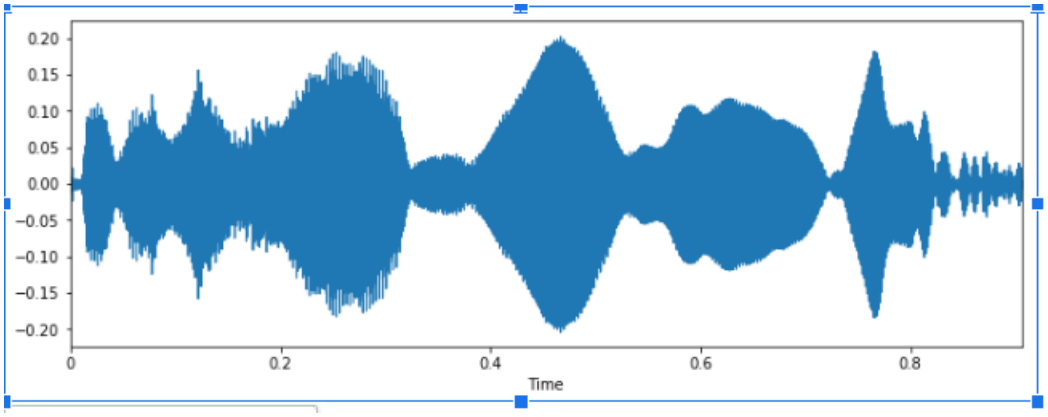

Figure 5.23: Time stretched "Three" wav file

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

There are still uncertainties regarding building an ideal neural network. However, with enriching of the dataset further it is very much possible to obtain a desired accuracy with the model we build for each of the example. In our research we have concluded that PCA used with MFCC can produce a very high accuracy. Our model obtained an accuracy of 94.12% for the test dates after 50 runs for 1D CNN and obtained an accuracy of 82.35% after 50 epochs. The two models that gave the poor result were both involving t-SNE with the CNN and LSTM network accuracy remained at around 49% -50% with high loss when trained in 1D CNN model and 50% accuracy for LSTM model on 50 epochs. This system also had a very high loss. Kernel PCA were not so far behind then PCA. The highest accuracy provider was MFCC with 1D CNN of 98.81% in 50 runs and when used with LSTM provided 96.49% accuracy for 50 runs.

6.2 Future Work

As we have seen t-SNE not giving ideal result directly computing on MFCC, we will apply PCA first on the MFCC coefficients and then compute t-SNE on the result. In this way we are suspecting the patterns are retained within the feature.

Another approach, completely different would be to work with the time series data instead of MFCC or Mel Spectrogram. We can also apply PCA, t-SNE and Kernel PCA on the time series values to reduce dimensions. Also, we can use other techniques such as mean, max, average, min etc. to work with the time series data.

Bibliography

[1] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma", *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[2] J. M. Zurada, *Introduction to artificial neural systems*. West publishing company St. Paul, 1992, vol. 8.

[3] J. H. Friedman, "On bias, variance, 0/1—loss, and the curse-of-dimensionality", *Data mining and knowledge discovery*, vol. 1, no. 1, pp. 55–77, 1997.

[4] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of mfcc", *Journal of Computer science and Technology*, vol. 16, no. 6, pp. 582–589, 2001.

[5] D. M. Hawkins, "The problem of overfitting", *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[6] A. Y. Ng, "Feature selection, l 1 vs. l 2 regularization, and rotational invariance", in *Proceedings of the twenty-first international conference on Machine learning*, ACM, 2004, p. 78.

[7] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various mfcc implementations on the speaker verification task", in *Proceedings of the SPECOM*, vol. 1, 2005, pp. 191–194.

[8] K. S. R. Murty and B. Yegnanarayana, "Combining evidence from residual phase and mfcc features for speaker recognition", *IEEE signal processing letters*, vol. 13, no. 1, pp. 52–55, 2005.

[9] S. Zhong and J. Ghosh, "Generative model-based document clustering: A comparative study", *Knowledge and Information Systems*, vol. 8, no. 3, pp. 374–384, 2005.

[10] M. Müller, *Information retrieval for music and motion*. Springer, 2007, vol. 2.

[11] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne", *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[12] M. H. Nguyen and F. Torre, "Robust kernel principal component analysis", in *Advances in Neural Information Processing Systems*, 2009, pp. 1185–1192.

[13] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: A comparative", *J Mach Learn Res*, vol. 10, no. 66-71, p. 13, 2009.

[14] T. Mikolov, M. Karafiát, L. Burget, J. Černock, and S. Khudanpur, "Recurrent neural network based language model", in *Eleventh annual conference of the international speech communication association*, 2010.

[15] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques", *arXiv preprint arXiv:1003.4083*, 2010.

[16] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 7–13, 2011.

[17] S. Wijoyo and S. Wijoyo, "Speech recognition using linear predictive coding and artificial neural network for controlling movement of mobile robot", in *Proceedings of 2011 International Conference on Information and Electronics Engineering (ICIEE 2011)*, 2011, pp. 28–29.

[18] C. Kurian and K. Balakrishnan, "Development & evaluation of different acoustic models for malayalam continuous speech recognition", *Procedia Engineering*, vol. 30, pp. 1081–1088, 2012.

[19] A. Lerch, *An introduction to audio content analysis: Applications in signal processing and music informatics*. Wiley-IEEE Press, 2012.

[20] V. Radha *et al.*, "Speaker independent isolated speech recognition system for tamil language using hmm", *Procedia Engineering*, vol. 30, pp. 1097–1102, 2012.

[21] M. A. Ali, M. Hossain, M. N. Bhuiyan, *et al.*, "Automatic speech recognition technique for bangla words", *International Journal of Advanced Science and Technology*, vol. 50, 2013.

[22] A. Choudhary, M. Chauhan, and M. G. Gupta, "Automatic speech recognition system for isolated and connected words of hindi language by using hidden markov model toolkit (htk)", *Association of computer electronics and electrical engineers (ACEEE)*, 2013.

[23] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks", in *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2013, pp. 6645–6649.

[24] P. Saini, P. Kaur, and M. Dua, "Hindi automatic speech recognition using htk", *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, no. 6, pp. 2223–2229, 2013.

[25] S. Swamy and K. Ramakrishnan, "An efficient speech recognition system", *Computer Science & Engineering*, vol. 3, no. 4, p. 21, 2013.

[26] M. Moneykumar, E. Sherly, and W. S. Varghese, "Malayalam word identification for speech recognition system", *An International Journal of Engineering Sciences*, pp. 22–26, 2014.

[27] G. Nijhawan and M. Soni, "Real time speaker recognition system for hindi words", *International Journal of Information Engineering and Electronic Business*, vol. 6, no. 2, p. 35, 2014.

[28] J. S. Pokhariya and S. Mathur, "Sanskrit speech recognition using hidden markov model toolkit", *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, no. 10, pp. 2278–0181, 2014.

[29] M. Saleh, N. Ibrahim, and D. Ramli, "Data reduction on mfcc features based on kernel pca for speaker verification system", *WALIA journal*, vol. 30, no. 2, pp. 56–62, 2014.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] D. Aleksendric and P. Carlone, *Soft Computing in the Design and Manufacturing of Composite Materials: Applications to Brake Friction and Thermoset Matrix Composites*. Woodhead Publishing, 2015.

[32] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition", in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[33] K. O'Shea and R. Nash, "An introduction to convolutional neural networks", *arXiv preprint arXiv:1511.08458*, 2015.

[34] M. M. H. Nahid, M. A. Islam, and M. S. Islam, "A noble approach for recognizing bangla real number automatically using cmu sphinx4", in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, IEEE, 2016, pp. 844–849.

[35] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-sne effectively", *Distill*, vol. 1, no. 10, e2, 2016.

[36] M. Beverly Engel, *The emotionally abused woman: Overcoming destructive patterns and reclaiming yourself*. Ballantine Books, 2017.

[37] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing lstm language models", *arXiv preprint arXiv:1708.02182*, 2017.

[38] M. Hussain, M. A. Haque, *et al.*, "Swishnet: A fast convolutional neural network for speech, music and noise classification and segmentation", *arXiv preprint arXiv:1812.00149*, 2018.

[39] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions", in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 4779–4783.