Bengali Isolated Speech Recognition

A comparative analysis of the effects of data augmentation on HMM and DNN based acoustic models

Thesis submitted in partial fulfilment of the requirement for the degree of

Bachelor of Science

In

Computer Science and Engineering[1]

Under the Supervision of

Mr. Moin Mostakim

And

Co-Supervision of

Mr. Matin Saad Abdullah
Mr. Md. Shamsul Kaonain

By

Warida Rashid
14301026

Mohi Reza
14101040

School of Engineering & Computer Science

Department of Computer Science & Engineering BRAC University

[1] Mohi Reza – B.Sc. in Computer Science, Warida Rashid – B.Sc. in Computer Science & Engineering

Declaration

We hereby declare that this thesis is based on results obtained from our own work. Due acknowledgement has been made in the text to all other material used. This thesis, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma.

Signature of Supervisor:

____

Moin Mostakim

Department of Computer Science & Engineering,
BRAC University

Supervisor

Signature of Authors:

____

Warida Rashid, 14301026

____

Mohi Reza, 14101040

Acknowledgements

We would like to express our sincere gratitude to our thesis supervisor Mr. Moin Mostakim for his continuous support and guidance. He introduced us to the exciting and challenging field of computational linguistics and his input has been of tremendous value to us.

We would also like to thank our co-supervisors Mr. Matin Saad Abdullah and Md. Shamsul Kaonain for extending every possible help whenever asked for and the engaging conversations that played an important role in the development of this study. They have been a constant source of inspiration over the past four years.

We are also grateful to all participants who helped us record and create *Prodorshok-I,* the isolated Bengali word dataset that has made this study possible.

Abstract

We have created an isolated-word dataset - *Prodorshok 1,* which consists of 34 Bengali words related to navigation with 1011 voice samples. The word set is intended to help design speaker dependent/independent, voice-command driven automated speech recognition (ASR) systems that can potentially improve human-computer interaction. This paper presents the results of an objective analysis that was undertaken using a subset of words from *Prodorshok I* to help assess its reliability in ASR systems that utilize Hidden Markov Models (HMM) with Gaussian emissions and Deep Neural Networks (DNN). The results show that simple data augmentation involving a small pitch shift can make surprisingly tangible improvements to accuracy levels in speech recognition, even when working with small datasets. *Prodorshok I* will be expanded upon and made publicly available for others to use under an Open Data License (ODbL).

Table of Contents

List of Figures and Tables

1. Introduction

Automatic Speech Recognition (ASR) is an exciting domain under computational linguistics that has garnered the interest of researchers for at least six decades. The aim of this study is to create a useful, speaker-independent dataset of isolated Bengali words – *Prodorshok* 1, and to test this dataset using HMM and DNN based acoustic models written in Python. In particular, we look at the effects of data augmentation on prediction accuracy levels.

1.1 Motivation

The key impetus behind the creation of *Prodorkshok I* is twofold: (i) it serves to help fill the lack of preprocessed, easy to use Bengali isolated-word datasets that are easily accessible and (ii) the potential speech-recognition based assistive technologies that can be derived from such a word set can considerably improve human-computer interaction by enabling hands-free navigation.

Contemporary software systems are mostly reliant upon increasingly rich graphical user interfaces. While this has brought drastic improvements in usability for the general population, many applications are not at all fit for usage for people suffering from disabilities such visual impairment or lack of mobility. Where accessibility features do exist, they are rarely designed with Bengali speaking users in mind. With an estimated 650 000 visually impaired adults in the Bangladesh [1], there is a tangible need for more inclusive alternatives to purely GUI-driven ways to navigate digital interfaces and *Prodorshok I* can help fill this void.

1.3 An Overview of Automatic Speech Recognition

Automatic Speech Recognition (ASR) is the process of identifying and responding to the distinct sounds produced by human speech. It enables a program or a machine to convert the spoken words or sentences in machine readable format and identify them. ASR systems can be classified into the following types:

1. **Speaker Dependent:** The speaker dependent system can recognize speech from a single speaker. These systems are easier to develop, cheaper and more accurate but not as flexible and adaptive to practical applications involving multiple speakers.

2. **Speaker Independent:** The speaker independent system is developed to work with any speaker. These systems are more complex, expensive and have less accuracy but are more flexible and adaptable for practical applications that need to handle the acoustic variability in speech from many speakers.

Speech recognition can also be classified into the following types:

1. **Isolated Word Recognition:** The words recognized by such systems are separated by pauses or include utterances of one single word at a time. These are easier to construct because the end points of the speech signals are easily detectable and the pronunciation of a word is not affected by other words that precede or follow. They can also be quite robust since all possible patterns for the inputs are known. Isolated word recognition systems can be designed and built for certain application oriented words such as, digits recognition for phone dialing, navigation related words e.g. left, right, forward, backward etc.

2. **Continuous Speech Recognition:** A continuous speech recognition system operates on words that are connected without pauses. It recognizes the natural flow of speech. The increased complexity of such systems arises because of a number of factors. First, it requires detection of start and end points of each word. Another problem is that, since the phonemes are connected together, utterance of each word is affected by the words that surround it. This is known as "co-articulation". It is also affected by the speed and rate of speech.

Some speech recognition systems may only need to recognize a few words, for example the digits, while others need a large set of words depending on the application. Automatic speech recognition systems can be further classified in the following ways based on the size of the vocabulary that is recognized by the system:

1. **Small Vocabulary:** Usually works with less than a hundred words. It is possible to get quite an accurate result for speaker independent systems with small vocabulary. Applications such as voice interface for phone dialing, navigation of robots, operating smartphones and so on.
2. **Medium Vocabulary:** Systems that use medium sized vocabulary usually work with a set of 1000-3000 words.
3. **Large vocabulary:** These systems use thousands or tens of thousands of words (e.g. 20000 words). It is difficult to attain a satisfactory level of accuracy with a large vocabulary dataset. They usually need to be speaker dependent to get a higher accuracy.

1.4 Previous Studies

1.4.1 Early ASR Systems

Instilling in machines the ability to recognize human speech is incredibly useful. As such, efforts in creating ASR systems have been ongoing for over seven decades. In 1952, Davis, Biddulph, and Balashek [2] of Bell Laboratories designed a system that recognized "telephone-quality" spoken digits through measuring formant frequencies during vowel regions of each digit. In 1956, Olsen and Belar [3] of RCA Labs built a 10-sylabal speaker-dependent recognizer. Three years later, Forgie and Forgie [4] of MIT Lincoln lab built a 10-vowel recognizer that was speaker independent. In the 1960s, several Japanese labs built special purpose speech recognizers. Two notable examples include the vowel recognizer by Suzuki and Nataka [5] of Radio Research Lab and the phoneme recognizer by Sakai and Doshita [6] of Kyoto University. These early systems were mostly based upon the theory of acoustic phonetics.

In the early 1970s, the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense funded the Speech Understanding Research (SUR) program that eventually led to the creation of "Harpy", a system built at Carnegie Melon University [7]. It utilized graph search techniques that traversed through a connected network of lexical representations of 1011 words and was reasonably accurate. The 1970s marked the start of two broad directions in ASR research, with IBM and AT&T Bell Laboratories leading the advancement these two schools of thought, driven to find useful commercial applications.

IBM was concerned with the creation of a *speaker-dependent* system called "Voice Activated Typewriter" (VAT) that converted spoken words into written text. They placed technical emphasis on maximizing vocabulary size and representing the grammatical structure of the language using statistical syntactical rules. These rules made up different *language models* that described how likely a sequence of symbols could appear in the speech signal. One such model that is highly popular is called the n-gram model, which has extensive applications in designing large vocabulary ASR systems.

AT&T Laboratories was concerned with creating a *speaker-independent* system that catered to large populations of people, removing the possibility of training with individual speakers. Technical emphasis was placed upon ability to deal with acoustic variability intrinsic to speech signals of different voices, often with notably different regional accents. These efforts sparked the creation of speech clustering algorithms that could handle such complexity, ultimately paving the way for the use of rigorous statistical modeling frameworks such as HMM and ANN based acoustic models, which are both used in this study. Juang and Rabiner [8] has written a detailed account of the development and history of ASR based systems that corroborates and expands upon the information presented in this section.

1.4.2 Research focusing on the Bengali Language

The research landscape for Bengali Speech Recognition is relatively nascent in comparison to the rich history of ASR system development involving other languages. The most notable Bengali dataset that is available for free is SHRUTI Bengali Continuous ASR Speech Corpus [9], made available by the Society of Natural Language Technical Research. There are some existing studies that have been derived from this continuous speech dataset. Das and Mitra [10] used a Hidden Markov Toolkit (HKT) to align its speech data that were later manually pruned. Mandal et al. [11] also used SHRUTI to create a phone recognition (PR) system that used an optimum text selection technique to decipher the smallest discrete unit of sound in uttered speech.

Mandal, Das and Mitra [12] recently introduced SHRUTI-II, a SPHINX3 based Bengali ASR System and demonstrated its use in an E-mail based computer application designed to aid visually impaired users. Mohanta and Sharma [13] did a small study on emotion detection in Bengali speech. Their goal was to identify neutrality, anger and sadness in speech using Linear Prediction Cepstral coefficient (LPCC), Mel-frequency Cepstral Coefficient (MFCC), pitch, intensity and formant. Bhowmik and Mandal [14] applied deep neural network based phonological feature extraction technique on Bengali continuous speech.

In the realm of text-based corpora, Basu et al. [15] from Louisiana State University has created "the noisy Bangla handwritten digit dataset" that consists of additive white Gaussian noise, motion blur and a combination of additive white Gaussian noise and reduced contrast. Adak, Chaudhuri and Blumstein [16] used a Convolutional Neural Network (CNN), coupled with a recurring model for offline cursive Bengali word recognition.

2. Methodology

In this chapter, the dataset used and the theory behind HMM with Gaussian Emissions and DNN based classifiers are explained. This chapter also discusses the theory and application of preprocessing of speech and feature extraction. ASR for isolated words has three main steps:

1. De-noising and Enhancement of Speech
2. Feature Extraction
3. Classification

In each step, the theories behind it and the experimental configuration are described.

2.1 De-noising and Enhancement of Speech

The first step in ASR process is preprocessing. It is necessary to de-noise and enhance the raw speech data before it goes through any further signal processing. The raw speech data can be corrupted by three kinds of noise:

i. Recording noise
ii. Electrical noise
iii. Environmental noise

The first two types of noises can be easily compensated by training the system with data that has similar noise. However, the third one can severely degrade the performance of the recognition because of its varying nature. The challenge of reducing noise is reducing external noise without affecting the low-intensity components of the speech [33].

All word samples were put through five stages of enhancement. First, stereo channels were merged into a single mono channel. Then, static background noise was attenuated using a noise-reduction algorithm based on Fourier analysis. Unique noise profiles were used for each word samples for best results. Then, the sound signals were normalized to have maximum amplitude of -1.0 dB and 0 mean amplitude displacement for uniformity. Any silence at the beginning or end was truncated. Finally, the audio samples were cloned into two separate datasets, one of which was then synthetically augmented by including pitch altered voice samples from existing data. The effect of these five stages on a particular audio sample representing the Bangla word for "first" can be seen in fig. 1. The end result is a concise audio sample that is ready to be used to train and test different acoustic models.

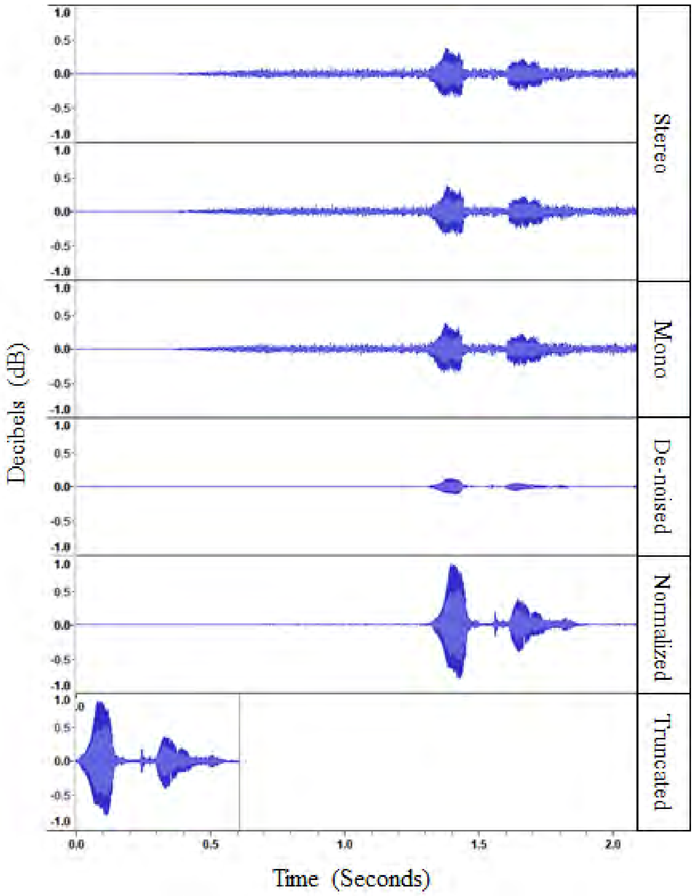


Figure 2.1.1: Visualizing Speech Enhancement

2.2 Feature Extraction

Feature extraction is the first step in Automatic Speech Recognition which encodes the speech data as a set of quantifiable feature vectors that are fed into the acoustic models. Speeches are certain sounds that are shaped by articulation of the vocal cord, nose, tongue, teeth and other organs. Speech signal is basically a one dimensional waveform which has some discrete and measureable qualities to it, such as, energy level, certain frequencies and so on. It is important to select feature vectors in a way that minimizes redundancy, gets rid of unwanted or unimportant features and focuses on features that distinctly represent different sample classes.

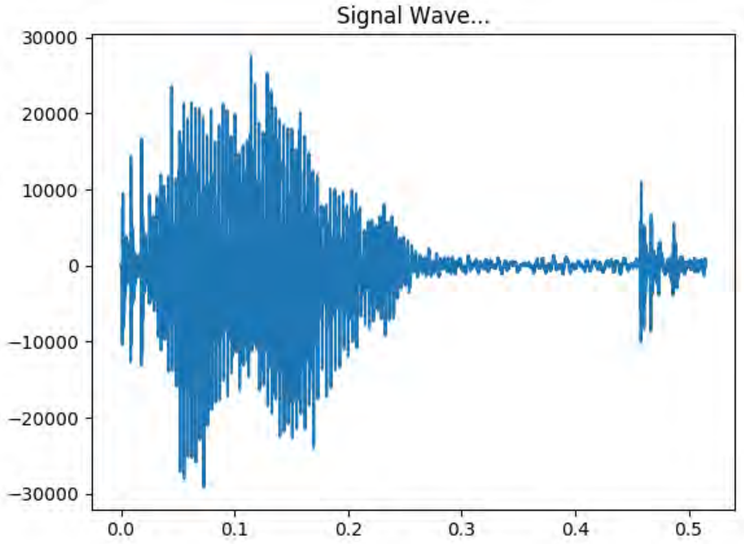


Figure 2.2.1: Waveform of a spoken word

Mel-Frequency Cepstral Coefficients (MFCC) is the most popular feature vectors for speech and voice recognition. Paul Mermelstein and S. B. Davis [17, 18] are accredited for this idea in the 1980's and mel-frequency cepstrum has been the state-of-the-art till date. The popularity of MFCC vectors is attributed to the fact that it closely mimics the way human ears perceive sound and respond accordingly. MFCCs are commonly derived in the steps [19, 20, 21] described in the paragraphs that follow.

Speech is a time varying signal, i.e. it is always changing. Therefore, the signal is divided into small segments, e.g. 20-30ms frames where the signal can be considered to be statistically stationary. If the frame is too long, the signal may change too much which will cause the feature vectors to be a lot less useful for accurate predictions. Frame steps are 10ms which causes some overlapping of frames. We used 25ms frames.

After dividing the signals into small frames, the second step is calculating the power spectrum of each frame. This was inspired by the mechanism of cochlea, an organ inside human ear. It vibrates at different spots depending on the frequency which in turn fires different neurons. The periodgram spectral estimates similarly detect the frequencies present in the input signal. This is done by taking the Discrete Fourier Transform (DFT) of each step. The DFT is represented by the following equation:

$$X_N [n] = X_N [n+N] \qquad (2.2.1)$$

Here, $X_N[N]$ is a signal with a period of N. If the DFT is $X_N[n]$, k is the length of the DFT,

$$X_N[k] = \sum_{n=0}^{N-1} x_N [n] e^{-j2\pi \mathrm{nk}/\mathrm{N}} \qquad (2.2.2)$$

$$X_N[n] = \frac{1}{N} \sum_{k=0}^{N-1} x_N [k] e^{j2\pi \mathrm{nk}/\mathrm{N}} \qquad (2.2.3)$$

The periodgram estimate of power spectrum is given by:

$$P_N[k] = \frac{1}{N} |x_n[k] |^2$$

257 out of 512 point transform are kept.

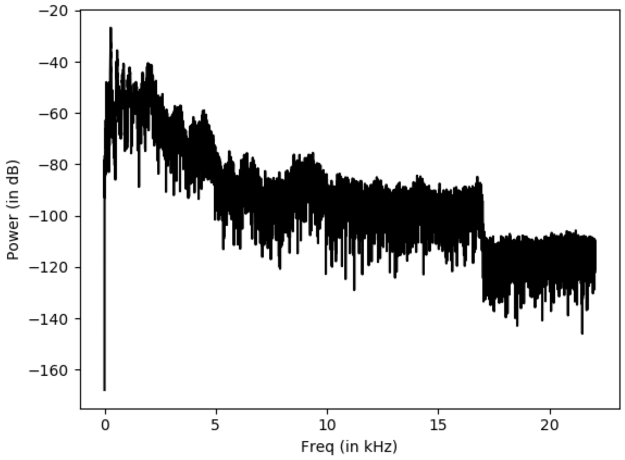


Figure 2.2.2: Power Spectral Density of the word "এক"

The initial periodgram estimated contain a lot of information that are unnecessary for Automatic Speech Recognition. The cochlea cannot effectively differentiate between two closely spaced frequencies. This is why the Mel Filter bank is used. The first filter grows from narrow to wider as it provides an estimate of the amount of energy present near zero hertz to higher frequencies. 20-40 (here 26 is used) triangular spaced filters are applied to the periodgram power estimates. This results in 26 vectors of length 257. Each filter bank is multiplied with the power spectrum and then the coefficients are added up to calculate the filter bank energies.

In the next step, the logarithm of the filter bank energies is taken. This is also influence by how humans perceive sound. It is necessary for the large variation of energy to sound similar. So the perceive loudness is increased so that the variations are insignificant since the sound was already loud enough compared to the energy variations to make it ignorable. Therefore, the logarithm of each of the 26 energies from the previous step is taken.

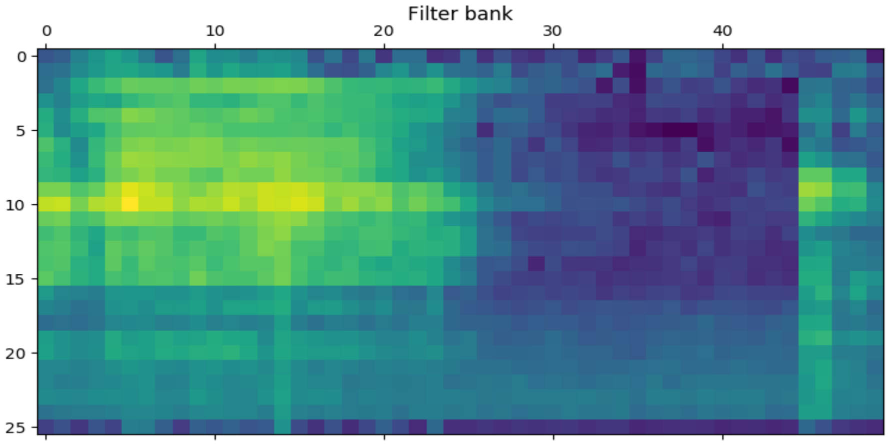


Figure 2.2.3: Log Filter bank

The last step of the feature extraction process is calculating the Discrete Cosine Transform (DCT) of the log filter bank energies from the previous step. The filter bank energies are correlated to each other since they are overlapping. This step fixes that so that these can be used for diagonal covariance matrices, e.g. the ones used for Hidden Markov Model based classifiers. This step results in 26 cepstral coefficients only the lower 13 of which are kept.

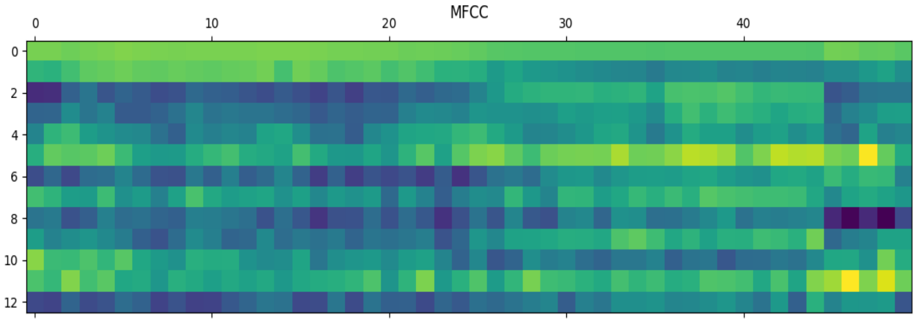


Figure 2.2.4: Extracted MFCC features

2.3 Deep Neural Network

Deep Neural Network has proven to be successful in speech recognition and is currently a widely researched area under this field. [24, 25, 26]

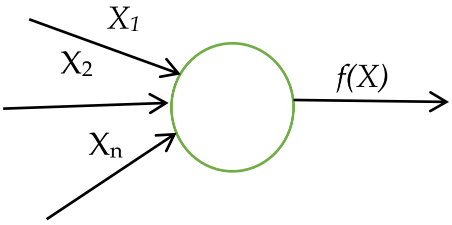
Artificial neural networks are simplified representation and simulation of the neuronal structure present in brains. Deep neural networks are artificial neural networks where multiple layers of neuron are used. To explain the analogy of how they work, neurons in brain can receive signals and there are communication links from one neuron others. A neuron can respond to input signals. If the strength of the signal is above a certain level, the particular neuron is activated or fired. The output of any neuron can be the input to other neurons. There is a feedback involved in the process depending on the result produced by activating a neuron. If the result is the desired output, the connection that caused that result is strengthened. The system learns through observations and the feedback mechanism.

The stated concept of neuronal structure is mimicked in Artificial Neural Network. The training of an Artificial Neural Network system can be of two types. First, supervised learning where a set of labeled input data is used for trainings. The labels are the desired outputs. On the other hand, in unsupervised learning, the network learns the structure of the given input and learns through feedback for the outcome it produces.

2.3.1 Activation Function

An artificial neuron is called a perceptron. The idea of perceptron was brought forth by Frank Rosenblatt in the 1950s and 1970s which was inspired by the earlier work of Warren McCulloch and Walter Pitts. However, different models for neurons are now being used one of which is the Sigmoid Neuron. [22]

As shown in the figure below, a perceptron receives inputs (X_1, X_2………X_n) and generates a binary output (Y). The output is a function of the input and weights (W_1, W_2.....Wn). Weights are variables that change during the learning process. There is a certain threshold value. If the weighted sum of the inputs is greater than the threshold, then the output of the perceptron is 1, or the perceptron will fire. Else, the output is 0. The function f(X) is the activation function for a perceptron.



$$f(X) = \begin{cases} 0, & \sum_j W_j\, X_j \leq \text{Threshold} \\ 1, & \sum_j W_j\, X_j > \text{Threshold} \end{cases} \qquad (2.3.1.1)$$

Figure 2.3.1.1: Artificial Neuron

On the other hand, for a sigmoid neuron, the sigmoid function works as the activation function. The sigmoid function is a widely used function for feed-forward network with back propagation because of its non-linearity and simplicity of computation [27]. The function is given by:

$$\text{f(X)} = \frac{1}{1+e^{-g(X)}} \qquad (2.3.1.2)$$

Here h(x) is the input. The function generates the following curve:

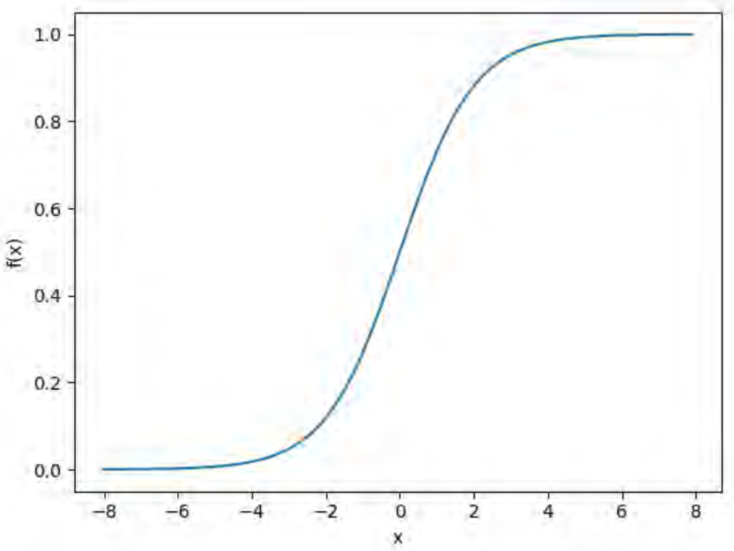


Figure 2.3.1.2: Plot of Sigmoid Function

In practical application of using the sigmoid function as an activation function, W_i is real valued weight, X_i is the input then the weighted input of a nod is given by:

$$g(X) = X_1W_1 + X_2W_2 + \ldots\ldots + X_iW_i + \ldots\ldots + X_nW_n + b \qquad (2.3.1.3)$$

The weight variable is changed depending on the how much the relationship between the inputs to the output need to be strengthened. As the figure shows, weights control the slope of the sigmoid function and the bias controls when the node activates. If x is greater than a certain value, the node can be activated. Biases ensure that condition.

The following figures show the effect of different weight to the sigmoid function:

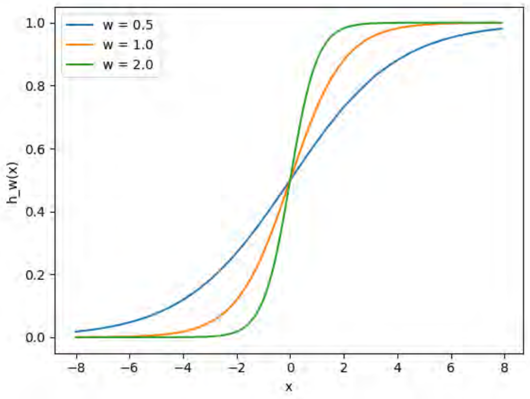


Figure 2.3.1.3 (a): Effect of Different Weights on a Sigmoid Function

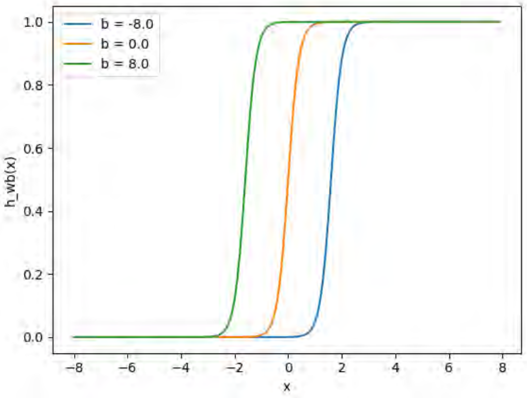


Figure 2.3.1.3 (b): Effect of Biases on a Sigmoid Function

2.3.2 Multi-Layered Feed-forward Network

Feed-forward network is the type of Artificial Neural Network where connections between the nodes do not form a cycle [23]. A simple three layered feed-forward neural network structure is shown below:

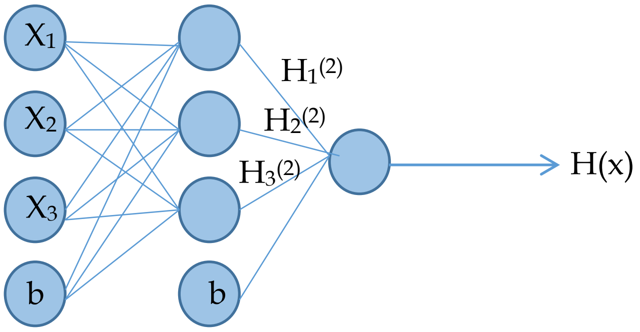


Figure 2.3.1.4: A three layered neural network

The network can be defined as follows:

$$\begin{gathered} H_1^{(2)} = f(W_{11}^{(1)}X_1 + W_{12}^{(1)}X_2 + W_{12}^{(1)}X_2 + b_1^{(1)}) \\ H_2^{(2)} = f(W_{21}^{(1)}X_1 + W_{22}^{(1)}X_2 + W_{22}^{(1)}X_2 + b_2^{(1)}) \\ H_3^{(2)} = f(W_{31}^{(1)}X_1 + W_{32}^{(1)}X_2 + W_{32}^{(1)}X_2 + b_3^{(1)}) \\ H(x) = f(W_{11}^{(2)} H_1^{(2)} + W_{12}^{(2)} H_2^{(2)} + W_{12}^{(2)} H_2^{(2)} + b_1^{(2)}) \end{gathered} \tag{2.3.1.4}$$

Here $W_{ij}^{(L)}$ represent the weight associated with the connection of node number i to j where j is in layer L and I is in layer L+1.The notation $b_i^{(L)}$ stands for the bias weight in node number i in layer L+1. Finally, H(x) is the output of the last node in layer 3. The third layer node takes the output of the nodes in the second layer and thus there exists hierarchical network of nodes.

2.3.3 Backpropagation

Backpropagation is the process of minimizing the differences between actual output and the desired output or the error based on the training samples with labels. This is used by optimization algorithms for adjusting the weight for each connection between neurons or nodes in different layers based on the accumulated error of a batch in the training data. The goal of backpropagation is to refine the mapping of inputs to outputs. The error is computed using a cost function which is propagated back through the network. Different optimization algorithms are used for the process.

2.3.4 Optimization Algorithm

As stated earlier, the weights associated with the connections between the nodes in different hierarchical layers decide the strength of the relationship between the inputs and outputs. The target is to minimize the error. An error occurs when an input does not produce the desired output. In supervised learning, the system is trained with data were for each input the correct output is known. In the optimization process, the weights are varied to minimize the error based on the given samples during the training. It is also necessary that the optimization do not over-fit to the training data and can generalize well for the unseen test data. The objective of an optimizer is to get to the minimum point of the error curve for different weights. There are different optimization techniques such as Stochastic Gradient Descent, Limited memory BFGS, Conjugate Gradient and so on [12]. For our experiment, Adam, a stochastic optimization method was used which combines the advantages of two popular methods AdaGrad and RMSProp. This technique was first introduced in 2014 [13]. It takes the following parameters:

- **Learning rate**: A floating point value. The learning rate.
- **beta1**: A float value or a constant float tensor. The exponential decay rate for the 1st moment estimates.
- **beta2**: A float value or a constant float tensor. The exponential decay rate for the 2nd moment estimates.
- **epsilon**: A small constant for numerical stability. Float >= 0. Fuzz factor.

2.3.4 The Cost Function

As mentioned before, the system minimizes the error iteratively by varying the weight with an optimization technique. A way of generalizing this process by not overfitting it to the training set is using cost function. It is a measure of how good a neural network does based on the training samples. For the experiment, softmax cross entropy with logits were used. It measures the probability error in discrete classification where each sample can belong to exactly one class.

2.3.5 Specifications for the experiment

The Deep neural network used for the experiment has total four layers including an input layer, an output layer and three hidden layers. Each hidden layer had 1500 nodes. For the cost function, softmax cross entropy with logits was used and for optimizer Adam optimizer [29] was used.

2.4 Hidden Markov Model with Gaussian Emission

Hidden Markov Models have been successfully used for time varying sequences such as audio signal processing. The underlying idea behind the Hidden Markov Model is that, it models sequences with discrete states. The way this maps to the problem of speech recognition is, during the feature extraction process, speech signals are transformed into features of discrete time slices or frames. Therefore there are a finite number of frames in a particular word. When a particular sequence of features is given, the model can yield the probability of that sequence being a certain word. Here, the phonemes i.e. the distinct units of sound that can be produced are discrete states and the sequences of MFCCs which represent the uttered word are observations. The probability of observing MFCC sequences given the state is performed using Gaussian Emissions.

The details [30] of how it works are stated below:
Hidden Markov Model is a probabilistic model which can produce a sequence of observation X by a sequence of hidden states Z. It is generated by a probabilistic function associated with each state.
An HMM is usually represented by λ where λ = (A, B, Π). It can be defined by the following parameters:
O = $\{o_{1,} o_2 \ldots\ldots\ldots\ldots o_m\}$. This is an output observation sequence. For speech recognition, this represents the MFCC feature vectors.
Ω = {1, 2,………,N}. This is a set of states. For speech recognition, it is the phoneme labels.
A = $\{a_{ij}\}$. This is the transition probability matrix. It represents the probability associated with transition from state i to state j.

B = {$b_i(k)$}. It is the output probability, i.e. the probability of emitting a certain observation o_k in the state i.

Π = Start probability vector.

There are three basic problems for HMM:

1. Estimating the optimal sequence of states given the parameters and observed data.
2. Calculating the likelihood or probability of a data given the parameters and observed data P (O | λ).
3. Adjusting the parameters given the observed data so that P(O | λ) is maximized.

For isolated word recognition, for each word in the vocabulary, a separate N-state HMM is designed. For each word, the model is trained with feature sequences (MFCCs) of multiple utterances of a single word by different people. This is represented by problem 3 which is estimating the optimal model parameters. The solution is manifested by problem 1. Here, the frames of each word in the vocabulary (as described under feature extraction) are a state and the properties of the model are evaluated based on the corresponding MFCCs that led to that observation. Word recognition is performed by solution to problem 2 where the likelihood of an observed sequence of MFCC is calculated given the model parameters.

2.4.1 Estimating the parameters

The solution to problem 3 is the most difficult one since there is no analytical method to maximize the probability in the training data. The idea is to estimate the model parameters so that P (O | λ) is maximized for the training observations. The optimal Gaussian mixture parameters for a given set of observations can be chosen such that the probability reaches maxima by using the Baum-Welch algorithm or Expectation Maximization (EM) algorithm [31]. It is a gradient based optimization method which is likely to converge at the local maxima.

2.4.2 Decoding

Estimating the state sequence S given an observation sequence X and the model λ is done using the Viterbi algorithm [32]. It is a formal technique for finding the best state sequence based on dynamic programming method [30].

2.5 Experiments on Prodorshok I

2.5.1 Dataset Description

The dataset consists of recordings of single utterances of 30 Bengali words by 35 native speakers in Dhaka, totaling 1050 voice samples. The word set has been specifically constructed to be used in systems that implement hands-free selection and navigation of digital interfaces. It includes Bengali words for 10 digits (0 to 9), 10 directional words (East West, North and South, up, down, left, right, forward, backward) and 10 positional words (First to tenth). See Appendix 8.1 for a full list of words.

2.5.2 Preparing the Dataset for Experimentation

In order to prepare the dataset for experimentation, the following steps were undertaken:

i. 30% of the dataset was reserved for testing, the remaining 70% was separated to be used for training the HMM-GMM and DNN classifiers.

ii. A copy of the training dataset was augmented by including pitch-shifted variations of the word set. We chose a -3% shift in pitch because anything more than that lead to loss in comprehension.

iii. A third set of data, including voice samples from a single speaker was created in order to test accuracy levels in speaker-dependent systems.

2.5.3 Running the Classifiers

Once the dataset was prepared, the two different classifiers, HMM-GMM and DNN were tested on the speaker dependent and a speaker independent systems. Further tests were conducted on the augmented version of the speaker-independent system. The intention was to evaluate the performance of the classifiers and the effect of data augmentation on accuracy levels.

3. Results

The experiments show that, in the speaker independent system without augmentation, the accuracy of prediction for both the classifiers are quite low. After data-augmentation, the performance of the HMM-GMM model increased by 6.12% and that of the DNN by 7.65%. The overall performance however was better with the HMM-GMM model. On the speaker dependent system, the performance of the HMM-GMM model is quite high with an accuracy level of 96.67%. The performance score for DNN for speaker independent system is comparatively lower, at 47.84% with augmentation and 40.19% without. Further experiments show a positive correlation between the number of utterances per word and the accuracy level.

3.1 Average Percentage Accuracy Levels

Each sub-category in Table 1 denotes the average accuracy score that were derived from three consecutive runs of a particular classifier. These results are presented visually in figure 3.1.1. A detailed list of words are provided in Appendix 8.1

		Classifier	
		HMM-GMM (%)	DNN (%)
Speaker Independent System	With Augmentation	56.28	47.84
	Without Augmentation	50.07	40.19
Speaker Dependent System		96.67	43.75

Table 3.1.1: Average Accuracy Levels

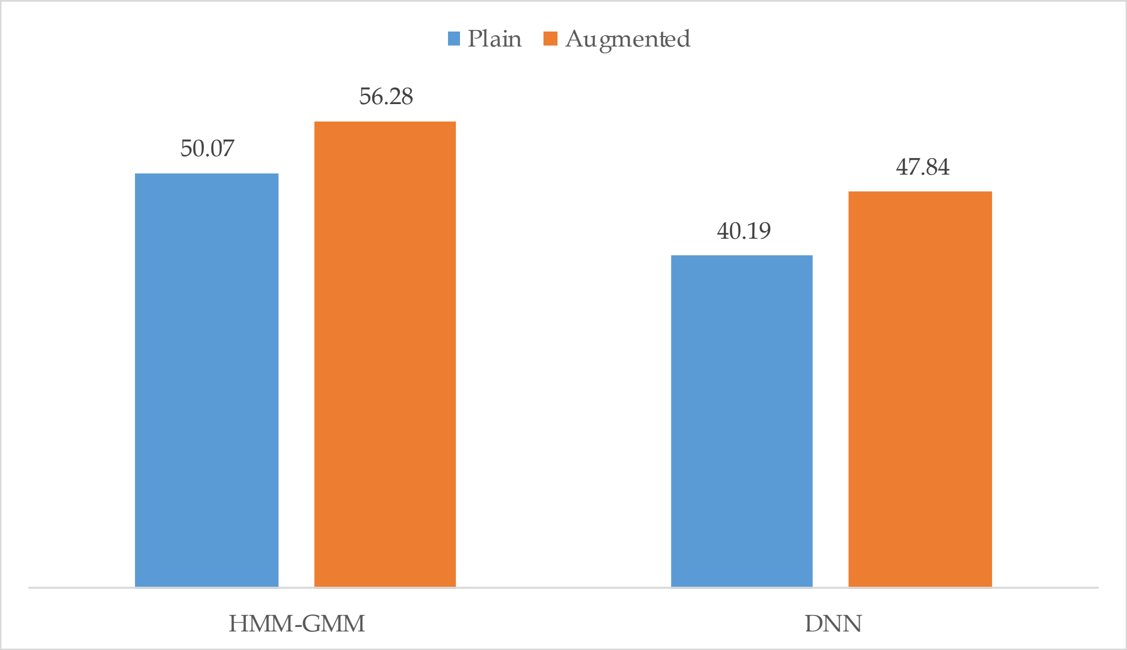


Figure 3.1.1: Effects of Augmentation on Accuracy Level

3.2 Correlation between Utterances per Word and Accuracy Level

Table 2 lists how accuracy levels derived from each classifier varied as the number of utterances for each word varied. These results are presented visually in figure 3.2.1.

Number of Utterances per Word			Classifier	
Total	Test	Train	HMM-GMM (%)	DNN (%)
10	3	7	34.93	21.53
15	4	11	35.38	22.89
20	6	14	36.97	28.57
25	7	18	41.00	31.65
30	9	21	50.75	34.63
35	10	25	52.51	40.19

Table 3.2.1: Number of Utterances per Word and Accuracy Level

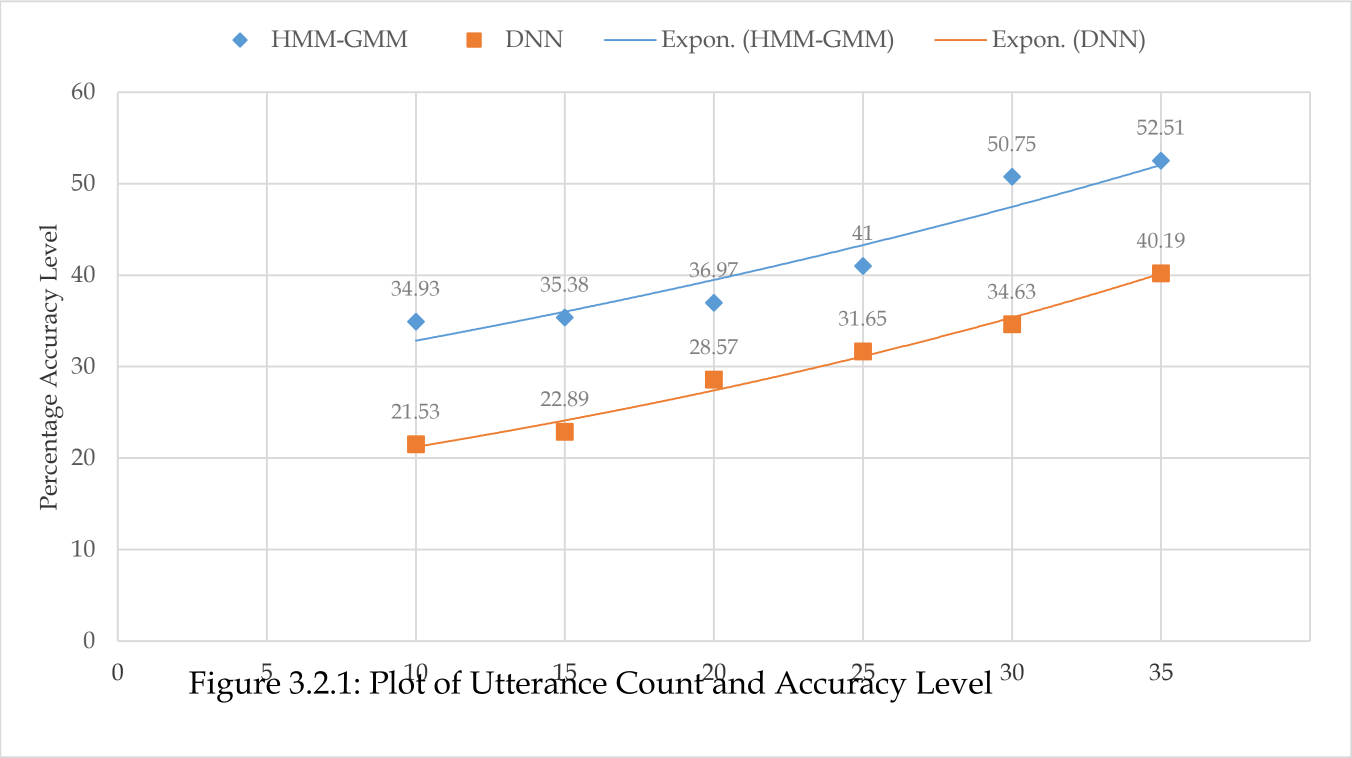


Figure 3.2.1: Plot of Utterance Count and Accuracy Level

These results can also be visualized using a grid of confusion matrices as shown in Figure 3.2.2. These are matrices of true labels versus predicted labels. The higher the accuracy, the greater the number values that lie on the diagonal.

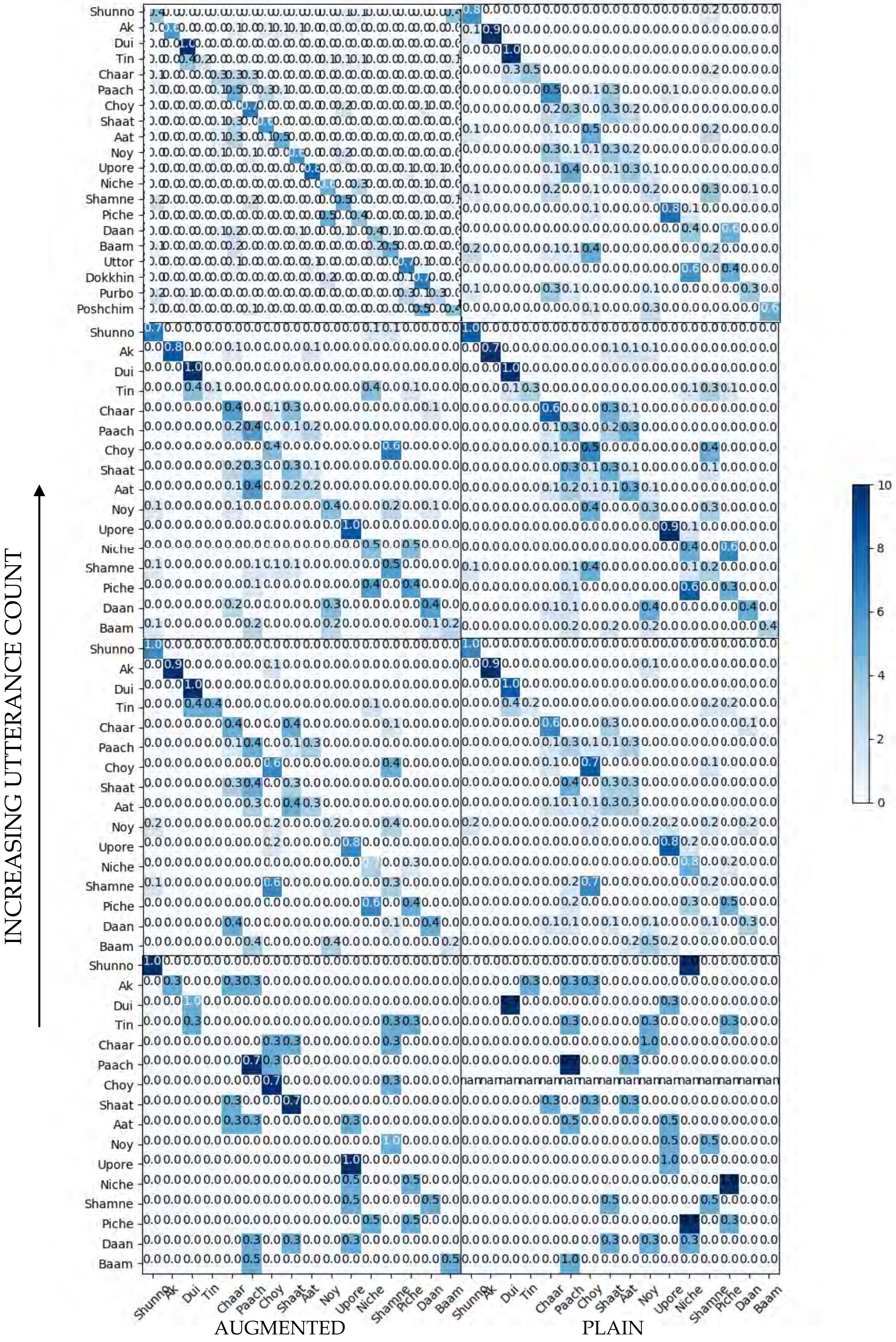


Figure 3.2.2: Confusion Matrices

4. Discussion

The two classifiers we have used are both based on tried and tested acoustic models. Yet, owing to the intrinsic acoustic variability in sound signals spoken by multiple speakers, accuracy levels for speaker-independent systems have been below 60%. The size of the corpus and the sparseness of our training data are what likely affected the performance of the classifiers. Judging by the experimental results summarized in Table 3.2.1, there appears to be a clear positive correlation between utterance count and accuracy levels. As such, expanding the current dataset to incorporate higher number of utterances per word will likely solve this issue.

Despite these limitations of working with a sparse dataset, experimental results summarized in Table 3.1.1 indicate that improvements can be made by augmenting the data the through simple measures. This observation is likely due to the fact that these simple measures such as a change in pitch or length of spoken signals help overcome data sparsity. It does this by including speech signals that have been transformed enough to mimic utterances by more people, thereby improving performance.

Results pertaining to speaker-dependent systems have been very promising, yielding accuracy levels averaging at 96.67% when using the HMM-GMM based classifier. Quite interestingly, the DNN classifier showed only negligible improvements. Judging by the trend depicted in Figure 3.2.1, it is fairly clear that an increase in utterance count is likely to improve accuracy levels. However, whether or not the rate of these improvements will be sufficient for tangible increases in reliability remains unclear, since both trend lines appear to be roughly parallel. Such questions can only be answered through further empirical analysis involving a larger dataset.

Based on the observations above, we can make the following recommendations to anyone who is interested in utilizing Prodorshok I in their own projects: (i) Prodorshok I in its current form can already be used to design highly reliable speaker-dependent systems, and (ii) observations from the upward sloping trend lines in Figure 3.2.1 give us hope of further improvements in speaker-independent systems if Prodorshok I is expanded.

5. Concluding Remarks

In this thesis, we addressed the problem of recognizing isolated words in Bengali using two classification algorithms that use HMM-GMM and DNN based acoustic modeling. One of the main contributions of our work is the creation of *Prodorshok 1,* the preprocessed, ready to use dataset of isolated Bengali words related to navigation and position. Using a data augmentation technique that relies upon a simple pitch shift, we have shown that tangible improvements in speech recognition accuracy levels can be achieved even in small datasets.

6. Future Work

It will be interesting to explore the performance of our dataset when using additional hybrid classifiers such as DNN-HMM. This could potentially address some of the discrepancies we faced when using classification algorithms relying on singular acoustic modeling techniques. Furthermore, it would be useful to see if the trend lines depicted in Figure 3.2.1 hold and if the two trend lines diverge, converge or intersect as the increase in utterance count is continued. Empirical analysis of this nature relies upon the availability of more data. Hence, there is a need for further expansions of our dataset to incorporate a larger vocabulary size and an increased variation in speech for every word for more in-depth analysis.

7. References

[1] B. Dineen, "Prevalence and causes of blindness and visual impairment in Bangladeshi adults: results of the National Blindness and Low Vision Survey of Bangladesh", British Journal of Ophthalmology, vol. 87, no. 7, pp. 820-828, 2003

[2] K. H. Davis, R. Biddulph, and S. Balashek, “Automatic Recognition of Spoken Digits,” The Journal of the Acoustical Society of America, vol. 24, no. 6, pp. 637–642, Nov. 1952.

[3] H. F. Olson and H. Belar, Phonetic Typewriter, J. Acoust. Soc. Am., Vol. 28, No. 6, pp. 1072-1081, 1956.

[4] J. W. Forgie and C. D. Forgie, Results Obtained from a Vowel Recognition Computer Program, J. Acoust. Soc. Am., Vol. 31, No. 11, pp. 1480-1489, 1959.

[5] J. Suzuki and K. Nakata, Recognition of Japanese Vowels—Preliminary to the Recognition of Speech, J. Radio Res. Lab, Vol. 37, No. 8, pp. 193-212, 1961.

[6] J. Sakai and S. Doshita, The Phonetic Typewriter, Information Processing 1962, Proc. IFIP Congress, Munich, 1962.

[7] B. Lowerre, The HARPY Speech Understanding System, Trends in Speech Recognition, W. Lea, Editor, Speech Science Publications, 1986, reprinted in Readings in Speech Recognition, A. Waibel and K. F. Lee, Editors, pp. 576-586, Morgan Kaufmann Publishers, 1990.

[8] B. H. Juang and L. R. Rabiner, “Automatic Speech Recognition – A Brief History of the Technology Development,” Elsevier Encyclopedia of Language and Linguistics, pp. 1–24, 2005.

[9] B. Das, S. Mandal and P. Mitra, *Shruti Bengali Bangla ASR Speech Corpus*. [online] Available at: http://cse.iitkgp.ac.in/~pabitra/shruti_corpus.html [Accessed 19 Aug. 2017].

[10] S. Mandal et al. "Developing Bengali Speech Corpus for Phone Recognizer Using Optimum Text Selection Technique," Proc. Conf. Asian Language Processing (IALP), 2011 pp.268-271, 2011

[11] B. Das, S. Mandal and P. Mitra, "Bengali speech corpus for continuous automatic speech recognition system," Proc. Conf. Speech Database and Assessments (Oriental COCOSDA), pp.51-55, Taiwan, 2011

[12] B. Das, S. Mandal and P. Mitra, “Shruti-II: A vernacular speech recognition system in Bengali and an application for visually impaired community,” in Students' Technology Symposium (TechSym), 2010 © IEEE. doi: 10.1109/TECHSYM.2010.5469156

[13] A. Mohanta and U. Sharma, “Bengali speech emotion recognition” in Computing for Sustainable Global Development (INDIACom), 2016 © IEEE

[14] T. Bhowmik, S. K. D. Mandal, “Deep neural network based phonological feature extraction for Bengali continuous speech” in Signal and Information Processing (IConSIP), 2016 © IEEE. doi: 10.1109/ICONSIP.2016.7857491

[15] S. Basu et al., The noisy Bangla handwritten digit dataset. [online] Available at: http://csc.lsu.edu/~saikat/noisy-bangla/ [Accessed 20 Aug. 2017].

[16] C. Adak ; B. B. Chaudhuri and M. Blumenstein, “Offline Cursive Bengali Word Recognition Using CNNs with a Recurrent Model” in Frontiers in Handwriting Recognition (ICFHR), 2016© IEEE. doi: 10.1109/ICFHR.2016.0086

[17] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," in Pattern Recognition and Artificial Intelligence, C. H. Chen, Ed., pp. 374–388. Academic, New York, 1976

[18] S.B. Davis, and P. Mermelstein (), "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," in IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.28, No. 4 1980, pp. 357–366.

[19] X. Huang, A. Acero, and H. Hon. Spoken Language Processing: A guide to theory, algorithm, and system development. Prentice Hall, 2001.

[20] M. Xu et al, "HMM-based audio keyword generation". In Kiyoharu Aizawa; Yuichi Nakamura; Shin'ichi Satoh. Advances in Multimedia Information Processing – PCM 2004: 5th Pacific Rim Conference on Multimedia (PDF). Springer. ISBN 3-540-23985-5, 2004.

[21] M. Sahidullah, G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker

recognition". Speech Communication. Vol. **54, No.** 4, 2012, pp. 543–565. doi:10.1016/j.specom.2011.11.004.

[22] M. A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015.

[23] A. Zell, Simulation Neuronaler Netze 1st ed. Addison-Wesley, p. 73. ISBN 3-89319-554-8.

[24] G. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," Signal Processing Magazine, IEEE, vol. 29, no. 6, pp. 82–97, 2012

[25] G. E. Dahl et al., "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," Audio, Speech, and Language Processing, IEEE Transactions on, vol. 20, no. 1, pp. 30–42, 2012.

[26] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in INTERSPEECH, 2011, pp. 437–440.

[27] J. Han, C. Morag, "The influence of the sigmoid function parameters on the speed of backpropagation learning". In Mira, José; Sandoval, Francisco. From Natural to Artificial Neural Computation, 1995.

[28] J. Ngiam et al., "On optimization methods for deep learning." Proceedings of the 28th international conference on machine learning (ICML-11). 2011.

[29] D. Kingma and B. Jimmy, "Adam: A method for stochastic optimization," 2014.

[30] L. R. Rabiner "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE 77.2, pp. 257-286, 1989.

[31] Bilmes, Jeff A. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models." International Computer Science Institute 4.510 (1998): 126.

[32] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE transactions on Information Theory 13.2 1967, pp. 260-269.

[33] A. G. Maher, R. W. Kind, J.G. Rathmell. A Comparison of Noise Reduction Techniques for Speech Recognition in Telecommunications Environments. In The Institution of Engineers Australia Communications Conference, Sydney, October, 1992.

8 Appendices

8.1 WORDS IN DATASET

NUMBER	GROUPING	BENGALI	ENGLISH
1	DIGITS	শূন্য	Zero
2		এক	One
3		দুই	Two
4		তিন	Three
5		চার	Four
6		পাঁচ	Five
7		ছয়	Six
8		সাত	Seven
9		আট	Eight
10		নয়	Nine
11	DIRECTION	উপরে	Up
12		নিচে	Down
13		সামনে	Forward
14		পিছনে	Backward
15		ডান	Right
16		বাম	Left
17		উত্তর	North
18		দক্ষিণ	South
19		পূর্ব	East
20		পশ্চিম	West
21	POSITION	প্রথম	First
22		দ্বিতীয়	Second
23		তৃতীয়	Third
24		চতুর্থ	Fourth
25		পঞ্চম	Fifth
26		ষষ্ট	Sixth
27		সপ্তম	Seventh
28		অষ্টম	Eighth
29		নবম	Ninth
30		দশম	Tenth

END