Speech Recognition of Punjabi Numerals Using Convolutional Neural Networks (CNNs)

A Thesis

submitted in partial fulfillment of the requirements for the award of the degree of

Master Of Engineering

in

Department of Computer Science and Engineering

by

Aditi Thakur
(801532002)
Under the supervision of
Dr. Karun Verma
Assistant Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA - 147004
July 2017

Certificate

I hereby certify that the work, which is being presented in the thesis, entitled **Speech Recognition of Punjabi Numerals Using Convolutional Neural Networks (CNNs)**, in partial fulfillment of the requirements for the award of the degree of **Master Of Engineering** and submitted to the institution is an authentic record of my own work carried out during the period **July 2015** to **July 2017** under the supervision of **Assistant Professor Karun Verma**. I have also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

The matter presented in this thesis has not been submitted elsewhere for the award of any other degree or diploma from any institution.

Date:

Aditi Thakur

Aditi Thakur
Candidate

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date:

Dr. Karun Verma
Supervisor
Assistant Professor

The M.E. Viva-Voice examination of **Aditi Thakur**, has been held on 09th August, 2017 .

Acknowledgement

First of all, I would like to express my gratitude towards **Thapar University**, for providing me a platform to do my thesis work at such an esteemed institute.

I wish to express my respect, deep sense of gratitude and indebtedness to my guide **Mr. Karun Verma**, Assistant Professor, Computer Science And Engineering Department, Thapar University, Patiala for their invaluable and enthusiastic guidance, useful suggestions, unfailing patience and sustained encouragement throughout this work.

I would like to thank **Dr. Maninder Singh**, Head of Computer Science And Engineering Department, Thapar University, Patiala for kind help, guidance, encouragement and providing the necessary facilities to carry out my research. I am indebted to the faculty members of the department for valuable suggestions, friendly support and full cooperation rendered by all of them.

Any vote of thanks is not enough to thank the supreme power **"The GOD"** one who has always guided me to work on the right path of the life. Without his grace, this would never come to be today's reality. With special thanks, I dedicate this thesis to GOD.

Last but not the least I would like to thank my family and friends specially Sonia Mittal, Aashima Sharma and Sheena Nanda for their help and support throughout the Thesis.

Aditi Thakur

Abstract

Speech is one of the most natural ways a human interacts and expresses. It is the most convenient form of giving an input to a system. With advancements in technology almost every object that surround humans is slowly progressing towards being automated. This means that in near future almost everything will be controlled using voice or gestures. Slowly and steadily the count of devices and objects that we come across daily in our lives being speech recognizable is increasing like ATMs for visually impaired people and various applications can be supported with speech recognizing system to provide employment opportunities for the differently abled people.

But achieving good accuracy in speech recognition and making the speech recognition system noise robust has always been one of the main concerns of this research area. The model that has dominated the speech recognition field has been GMM-HMM, but with the advancement in the big data field and the computing power, the deep net models have leveraged these gains and used them to outperform GMM-HMM model .But still there is a race of minimizing the error rate.

Achieving accuracy for speech recognition has been a huge obstacle in the domain of Natural Language Processing. The model used predominantly for recognizing speech is GMM-HMM. But with the boom of Deep learning, it has took primacy over the earlier model. With the advancement in the parallel processing and usage of the GPU power, Deep Learning has emanated throughout and has set forth results that has asserted the fact of it outperforming the GMM-HMM.

In this research work we implemented deep learning algorithm - Convolutional Neural network (CNN) with the purpose of achieving good accuracy using the data set. The data is audio data (.wav files) capturing recital of counting from 0 to 100 in Punjabi Language. Data has been targeted to achieve a good balance of male and female speakers. The CNN model architecture comprises of four stack of convolutional layer , ReLU unit and Max pooling unit and further the output from these stacks is passed on to the two fully connected layer . The first fully connected layer has a drop out of 25%. The results obtained from this work has shown better performance as compared to the existing work.

Keywords : Convolutional Neural Network , speech recognition , dropout , pooling, Back Propagation , Gradient Descent

Table of Contents

Title **Page No.**

Certificate . . . . . **i**

Abstract . . . . . **iii**

Table of Contents . . . . . **iv**

List of Figures . . . . . **vi**

List of Tables . . . . . **viii**

Chapter 1 Introduction **1**
1.1 Speech Recognition 1
1.1.1 Types of Speech Recognition 1
1.1.2 Basic Methodology of Speech Recognition System 2
1.1.3 Technical aspect of Speech Recognition System 3
1.2 Application of Speech Recognition 4
1.3 Gaussian Mixture Model –Hidden Markov Model 5
1.4 Neural Network 5
1.4.1 How Do Neural Nets Work? 6
1.5 Deep Learning 7
1.5.1 Types of Deep Neural Nets 7
1.6 Noise Reduction and Silence Removal 9
1.7 Thesis Organization 9

Chapter 2 Literature Survey **10**
2.1 Deep Learning 10
2.2 Silence Removal 15
2.3 Noise Reduction 15

Chapter 3 Problem Statement **17**
3.1 Problem Statement 17
3.2 Research Gaps 18
3.3 Research Objectives 19

Chapter 4 RESEARCH METHODOLOGY **20**
4.1 Data Collection 20
4.1.1 Recording 20
4.1.2 Labelling 23
4.2 Data Pre Processing 25
4.2.1 Noise Reduction 25
4.2.2 Silence Removal 25
4.3 Framing Window 29
4.4 Feature Extraction 29
4.5 Deep Learning Model: CNN 29
4.5.1 Layers Of CNN 30
4.5.2 Tuning Parameters 32
4.5.3 Mechanisms in CNN 32
4.5.4 Normalization Using Activation Function 33
4.5.5 Optimizer 34
4.6 Architecture Used 34

Chapter 5 EXPERIMENTAL RESULTS **36**

Chapter 6 CONCLUSION AND FUTURE SCOPE **40**
6.1 Conclusion 40
6.2 Scope for Future Work 41

References **42**

Appendix **45**
A.1 Tensorflow 45
A.2 LIBROSA 46

List of Figures

Figure No.	Title	Page No.
1.1	Basic Methodology of Speech Recognition System	3
1.2	Basic Neural Network	6
1.3	Basic Convolutional Network Layer	8
3.1	Various approaches in field of AI applied to mimic human brain	18
4.1	Punjabi Numerals	21
4.2	Partition Of Data in Training, Validation and Testing Ratio Samples: 60:20:20	22
4.3	Partition Of Data in Training, Validation and Testing Ratio Samples: 40:20:40	22
4.4	Partition Of Data in Training, Validation and Testing Ratio Samples: 50:20:30	22
4.5	Audacity tool for recording data	23
4.6	This depicts the first step to load a batch of .wav files in Ant Renamer Tool	24
4.7	This depicts the second step to rename the batch by using regular expressions	24
4.8	This depicts the first step to load the batch of .wav file on which the noise reduction is to be performed	25
4.9	This depicts the application of Noise Reduction algorithm : Auto Spectral Subtraction	26
4.10	This depicts the final processing of the audio files for removing the noise	26
4.11	Audio wave shown in the wave pad editor before Noise Reduction	27
4.12	Audio wave shown in the wave pad editor after Noise Reduction	27
4.13	Audio Sound Before Removing Silence	28
4.14	Setting the maximum amplitude to approximately 0.03 for the silence frame	28
4.15	Audio Sound After Removing Silence	29
4.16	Max Pooling Layer	31
4.17	Stack of convolutional layers	35
4.18	Architecture of fully connected and output layer	35
5.1	Comparison of Testing and Validation Accuracy with Noise reduction	37

5.2 Comparison of Testing and Validation Accuracy without Noise reduction 37
5.3 Comparison of accuracies obtained using CNN and GMM-HMM with Noise Reduction 38
5.4 Comparison of accuracies obtained using CNN and GMM-HMM without Noise Reduction 38
5.5 Comparison of Testing Accuracies obtained with or without noise reduction 38
5.6 Graphical representation of reduction in error rate 39

List of Tables

Table No.	Title	Page No.
4.1	Data division into different ratios of Training:Validation:Testing	20
4.2	Convolutional Layer	31
5.1	Results obtained for dataset with Noise Reduction	36
5.2	Results obtained for dataset without Noise Reduction	37
5.3	Comparison of accuracies obtained using CNN and GMM-HMM with Noise Reduction	37
5.4	Comparison of accuracies obtained using CNN and GMM-HMM without Noise Reduction	37
5.5	Reduction in error rate	39

Chapter 1

Introduction

Speech is one of the most natural ways a human interacts and expresses. It is the most convenient form of giving an input to a system. We humans express ourselves in different languages, every language has different style, tone and pronunciations associated with it. The words that are identified in this thesis is numerals from 0 to 100 in Punjabi Language. Punjabi is an Indo-Aryan language which is widely spoken in various parts of the world but predominantly in Indian subcontinent.

In this thesis the model used to recognize spoken Punjabi numerals is a Deep Neural Net. The whole concept of how deep neural net learns to recognize the pattern in the data is termed as Deep Learning. It originated from Machine Learning which is a sub branch of artificial Intelligence. The basic motive is to solve problem that are easy for humans to perform but difficult for them to describe or solve intuitively. Deep learning is based on mainly Artificial Neural network with a concept to mimic the human brain. This thesis revolves around Natural language processing (NLP) (Speech recognition), one of the applications of Deep Learning.

1.1 Speech Recognition

Speech recognition is the mechanism that allows interaction between a human and a system. It is also known as Automatic Speech Recognition. It involves recognizing specific words in a given language. The words that are identified in this thesis are numerals from 0 to 100 in Punjabi Language.

Punjabi is an Indo-Aryan language which is widely spoken in various parts of the world mainly comprising of Indian subcontinent. It is the fourth most spoken language in UK and third most spoken native language in Canada.

1.1.1 Types of Speech Recognition

Types of Speech recognition System:

- Isolated Words
- Connected Words
- Continuous Speech
- Spontaneous Speech

Isolated Words: The speech recognition system which work for isolated words take in input in a format in which each utterance has silence before and after the sample window. It can accept multiple words but the utterance should be one at a time.

Connected Words: These are similar to isolated words but allow utterances with a minimal gap in them.

Continuous Speech: These recognizers are most difficult to create because these need additional functionality to find out the boundaries of the utterances.

Spontaneous Speech: Spontaneous speech itself is a speech that sounds natural and not rehearsed. This type of recognizer should be able to identify natural speech feature along with the utterance boundaries.

1.1.2 Basic Methodology of Speech Recognition System

Traditionally, the basic automatic speech recognition comprises of:

- Raw audio file
- Feature representation
- Acoustic Model
- Language Model

Steps for processing a speech signal by a speech recognition system:

- Initially the audio wave is converted to a feature representation.
- The Acoustic Models responsibility is to learn the relation between the features to the word being spoken.
- The language model encapsulates the knowledge about the language which comprises of what kind of words or what combination of words exist for the language that is getting transcribed.

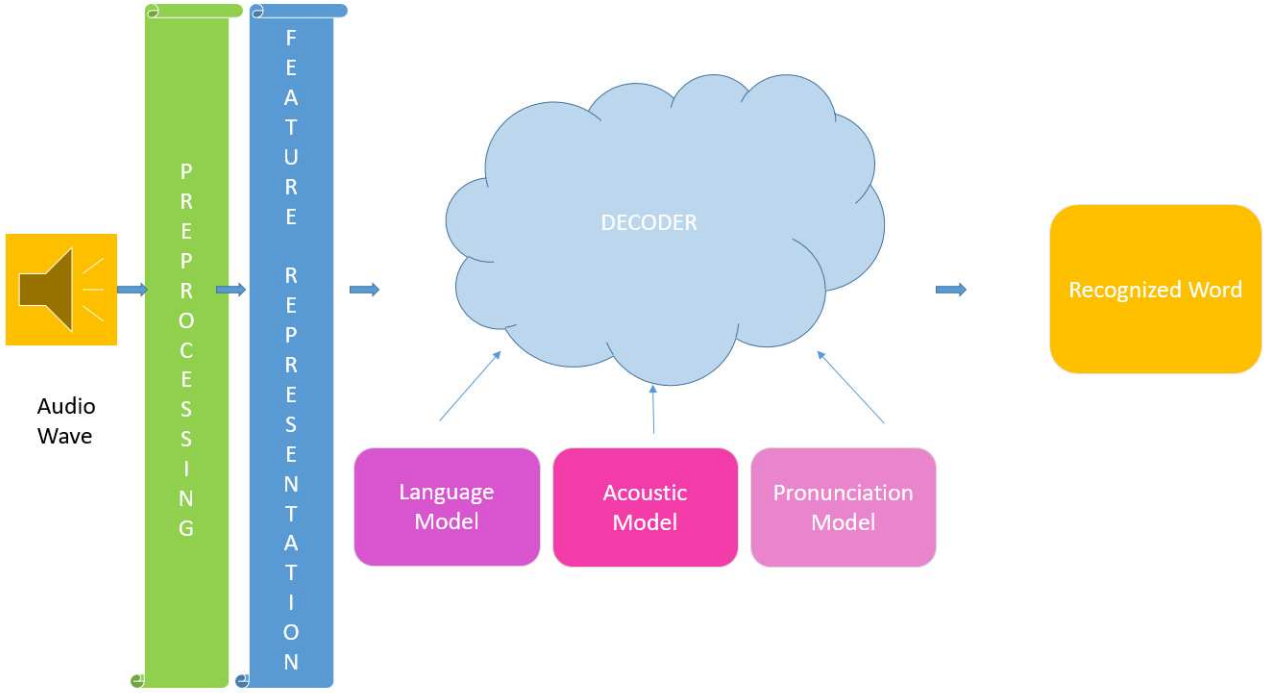


Figure 1.1: Basic Methodology of Speech Recognition System

- The decoder comprises of machine learning algorithms, acoustic model and language model.
- Decoders job is to maximize the accuracy of transcribing the spoken word.

Additionally if phoneme representation is present, it adds complexity to the whole ASR system and another component i.e. Pronunciation model needs to be added to the system.

1.1.3 Technical aspect of Speech Recognition System

From the technical point of view, the goal of speech recognition is to predict the optimal word sequence W, given the spoken speech signal X, where optimality refers to maximizing the a posteriori probability (maximum a posteriori, MAP) :

$$\hat{W} = argmax_W P_{\bigwedge,\Gamma}(W|X), \tag{1.1}$$

where $\bigwedge$ and Γ are the acoustic model and language model parameters. Using Bayes rule

$$P_{\bigwedge,\Gamma}(W|X) = p_{\bigwedge}(X|W)P_{\Gamma}(W)|p(X), \tag{1.2}$$

Equation 2.1 can be re-written as:

$$\hat{W} = argmax_W p_\Lambda(X|W) P_\Gamma(W), \tag{1.3}$$

where $p_\Lambda(X|W)$ is the AM likelihood and $P_\Gamma(W)$ is the LM probability. When the time sequence is expanded and the observations xt are assumed to be generated by hidden Markov models (HMMs) with hidden states θ_t, we have

$$\hat{W} = argmax_W P_\Gamma(W) \Sigma_\theta \Pi_{t=1}^{T} p_\Lambda(x_t|\theta_t) P_\Lambda(\theta_t|\theta_{t-1}), \tag{1.4}$$

where θ belongs to the set of all possible state sequences for the transcription W. The speech signal is first processed by the feature extraction module to obtain the acoustic feature. The feature extraction module is often referred as the front-end of speech recognition systems. The acoustic features will be passed to the acoustic model and the language model to compute the probability of the word sequence under consideration. The output is a word sequence with the largest probability from acoustic and language models. The combination of acoustic and language models are usually referred as the back-end of speech recognition systems.

1.2 Application of Speech Recognition

Historically, ASR applications have been:

- Voice command and control, Voice dialing and Content-based spoken audio search.
- Call routing, Data entry and dictation and structured document creation (e.g., medical and legal transcriptions).
- Gaming, Computer-aided language learning and Robotics.
- Appliance control by voice and Interactive voice response.

Recently there has been advancements in the field of big data and vast improvement in the computing power. This has led to the advancement in Automatic Speech Recognition Technology.

A few examples are:

- Voice search

- Digital assistance and interactions with mobile devices (e.g., Siri on iPhone, Bing voice search and Cortana on winPhone and Windows 10 OS, and Google Now on Android)
- Voice control in home entertainment systems (e.g., Kinect on xBox)
- Machine translation
- Home automation
- In-vehicle navigation and entertainment
- Various speech-centric information processing applications

1.3 Gaussian Mixture Model –Hidden Markov Model

Gaussian Mixture Model(GMM) is mostly used for statistical classification and modelling of data. The advantages of these models is that these can represent complex distributions easily. In speech related research the GMM classifiers are used for the purpose of speech recognition and speaker recognition and noise tracking applications. For the Automatic speech Recognition the GMM is integrated with the HMM.

The speech feature vectors linked with each state of Hidden Markov Model allow modelling the distributions over them. This modelling is done by Gaussian Mixture model. Gaussian Mixture Model –Hidden Markov Model (GMM –HMM) variants have remained the state of art in the field of speech recognition for some time, but with the advancement of the computation power and ability to process big data, the deep learning models have outperformed these models. Also, one of the disadvantage associated with this model is that it is statistically inefficient for modeling data that lie on or near a nonlinear manifold in the data space.

1.4 Neural Network

There is an interconnected web of nodes called neurons and edges that join all the neurons together. In this thesis we are working on a classification problem. Classification is a process of categorizing a group of objects using some basic data features that describe them. There are lots of classifiers available like logistic regression, SVM, Nave Bayes and neural Network. Neural net is a web of classifiers. The firing of a classifier or activation produces a score based on the input received.

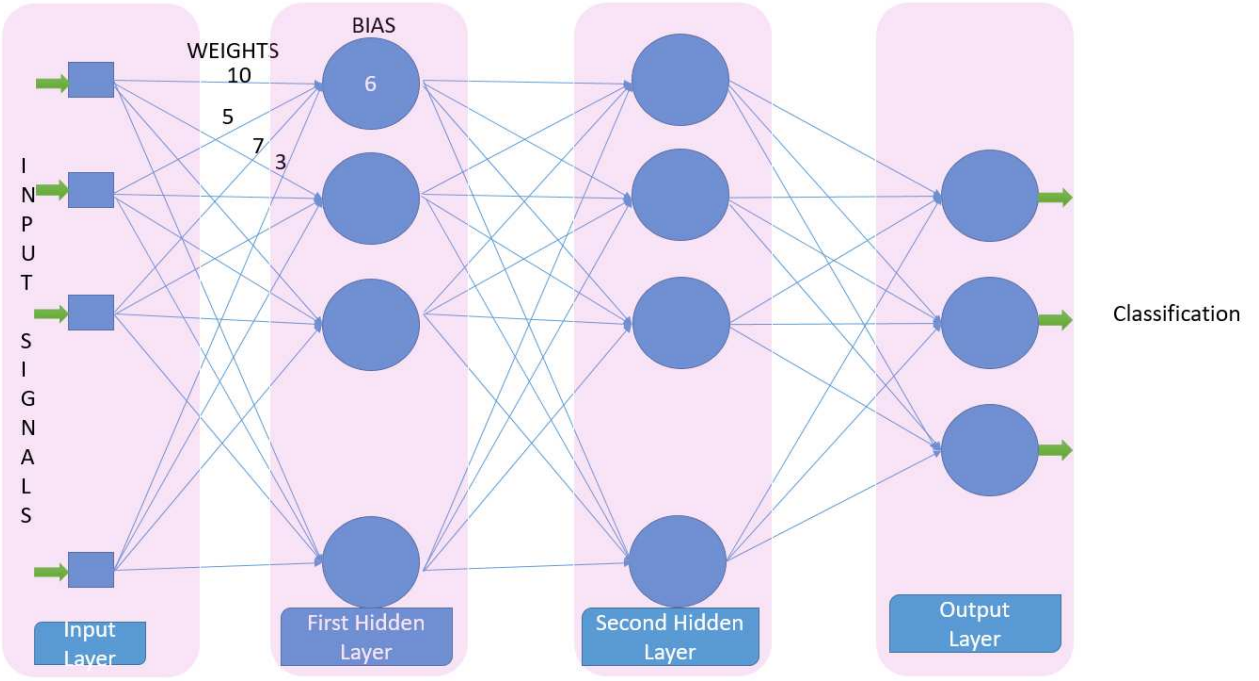


Figure 1.2: Basic Neural Network

1.4.1 How Do Neural Nets Work?

The neural network is highly structured and has many layers. The first layer is the input layer. The final layer is the output layer and all layers in between are known as hidden layers. Each node in the hidden and output layer has its own classifier. When the input is received from the input layer a score is computed by each node in the hidden layer and thus each node fires an activation and passes it on to the next immediate layer. This procedure repeats until output layer is reached. At output layer result of the classification is determined using the final scores at each node. This whole procedure happens for each input node. This series of events starting from the input where the activation is send over the layers and reaches all the way to output layer is termed as Forward propagation. In short, we can say forward propagation is Neural Nets way to classify inputs.

Each set of inputs is modified by unique set of weights and biases. Each edge has a unique weight and each node has a unique bias so this means that the combination used for each activation is also unique so each node fires differently.

The process of improving a neural nets accuracy is called training. To train the neural net the output from the forward prop is compared to the output that is known to be correct and the difference of the above two is known as cost. The point of training is to make that cost as small as possible across millions of training examples. To do this the

neural net tweaks the weight and biases step by step until the prediction closely matches the correct output. Once trained well the neural net has the potential to make accurate prediction each time.

Backpropagation is also a technique used to train Neural Net. It calculates the gradient from the output layers towards the initial layers, moving in the opposite direction of the forward propagation. The training process utilizes gradient. Gradient is the rate at which cost changes with respect to change in weight or bias. Gradient at a layer is the product of all the gradients at prior layer. The fundamental problem that the DNN had faced at its earlier stages was "Vanishing Gradient". When the Gradient is large the net will train quickly but when the gradient is small the net will train slowly. The gradients are much smaller in the earlier layers as a result the earlier layers are slowest to train but these are layers which detect the simplest patterns. So if earlier layers go wrong the resultant layers will be impacted. So backpropagation takes up a lot of time to train the data and accuracy is often low.

1.5 Deep Learning

Deep learning is a re branded concept that has been in presence since 1960. In the past it did not give good results because of the lack of the huge amount of data that is required to train it and the amount of computing power that is required. Now with the technological advancement deep learning is evolving as the state of the art. The deep nets are the modified versions of the neural nets with much more layers and added functionality. There are many types of deep net models available, each one having a signature functionality. The key is that deep nets are able to break the complex pattern into series of simpler patterns. These were inspired by the structure of our human brains. These decipher patterns just like deep nets do in layers. First simple patterns are detected in order to identify the complex pattern as whole.

1.5.1 Types of Deep Neural Nets

- **For Unsupervised Learning:** Restricted Boltzmann Machine(RBM), Auto encoders, etc.
- **For Supervised Learning:** Recurrent Neural Net(RNN), Convolutional Neural Net (CNN), etc.

Unsupervised Learning: to extract patterns from a set of unlabeled data.

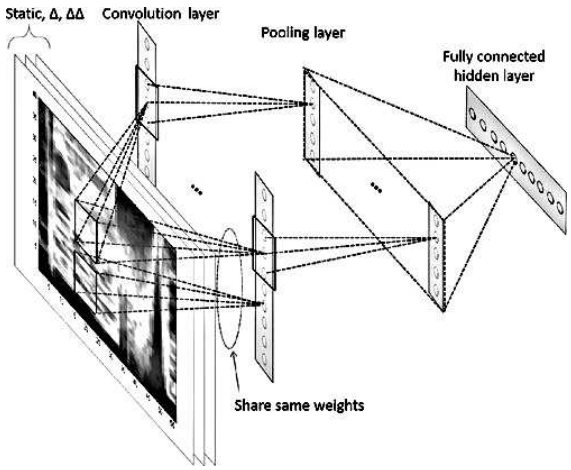


Figure 1.3: Basic Convolutional Network Layer

- **Restricted Boltzmann Machine (RBM):** is a shallow two layer net. The first layer is known as the visible layer and the second layer is known as the hidden layer. Each node in the visible layer is connected to every node in the hidden layer. RBM is considered restricted because no two nodes in a layer are connected within. It is the mathematical equivalent of a two way translator. In the forward pass it takes the input and translates them into a set of numbers that encode the input. In the backward pass it translates them back to form the reconstructed input. Weights and bias play an important role. These help RBM to detect patterns and helps to decide which input feature are the most important ones. (DBN: It is a stack of RBM. It is a formulated solution for vanishing gradient problem.)
- **Autoencoder:** is neural network with 3 layers. In this the output layer is directly connected to input layer for every node. In this model number of hidden units are very less compared to the input output units.

Supervised learning: to predict/classify the unknown input, based on the pattern learned from set of labelled data.

- **RNN:** In this deep net model there are connections between different units and these form a directed cycle and it processes sequential data. It can be said that this model possesses a memory unit because as input it takes the actual input at the current timestamp and the output of the net in the previous timestamp.
- **CNN:** In this deep net model there are many layers in this such as Convolutional Layer, RELULayer, Pooling layer and fully connected layers. This net filters through the image for a specific pattern. It works on data that can be arranged spatially or in a format of image pixels.

1.6 Noise Reduction and Silence Removal

Wavepad is a tool used to remove noise from the audio files. The removal of noise does not affect the quality of the audio but it just reduces the noise component. Wavepad allows removal of noise for batch input.

Silence removal is a technique to remove silence before and after the isolated word. This helps in extracting out just the speech part. The Signal Processing module from the $matlab_R2017a_win64$ complete package has been used to remove silences from the audio clip.

1.7 Thesis Organization

- **Chapter 1:** This chapter introduces basics of speech recognition and deep learning. It also introduces the basic methodology about how an audio signal is processed by a speech processing system. It also describes different types of deep neural networks. It gives a glimpse of comparison of different Approaches followed to mimic Human Brain.
- **Chapter 2:** This chapter presents the main findings of the work that has already been done in the field of Speech Recognition and Deep Learning. Further it also presents some findings of the work related to Noise Reduction and Silence Removal.
- **Chapter 3:** This chapter defines the research problem and presents the current gaps in the domain of research. It also describes the objectives of the thesis.
- **Chapter 4:** This chapter explains about the datasets and the methodology and the details of the model used. It also explains the Architecture that has been followed in order to implement the model.
- **Chapter 5:** In this chapter the results of thesis are discussed. In this the accuracy obtained for different ratios of training and testing are compared for the datasets with and without noise reduction.
- **Chapter 6:** This chapter states the conclusion and the scope for future work.

Chapter 2

Literature Survey

2.1 Deep Learning

Geoff Hinton in University of Toronto was one of the first researcher to device a breakthrough idea for training deepnet. His approach lead to the creation of the RBM and DBN. Because of his pioneering work. He is often referred to as father of machine learning. CNN were pioneered by Yann Legun of NYU.

Looking at the extreme beginning with the birth of perceptron which was the first artificial Neural Network took place in 1958. Rosenblatt defined the linear classifier. The algorithm used by Rosenblatt worked with a perceptron with training set and weights. After the output is obtained it was matched with the correct output and if it did not matched then the weights were decreased otherwise the weights were increased. This perceptron was implemented in custom hardware.

In 1960s Backpropagation was derived and implemented by Seppo Linnainmaa. Paul Werbos analyzed backpropagation in his PhD thesis and proposed that it could be used for backpropagation.

In 1986 David Rumelhart , Ronald Williams and Geoffery Hinton came up with two papers on learning representations by back- propagating errors and learning internal representations by error propagation.

A 1970s survey on speech recognition figured out that the major problem with the speech recognition system has been an interference of the noise via external environment. In 1990s there was a shift to robust speech recognition.

In the year 1988 Rumelhart et al.stated a new mechanism called as back-propagation [1, 2]. This mechanism adjusted the weights of the edges by comparing the actual output and the desired output. And thus this ways the hidden units learn about the data and represent important features.

In 1989 Yann Le Cun et.al at the AT and T Labs demonstrated that one single neural network was able to learn everything in order to recognize a character [3]. It could do

everything from normalizing an image to the classification process. This was the first time the CNN demonstration was done. As explained earlier the initial hidden layers of the neural net were called as convolutional. The neurons were applied in a bunch to subsets of image. Features collected in the previous layer are passed on to next layers and so on. The last layers decide the output.

In the 1990s Neural Net again began to lose hold. Support vector Machines (SVM) and Random Forest began to grow in popularity. Le Cun et al. in 1995 published in his paper Comparison of Learning Algorithms for Handwritten Digit Recognition that SVM was better or equivalent in performance as compared to the neural net[4].

In 1999 Schmidhuber et al. published papers on LSTM describing the inability of learning long-term information due to limitation of backpropagation[5]. RNNs were difficult to train with backpropagation but with a basic idea of LSTM Long Short term memory. It was now possible for RNN to memorize steps that happened thousands of discrete time steps ago.

G. Hinton came up with a paper in 2002 in which he stated that the way to model the complicated and high dimensional data distributions is to use large numbers of simple probabilistic models and to combine the distributions specified by each model [5].

The major breakthrough paper which led to rebranding of the old neural net as "Deep Learning"was published in 2006 by G. Hinton et al.The main findings of the paper were that if weights are initialized in a particular manner and not just randomly, a large neural network can be trained well. The greedy algorithm stated in this paper is used to initialize a slower learning procedure which finely tunes the weights using a different version of wake-sleep algorithm [6]. The model obtained after tuning the parameters performs better at classifying digits than the discriminative learning algorithms.

In 2010 Yoshua bengio et al. came up with a paper about understanding the difficulty of training deep feedforward neural networks [7]. Their main findings were that the selection of a non-linear activation function impacts the performance on a huge scale and the weights should not just be random but the scale should vary according to the layers. These slight changes can be of great help in order to improve the performance.

In 2010 J Schmidhuber et.all proved that the parallel computing approach was very crucial in order to work with huge amount of data and to use huge neural nets [8]. Good amount of improvement in error rate was observed and was presented in the paper regardingDeep Big Simple Neural Nets Excel on Handwritten Digit Recognition.

Piczak, Karol J released a paper in 2015 regarding Environmental Sound Classification

with Convolutional neural network describes the classification of audio clips of environmental sounds using Convolutional Neural Network [9]. The architecture followed by the author comprises of convolutional layer and max pooling layer and fully connected layers. The pair of convolutional layer and max pooling layer has been used twice and 2 fully connected layers have been used. The above model is trained using two channels, the first being the melspectograms values and the second being their delta values. For the purpose of evaluation of accuracy 3 public datasets of audio are used. This model outperforms baseline implementations.

Abdel-Hamidet al. presented a paper in 2014 regarding the use of convolutional neural networks for speech Recognition. This paper describes the hybrid model, which is the combination of Deep Neural Net (DNN) and Hidden Markov Model (HMM) and the error rate considerably reduces for this hybrid model as compared to the conventional Gaussian Mixture Model (GMM)–HMM [10]. This paper also discusses about the limited- weight sharing scheme which can lead to improvement in modelling the speech feature. The data set that has been used here is the TIMIT dataset. The reduction in the error rate be using the hybrid model is observed to be 6% to 10%

In year 2013 Abdel-Hamidet al. presented a paper on exploring convolutional neural network Structures and Optimization Techniques for Speech Recognition. On the phone recognition task ,as of late, convolutional neural networks (CNNs) have been appeared to beat the standard fully connected deep neural networks inside the hybrid deep neural network/hidden Markov model (DNN/HMM) framework in terms of accuracy [11]. This paper has explored CNN in multiple ways by investigating different CNN architectures and it also states that CNN pretraining gives better results on large vocabulary speech recognition task. The architecture is a combination of the variable stacking of different layers and weight sharing i.e. full and limited. The architecture also involves a novel Softmax pooling layer (weighted). This paper states that the used CNN architectures gives better results than the usual DNN on phone recognition and large vocabulary speech recognition tasks. Also the limited weight sharing technique performs better than the full weight sharing architecture.

In year 2012 a paper was presented by G. Hinton et al. regarding the deep neural networks in Speech Recognition. This paper states that a feedforward neural network can be used to evaluate how well each state fits into a frame / frames. The layers takes many frames of coefficients as input and as output gives out the posterior probabilities over HMM states. Traditionally, in order to determine how well every state of each HMM fits into a frame or into a short window of frames of coefficients which represents the acoustic input ,many speech recognition system uses Gaussian Mixture Model (GMM) and in order to

deal with the temporal variability of speech many speech recognition systems use hidden markov Models(HMM) and it can be concluded from this paper that Gaussian Mixture Model (GMM) lack behind the Deep Neural networks (DNN) in terms of performance as the later comprises of many hidden layers that are trained using various methods [12].

Jaitly et al. came out with a paper in 2012 .In this paper results of a DBN-pretrained context-dependent ANN/HMM system trained on two datasets that are much larger than any reported previously with DBN –pretrained ANN/HMM systems - 5870 hours of Voice Search and 1400 hours of YouTube data. The Artificial Neural Network - Hidden Markov Model (ANN/HMM) hybrid outperforms the Gaussian Mixture Model –Hidden Markov Model (GMM/HMM) in case of both the above mentioned datasets [13]. For the first dataset the improvement is seen by 3.7% absolute WER and 4.7% absolute for the second dataset. Maximum Mutual Information (MMI) fine tuning and model combination using Segmental Conditional Random Fields (SCARF) give additional gains of 0.1% and 0.4% on the first dataset and 0.5% and 0.9% absolute on the second dataset [13].

In 2011 Vanhoucke et al. states various techniques that can be used in order to reduce the computational cost on x86 CPUs. This paper provides an effective way to the use of specialized hardware. Running Deep Neural Nets on the modern CPUs can create a computational burden on them as the deep net train on large datasets and deep nets are used with large number of layers. In order to bridge this computational gap the GPUs are used. This paper emphasizes data layout, batching of the computation, the use of SSE2 instructions, and particularly leverage SSSE3 and SSE4 fixed-point instructions which provide a 3 x improvement over an optimized floating-point baseline. In order to show the speedup a hybrid model and large vocabulary system is used [14].

In 2012 a paper was presented by Dahlet al. presents a model i.e. pre-trained deep neural network hidden Markov model (DNN –HMM) hybrid model. It trains the DNN to produce a distribution over senones (tied triphone states) as its output. In order to decrease generalization error one can initialize DBN by pre-training it using the algorithm. This paper also describes the entire procedure for applying CD-DNN-HMM to the large vocabulary speech recognition [15]. The conclusion signifies that Context Dependent –Deep belief Network- Hidden markov model outperforms the context –dependent Gaussian mixture model (GMM)–HMMs. The accuracy improvement of 5.8% and 9.2% (or relative error reduction of 16.0% and 23.2%) over the CD –GMM –HMMs trained using the minimum phone error rate (MPE) and maximum-likelihood (ML) criteria, respectively [15].

In 2009 Larochelleet al. presented a paper on exploring different strategies for the purpose

of training deep neural networks. This paper presents a study of two algorithms for initialization for the gradient based optimization. If initialization is done randomly the gradient leads to bad solutions. Hinton et al. proposed a greedy layer-wise unsupervised learning procedure relying on the training algorithm of restricted Boltzmann machines (RBM) to initialize the parameters of a deep belief network (DBN), a generative model with many layers of hidden causal variables [16]. Another method for initialization is the autoassociator networks.This paper concludes that the greedy layer-wise unsupervised training strategy helps the optimization by initializing weights in a region near a good local minimum.

In the year 2015 Palaz et al. presented a paper regarding the Speech Recognition systems using raw speech as input. This paper stated that automatic speech recognition frameworks normally demonstrate the connection between the acoustic speech signal and the telephones in two separate strides: feature selection and classifier training. In their current works they have demonstrated that, in the structure of convolutional neural systems (CNN), the connection between the crude speech signal and the telephones can be specifically modeled and ASR frameworks focused to standard approach can be assembled. In this paper, they initially analyzed and demonstrate that, between the initial two convolutional layers, the CNN learns (in parts) and models the telephone particular spectral envelope data of 2 –4 ms speech [17]. Given that they demonstrate that the CNN –based approach yields ASR patterns like standard here and now spectral based ASR framework under crisscrossed (loud) conditions, with the CNN –based approach being more powerful.

In the year 2016 Qian et al. presented a paper on noise robust speech recognition systems. Making a system fully noise Robust has always been one of the main goals of the Automatic Speech Recognition research and CNN have been performing to achieve this goal. This paper presents an architecture for CNN for creating a noise robust system. The results show that a 10% reduction in the error rate as compared to the vanilla CNN [18].

In the year 2017 a paper was presented by Zhang et al. for end to end speech recognition systems based on CNN. This paper presents a method to train CNN in a manner which adds more expressive power and better generalization. In order to build deep convolutional and recurrent networks batch normalization, network-in-network principles, convolutional LSTMs and residual connections have been applied. The model is well prepared to handle the overfitting issues. This model uses WSJ ASR task and achieve 10.5% word error rate without any dictionary or language using a 15 layer deep network [19].

2.2 Silence Removal

In the year 2005 Saha et al. presented a paper on Silence removal tecniques. The fundamental steps for preprocessing of speech signal includes Noise removal, Framing and windowing, Endpoint Detection and so forth. Out of the above the essential steps are removing Noise and Endpoint Detection. The method proposed in this paper uses a Linear Pattern Classifier for classification of Voiced part of a speech from silence/unvoiced part of a speech from silence/unvoiced part and Probability Density Function (PDF) of the background noise [20]. The method proposed in this paper presents better end point detection and silence removal than the existing methods.

Atal, B and Rabiner, L presented a paper on classification of speech segment into voiced, unvoiced or silence. This paper describes a pattern recognition approach for figuring out the type for a segment of a speech signal. The segment can be voiced speech, unvoiced speech, or silence. The parameters selected to in order to classify the signal in the above three categories is the speech energy and the zero-crossing rate and the correlation between adjacent speech samples, the first predictor coefficient from a 12 –pole linear predictive coding (LPC) analysis, and the energy in the prediction error [21]. The rule followed in order to assign a segment to a class is based on a minimum distance rule obtained under the assumption that the measured parameters are distributed according to the multidimensional Gaussian probability density function. This paper also presents a smoothing algorithm which provides a smooth 3 –level contour of utterance.

2.3 Noise Reduction

Noise Reduction Based on Modified Spectral Subtraction Method Verteletskaya et al. presented a paper on spectral subtraction method for noise reduction. In this paper method is suggested which is based on spectral subtraction technique. It states the shortcomings of this method and suggests a modified version of this algorithm which overcomes these shortcomings. A technique which allows application of weighted function. This function attenuates frequency spectrum components lying outside identified formants regions [22]. This modified version of algorithm shows results which improves the quality of speech signal significantly than the basic spectral subtraction technique.

In 2011 Fukane et al. presented a paper on noise reduction method. This paper presents review of spectral subtraction algorithm and also discusses the short comings of it. It also states modified versions of spectral subtraction algorithm such as Spectral Subtrac-

tion with over subtraction factor, Nonlinear Spectral Subtraction, Multiband Spectral Subtraction, Minimum mean square Error Spectral Subtraction and Selective Spectral Subtraction and how these are overcome the shortcomings of basic spectral subtraction algorithm [23]. All in all it gives performance analysis of different spectral subtraction algorithms. These algorithms are used to enhance speech signal which gets degraded due to noise. These algorithms help in improving the quality of the sound signal.

Chapter 3

Problem Statement

3.1 Problem Statement

Speech recognition is a huge area of research. With advancements in technology almost every object that surround humans is slowly progressing towards being automated. This means that in near future almost everything will be controlled using voice or gestures. Technology has already reached quiet far in this race. But to a normal human speech recognition has entered in his life through mobile devices. This particular device has marked as the entry pass for a normal person into the world of devices that operate through speech. Slowly and steadily the count of devices and objects that we come across daily in our lives are being speech recognize able is increasing. But achieving good accuracy in speech recognition and making the speech recognition system noise robust has always been one of the main concerns of this research area. Till date different models have been proposed to achieve results and this has led to a significant improvement of accuracy. But still there is a race of achieving a 100% accuracy for data that is nearest to the real world data (without noise reduction).

Achieving accuracy for speech recognition has been a huge obstacle in the domain of Natural Language Processing. The model used predominantly for recognizing speech is GMM –HMM. But with the boom of Deep learning, it has took primacy over the earlier model. With the advancement in the parallel processing and usage of the GPU power, Deep Learning has emanated throughout and has set forth results that has asserted the fact of it outperforming the GMM –HMM. The main objective of this paper is to implement deep learning algorithm - Convolutional Neural network (CNN) with the purpose of analyzing the accuracy of this model using the data set. The CNN model architecture comprises of four stack of convolutional layer , RELU unit and Max pooling unit and further the output from these stacks is passed on to the two fully connected layer . The first fully connected layer has a drop out of 25%.The data is audio data (.wav files) capturing recital of counting from 0 to 100 in Punjabi Language. Data has been targeted to achieve a good balance of male and female speakers. Results for data with and without noise reduction has been analyzed.

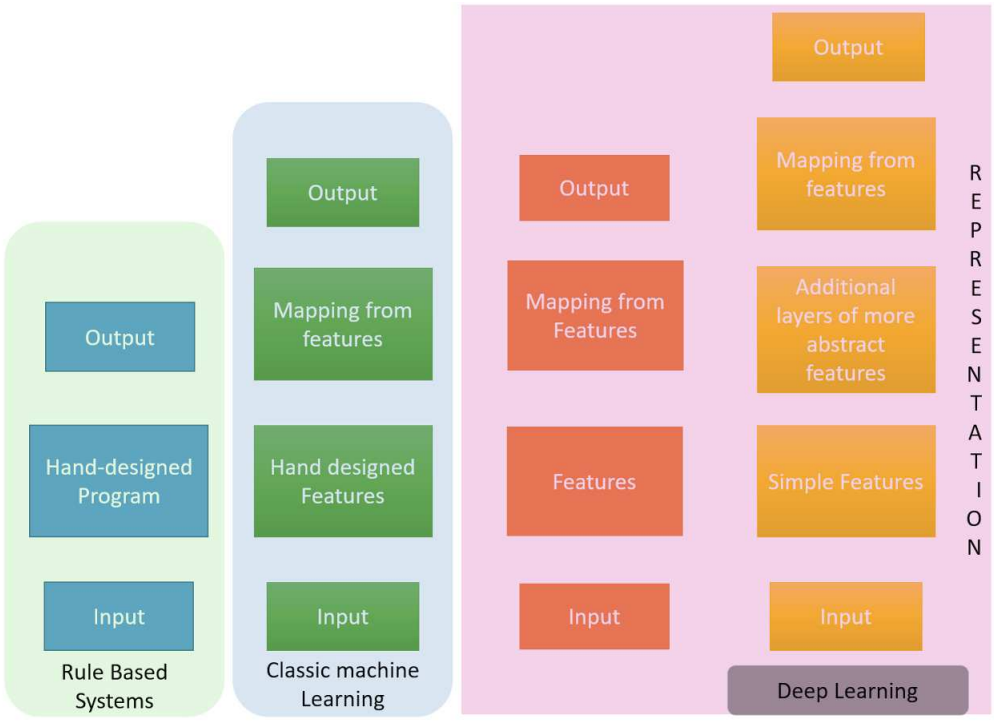


Figure 3.1: Various approaches in field of AI applied to mimic human brain

3.2 Research Gaps

A human being in everyday life requires an immense amount of knowledge about the world. This knowledge is subjective and intuitive both. In order to act in an intelligent way the computers need to have the same amount of knowledge. One of the main challenge is to get this informal knowledge into a computer.

- **Approach 1:** The KNOWLEDGE BASE APPROACH: In this the knowledge is hardcoded
- **Approach 2:** THE MACHINE LEARNING APPROACH: The system having ability to extract patterns from raw data. The performance depends on the representation of the data that it is given(features).
- **Approach 3:** THE DEEP LEARNING APPROACH: Sometimes it is difficult to know which features need to be extracted. And it is also difficult to capture data by a system how exactly the human brain capture it. It is difficult to capture high-level abstract feature from data.

In case of speech recognition some minute factors such as speakers accent using algorithms which are equivalent to human level understanding of data are difficult to capture. Deep learning allows the computer to build complex concepts out of simpler concepts. In

this era the success of Deep Learning is more due to the availability of more data and the availability of more powerful computational engines like GPU (Graphic Processing Units).

How do deep Net recognize these complex patterns: The key is that deep nets are able to break the complex pattern into series of simpler patterns. The deep nets were inspired by the structure of our human brains. Deep nets decipher patterns just like our brains do in layers. At first simple patterns are detected inorder to identify the complex pattern as whole.

For example: A neural net has to identify whether an image contains a human face or not. A deep net would first use edges to detect different parts of face. And then further combine the result together to form the whole face.

The major gap that need to be covered:

- Firstly,to improve the accuracy in recognizing speech for Punjabi dataset.
- Secondly, we need to work with a model such that it can take input data nearest to the real world scenario (data with noise) and can still give good accuracy.

3.3 Research Objectives

- To design and implement a model that improves the accuracy for recognizing words compared to existing models being used [24].
- To improve the error rate for speech signals without noise reduction so that the speech recognition system can be as close to the real world scenarios.

Chapter 4

RESEARCH METHODOLOGY

4.1 Data Collection

The major problem with the Deep Nets is that in order to perform well these need huge amount of data and in thus a great computational power is needed to process this much data. There are no publicly available data set of the recitation of 0 to 100 in Punjabi so the dataset preparation is a task itself using tools like Audacity, Wavepad and Matlab for signal processing. Two data sets are prepared. The first data set contains audio files with noise reduction and the second data set contains audio files without noise reduction.

The data is divided into different ratios to compare the performance. It is divided into:

- **Training:** The data that is actually used to train the model
- **Validation:** The data that is used to validate the model or to test how well the model is built with the tuned parameters.
- **Testing:** This is the data that is used for testing the model.

4.1.1 Recording

The first data set comprises of audio recording of the spoken words i.e. counting from 0 to 100 in Punjabi. This recording has 101 samples for each person. Data for 15 males and 15 females is recorded in order to achieve a good balance. The total samples obtained are $101 \times 30 = 3030$. These audio files are recoded using Plantronics PLNAUDIO478 Stereo

Table 4.1: Data division into different ratios of Training:Validation:Testing

Training:Validation:Testing	Training Samples	Validation Samples	Testing Samples
60/20/20	1818	606	606
50/20/30	1515	606	909
40/20/40	1212	606	1212

Word	Phonemes	Word	Phonemes	Word	Phonemes
cipher	s ih ph a r	painti	p eh n t ih	sattar	s a tt a r
ikk	ih k	chhatti	ch a t ih	ikhttar	ih k a h a tt a r
do	d ow	sainti	s eh n t ih	bahttar	b a h a tt a r
tinn	t ih n	athtti	a th t ih	tihattar	t ih h a tt a r
char	ch ah r	untali	u n t ah l ih	chuhattar	ch u h a tt a r
panj	p ah n j	chali	ch ah l ih	panjhattar	p a ch a tt a r
chhe	ch eh	iktali	ih k t ah l ih	chhihattar	ch ih h a tt a r
satt	s ah tt	batali	b a t ah l ih	satattar	s a t a tt a r
atth	ah th	tartali	t a r t ah l ih	athattar	a th a tt a r
naum	n ow	chutali	ch ow t ah l ih	unasi	u n ah s ih
das	d a s	pantali	p a n t ah l ih	assi	a s ih
giaram	g ih ah r ah	chhiali	ch ih ah l ih	ikasi	ih k ah s ih
baram	b ah r ah	santali	s a n t ah l ih	biasi	b ih ah s ih
teram	t eh r ah	athtali	a td a t ah l ih	tariasi	t a r ih ah s ih
chaudam	ch ow d ah	unanja	u n a n j ah	churasi	ch u r ah s ih
pandram	p a n d r ah	panjah	p a n j ah	pachai	p a ch ah s ih
solam	s ow l ah	ikvanja	ih k v a n j ah	chhiasi	ch ih ah s ih
sataram	s a t ah r ah	bavanja	b a v a n j ah	satasi	s a t ah s ih
atharam	a th ah r ah	tarvanja	t a r v a n j ah	athasi	a th ah ih
unni	u n ih	churanja	ch u r a n j ah	unanve	u n ah n v eh
vih	v ih	pachvanja	p a ch v a n j ah	nabbe	n a b eh
ikki	ih k ih	chhapanja	ch a p a n j ah	ikanvem	ih k ah n v eh
bai	b ah ih	satvanja	s a tt v a n j ah	banvem	b ah n v eh
tei	t eh ih	athvanja	a th a v a n j ah	tirianavan	t ih r ih ah n v eh
chauvi	ch ow v ih	unahat	u n ah a th	churranavan	ch u r ah n v eh
pachi	p a ch ih	satth	s a th	pachannaven	p a ch ah n v eh
chhabbi	ch ah b ih	ikahath	ih k ah a th	chhiannaven	ch ih ah n v eh
satai	s a t ah ih	bahath	b ah a th	satannaven	s a t ah n v eh
athai	a th ah ih	trehat	t r eh ah a th	athannaven	a th ah n v eh
untti	u n t ih	chaunhat	ch ow ah a th	narhinnaven	n a td ih n v eh
tih	t ih	pehant	p eh ah a th	sau	s ow
iktti	ih k t ih	chhehat	ch eh ah a th		
batti	b a t ih	satahat	s a t ah a th		
teti	t eh t ih	atahat	a th ah a th		
chaunti	ch ow n t ih	unattar	u n a tt a r		

Figure 4.1: Punjabi Numerals

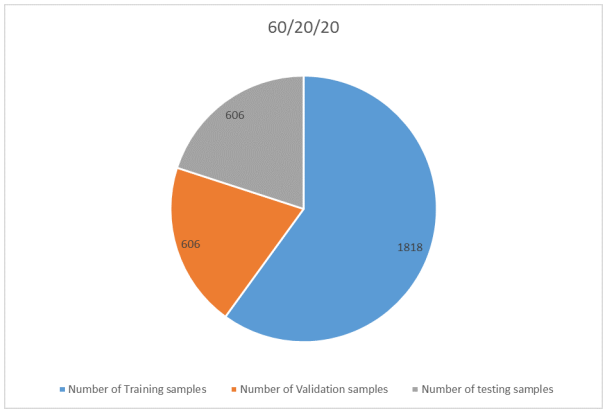


Figure 4.2: Partition Of Data in Training, Validation and Testing Ratio Samples: 60:20:20

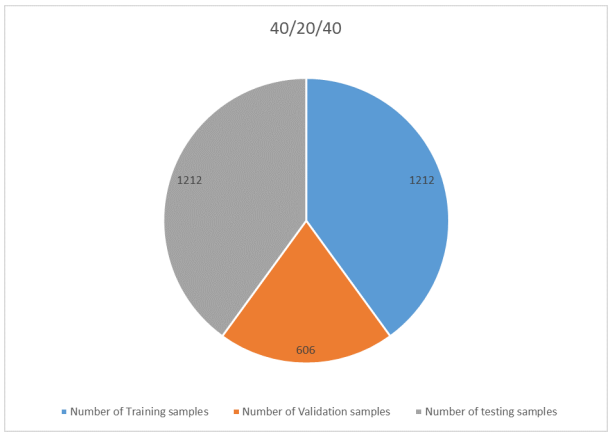


Figure 4.3: Partition Of Data in Training, Validation and Testing Ratio Samples: 40:20:40

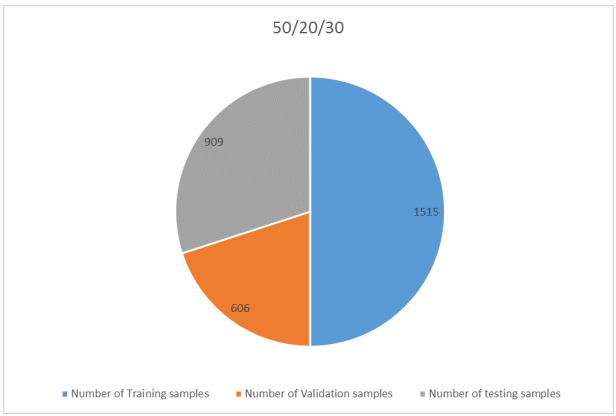


Figure 4.4: Partition Of Data in Training, Validation and Testing Ratio Samples: 50:20:30

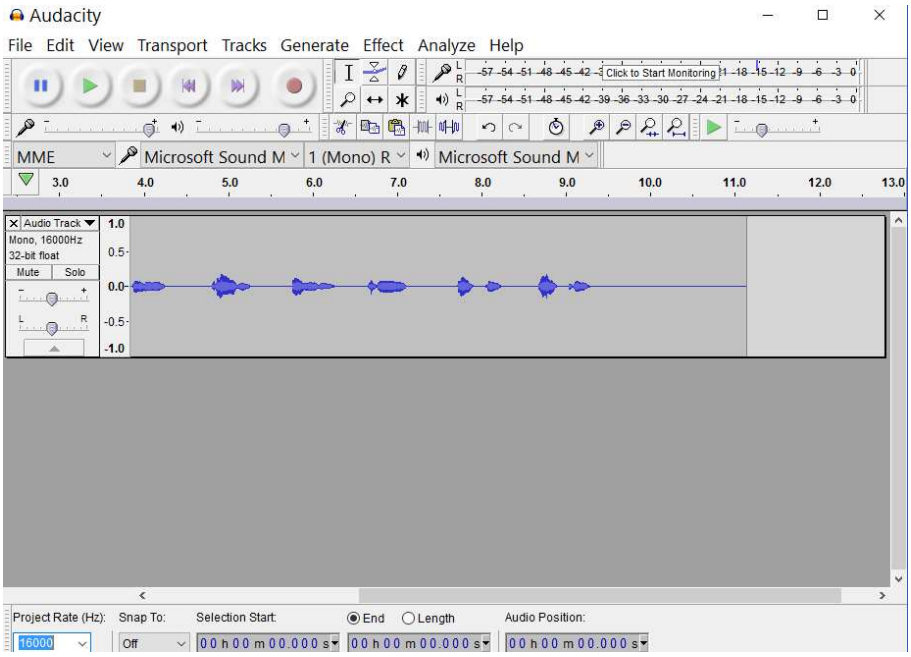


Figure 4.5: Audacity tool for recording data

mic and audacity software. The sample frequency rate is set to 16 KHz. In Figure 4.4 a waveform is obtained for Punjabi words with the help of Audacity Software.

4.1.2 Labelling

In order to label the data a very simple approach has been followed. The names of the audio files are picked up as the labels . The digit after the second dash i.e. $-$ is the correct label for an audio file. The labels are assigned from the name of the audio itself hence, each audio is named in this fashion. And then the labels have been hot encoded. One hot encoded is a group of bits which represent a unique label . These are combination of values which comprise of single high bit and the other bits are low. Eg : $108041 - 0 - 3 - 1.wav$: The digit after the second dash is - is 3 . So the audio wav for the word tinn has been labelled as 3. To aid the labelling the process Ant-renamer 2.12 software is used.

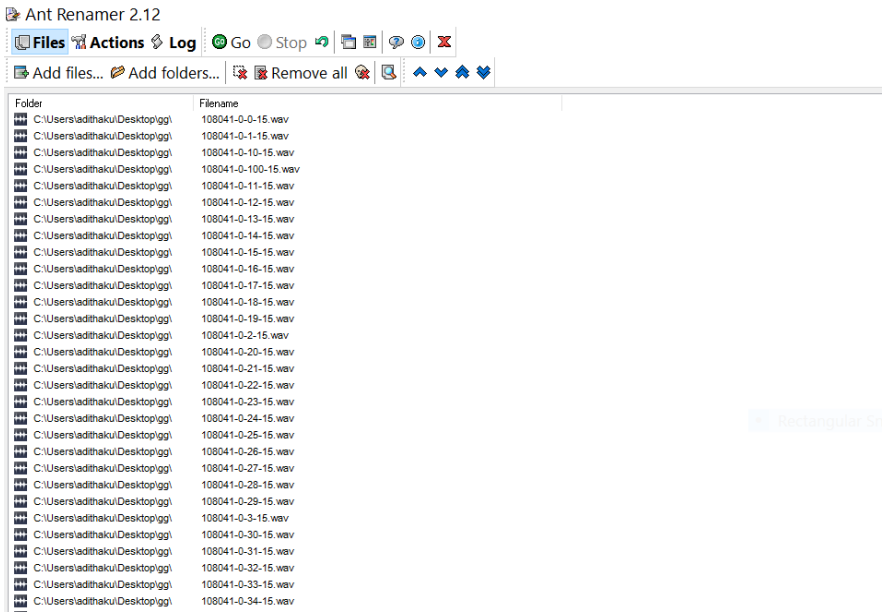


Figure 4.6: This depicts the first step to load a batch of .wav files in Ant Renamer Tool

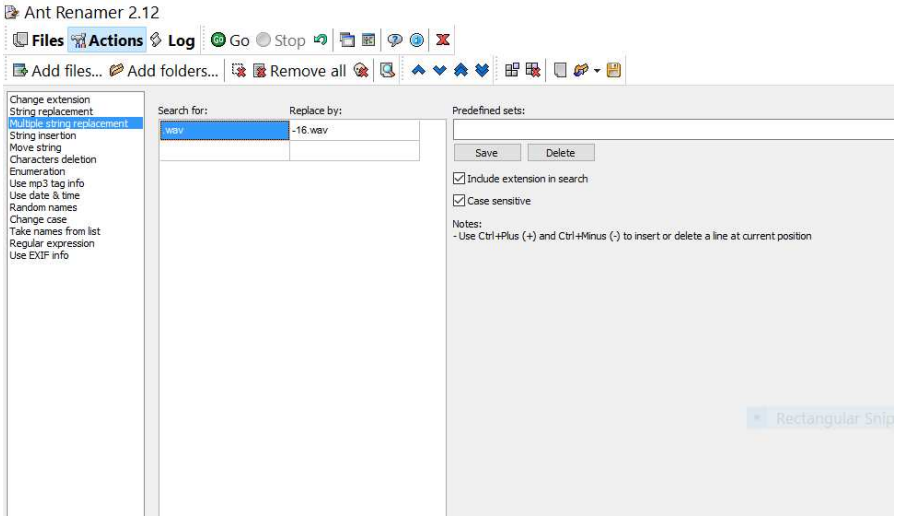


Figure 4.7: This depicts the second step to rename the batch by using regular expressions

4.2 Data Pre Processing

4.2.1 Noise Reduction

Wavepad is used to remove noise from the audio files. The removal of noise does not affect the quality of the audio but it just reduces the noise component. Wavepad allows removal of noise for batch input.

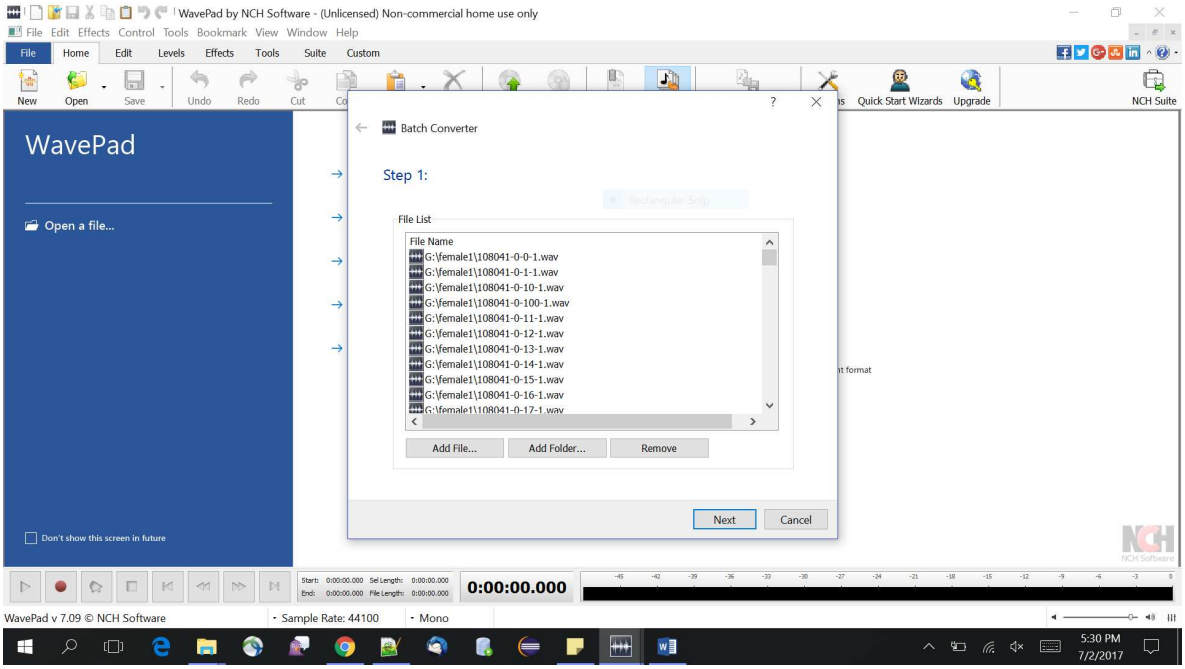


Figure 4.8: This depicts the first step to load the batch of .wav file on which the noise reduction is to be performed

4.2.2 Silence Removal

For the purpose of removing silence in audio files End- point detection is a technique of identifying speech parts in audio signal. Inputs which contain speech surrounded by silence are easier to work with in terms of energy comparison. Silence removal is a technique to remove silence before and after the isolated word. This helps in extracting out just the speech part.

Classifying labeling events in speech using 3 state representation:

- **Silence:** where no speech is produced

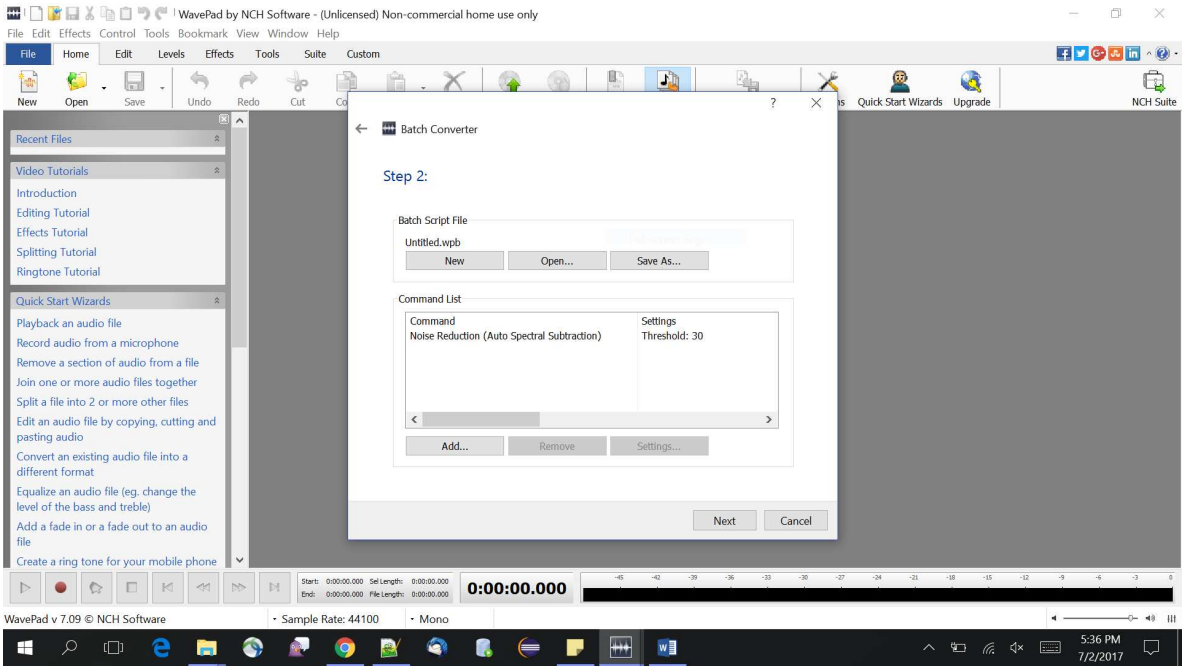


Figure 4.9: This depicts the application of Noise Reduction algorithm : Auto Spectral Subtraction

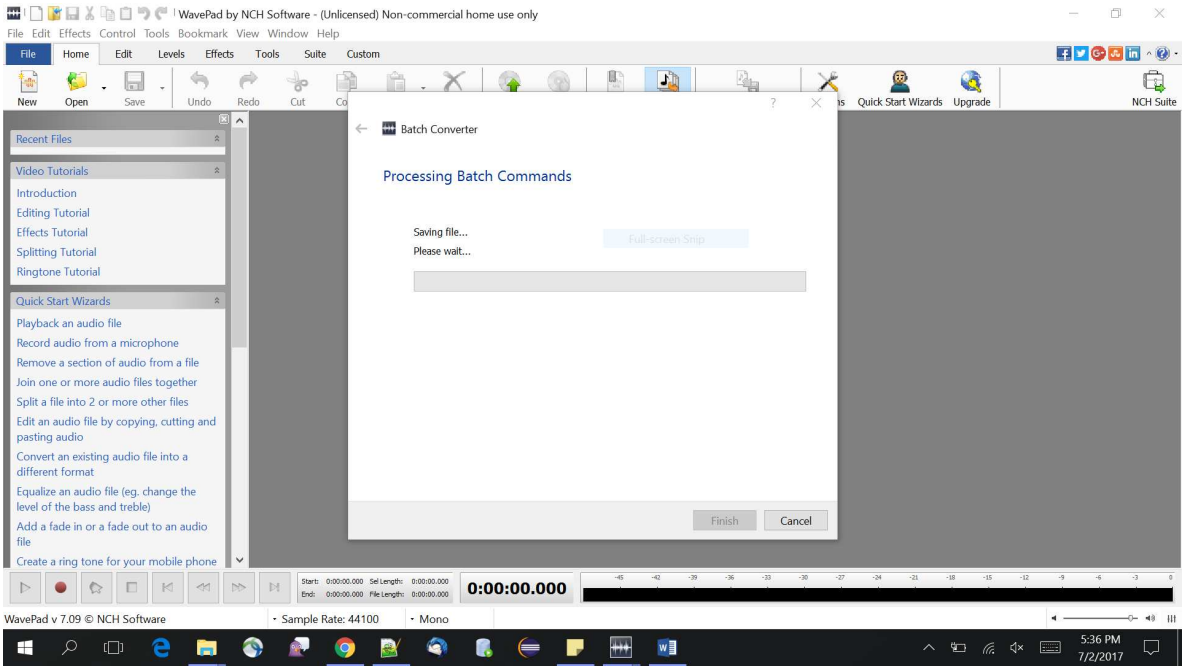


Figure 4.10: This depicts the final processing of the audio files for removing the noise

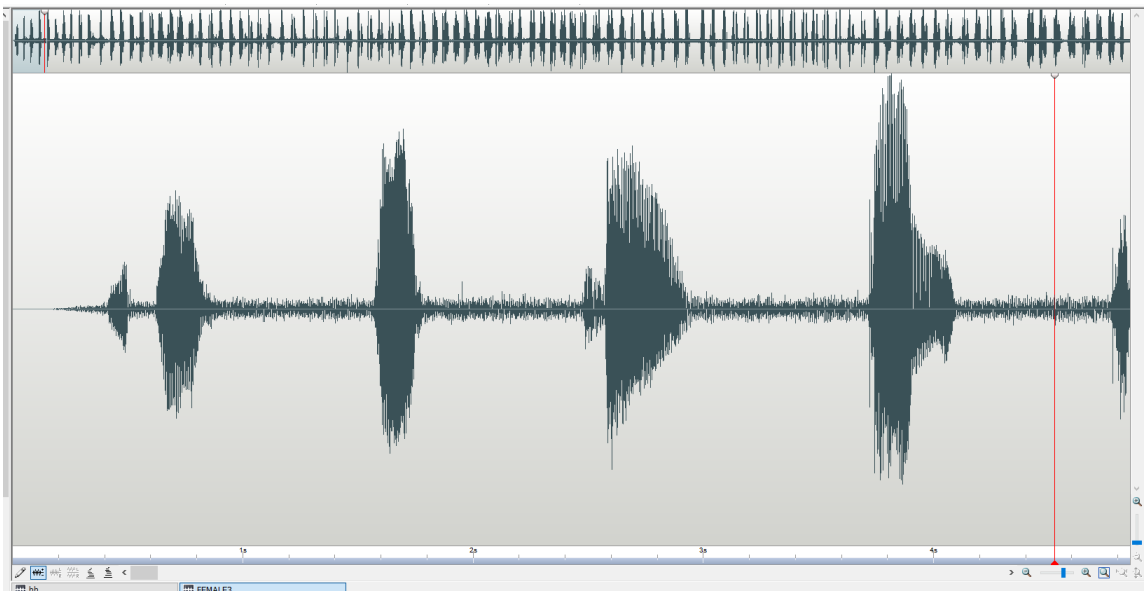


Figure 4.11: Audio wave shown in the wave pad editor before Noise Reduction

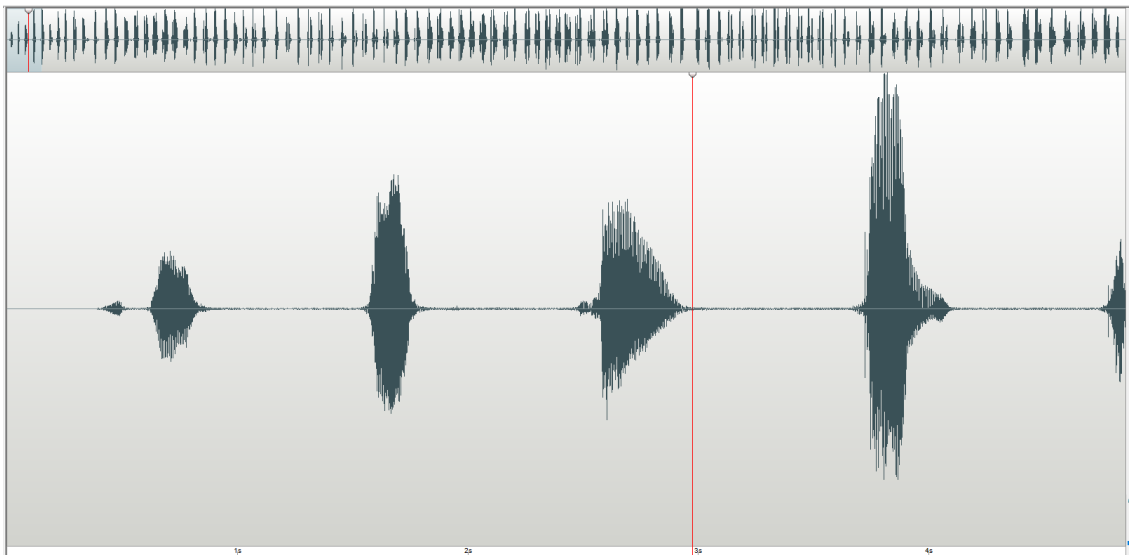


Figure 4.12: Audio wave shown in the wave pad editor after Noise Reduction

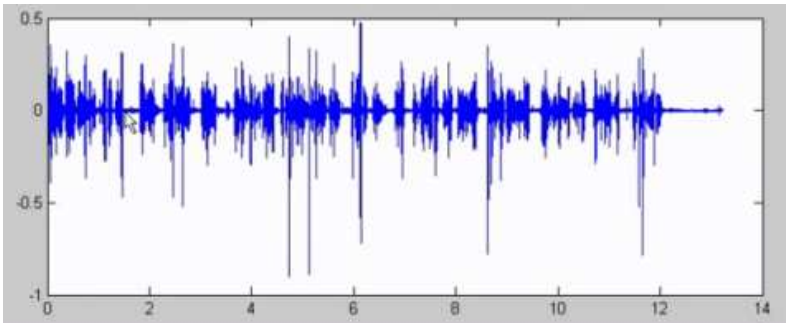


Figure 4.13: Audio Sound Before Removing Silence

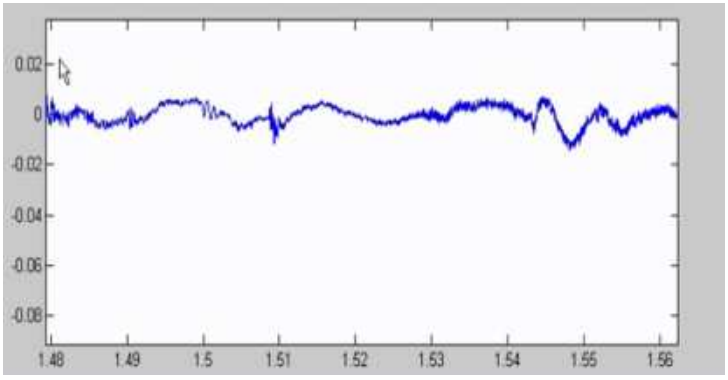


Figure 4.14: Setting the maximum amplitude to approximately 0.03 for the silence frame

- **Unvoiced:** where vocal cords are not vibrating. This is aperiodic and random in nature
- **Voiced:** where vocal cords are vibrating. This is almost periodic in nature.

Silence and Unvoiced are classified together because these result in low energy content.

The Signal Processing module from the matlab_R2017a_win64 complete package has been used to remove silences from the audio clip.

In order to remove silence, break the signal into frames of 0.1 seconds. Then identify silence by finding frames with max amplitude less than 0.03(This can be deduced from figure 2 in table above). At last create a new signal which does not contain silent frames.

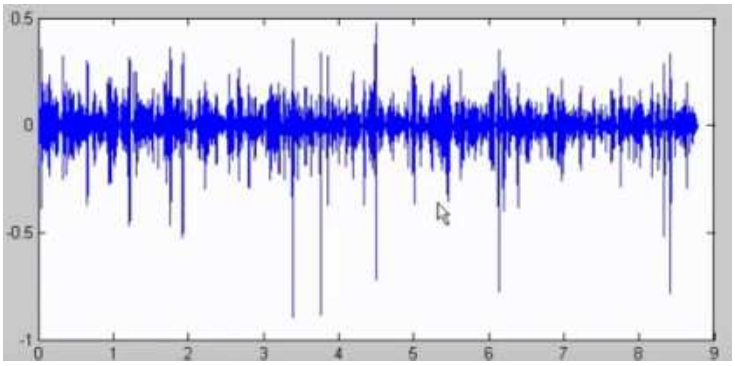


Figure 4.15: Audio Sound After Removing Silence

4.3 Framing Window

Librosa package has been used for the audio analysis. This package provides some core functionality like loading audio files and provides various methods for analyzing audio. The audio has been loaded and decoded as time series, which is represented as a one-dimensional Numpy floating point array. Time series are decomposed in terms of sinusoid presence. The sampling rate by default is approximately 22 KHz but it can be set by setting the parameter sr = 16000 (16 KHz). The sampling rate is defined as the number of samples of audio carried per second. For each audio clip the sound wave has been windowed into a size of 15360 reading from the complete one dimensional floating point array. Because of windowing we need to take overlapping segments to make up for the attenuated parts of the input.

4.4 Feature Extraction

For the selected window size melspectogram is computed and then mapped directly onto the mel scale. Melspectogarm is a representation of power spectrum of sound. Then further logamplitude of the melspectoram is computed. This is done to scale the mel-spectogram in a stable format.

4.5 Deep Learning Model: CNN

CNN are quiet similar to basic neural Networks. These have input, output layers and hidden layers. CNN works efficiently for image recognition but has also shown good results for speech recognition. The input that is fed to the CNN consists of $[width \times$

$height \times channel$]. As the dataset is audio so instead of width and height there are bands and frames [$bands \times frames \times channel$]. The channel for images is of value 2 for black and white and for colored it is 3 (Red, blue and green). In this case of audio two channels are used. One being the melspectogram values and the other being its delta values. If any other feature needs to be added it can be done by simply adding another channel. The input and output is defined in a tensorflow placeholder.

A layer stack is created using Convolution, Rectified linear Unit layer and Pooling layer (These layers are defined in detail in the next section).The output of one layer becomes the input of other. This stack can be repeated multiple times thus leading to deep stacking. Here in this architecture 4 stacks are used. Each time the image get more filtered as it goes through convolution layers and it keeps on getting smaller as it goes through pooling layers.

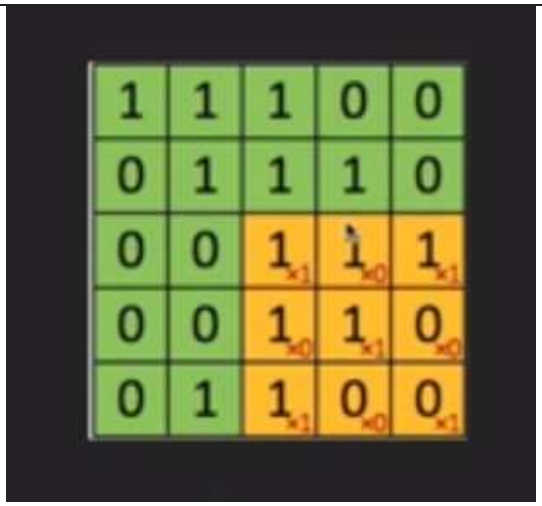
The final layer is the fully connected layer. Here every value gets a vote on what the outcome is going to be. The input for this layer is the filtered and reduced size of the stack of vectors. These are flattened out and rearranged in a single list so that it becomes easy to visualize. In the end the values present here are the probabilities or values corresponding to each class that helps in identifying to which class a particular input belongs to. It can be said that this is the final voting layer.

4.5.1 Layers Of CNN

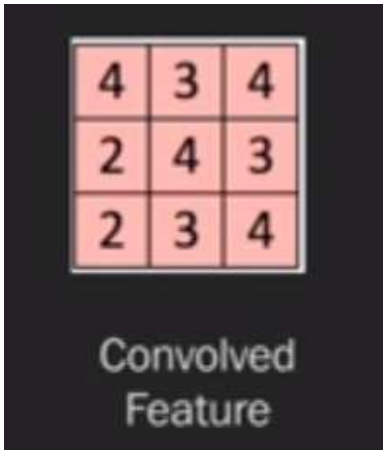
- Convolutional Layer
- Pooling Layer
- Fully Connected Layer

Convulational Layer: Convolution are a sequence of sliding and projection operations. The sliding is defined by the displacement and the projection is the net product between two functions .For the case of vectors the filter slides on and that filter dot product along with the actual data values is taken and as a result we obtain the convolved feature. The method that has been used in our model is conv2D as the sound data is depicted as a two –dimensional data. Before feeding the input tensor into the convolution layer the weights and biases values are set. The first convolutional layer of the first stack receives the input tensor and the weights. The output of this is added to the bias and is then fed into the RELU layer. The output from this function is then passed on to the pooling layer. In case of a one dimensional:

Table 4.2: Convolutional Layer



3×3 kernel size window used over 5×5 size convolutional layer



Resultant feature map

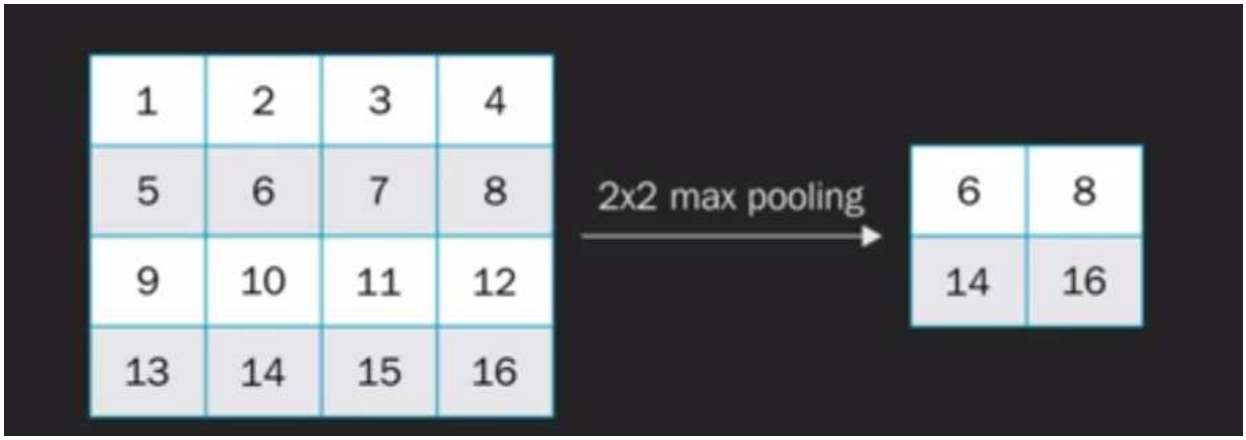


Figure 4.16: Max Pooling Layer

Pooling Layer: is a way to scale down a vector. The pooling that has been used here is max pooling. In max pooling the largest number is taken in a vector neighborhood. The largest numbers in a convolved vector are those where the convolutional filter forms the best matching inputs. From this point of view it can be said that convolutions contribute to feature matching and pooling gives the best match detection. The combination of both convolutional layer and pooling layer is what gives CNN a great shift invariant representations. As input the pooling layer needs the output from the conv and RELU layer and the filter size and the stride size. The filter window is the sliding window with which the dot product is taken and the stride value is the amount by which the filter window slides. This layer completes the stack . It can be further passed on to the next layer or the fully connected layer.

Fully Connected Layer: This layer is the basic neural network layer that is fully connected. This layer further calculates the activation function. The output of this layer is a symbolic variable (tensor flow variable).

4.5.2 Tuning Parameters

For the convolution layer the number of features and the size of the features can be tuned in. For the Pooling layer the window size and the stride size can be chosen. For the fully connected layers the number of neurons can be tuned. The other choices that are present while designing your own CNN are number of each type of layer required and the order in which these layers should be stacked. Particularly in case of sound we can say we have time steps or intensity in each frequency band instead of pixels for the analogous purpose. So in that case the frequency bands that are closer together are more closely related and when this converted into a tensor it looks exactly the same as a tensor for 2D image would appear.

4.5.3 Mechanisms in CNN

Forward Propagation: is the way the data propagates in the forward direction .The input data is provided to the first convolutional layer. Before processing the data the weights and biases are set for each node and edge. Further the data is processed and the output is provided to the RELU unit. This unit normalizes the values present in the output vector by converting the negative values to zero. The output from this layer is further pushed to the max pooling layer. The output from this layer is passed on to the next stack of convolutional, RELU, max- pool layer or it can also be passed directly to the fully connected layer. This layer gives out the final values associated with each class. The last unit is the Softmax layer which converts the final values into a range of 0 to 1 and the sum totaling to 1 giving final result as probabilities for each class.

Backpropagation: The principle behind backpropagation is that the error in the final answer is used to determine how much the network needs to adjust. This error drives a process called gradient descent. It is calculated from the output layers towards the initial layers, moving in the opposite direction of the forward propagation. The training process utilizes it. Gradient is the rate at which cost changes with respect to change in weight or bias. Gradient at a layer is the product of all the Gradient at prior layer. The fundamental problem that the DNN had faced at its earlier stages was Vanishing Gradient. When the Gradient is large the net will train quickly but when the gradient

is small the net will train slowly. The gradients are much smaller in the earlier layers as a result the earlier layers are slowest to train. These are layers detect the simplest patterns.

Dropout: It is a method of regularization in which randomly selected neurons are ignored during training. This implies their contribution to the activation of downstream neurons is temporally neglected on the forward pass and any weight updates are not applied to the neuron on the backward pass.

Now if neurons are randomly dropped out during training, so in order to cope up with this situation other neurons will have to step in and handle the representation that is necessary to make predictions for the missing neurons.

The size of the values being propagated forward needs to be increased. This has to be done in proportion to the number of values being turned off. As a result the network becomes less sensitive to the specific weights of neurons and leads to a network that most probably will not over fit the training data and gives better generalization. In order to perform dropout on a layer set some of the values randomly to zero. Drop out is only done during training. According to a paper by G. Hinton tuning of dropout should be done with respect to tuning the size of hidden layer. Until the data fits perfectly turn off the dropout and increase the hidden layer size. Then turn the dropout on and train with the hidden layer size as set in previous step. Finally turnoff the dropout as soon as the training is over.

4.5.4 Normalization Using Activation Function

- Rectified Linear Unit
- Softmax

RELU: This Computational unit performs normalization by removing all negative values and changing it to 0. The RELU function is f(x) =max(x, 0) these help in speeding up the training process. RELU is computed after the convolution layer. It is a nonlinear activation function and it also does not suffer from the vanishing gradient problem. But if the learning rate is set too high then the RELU units might die if a large gradient is flowing through the RELU unit hence, the learning rate needs to be tuned.

Softmax: It is used as a classifier at the end of the neural Network. The output of each unit is squashed in between a range of 0 to1 and along with that it sets all outputs in a manner so that the sum total of the outputs is equal to 1. The output of this Softmax

function gives us the probabilities of each class being true for an input.

4.5.5 Optimizer

Adam Optimizer: is a replacement optimization algorithm for stochastic gradient descent for training deep learning models.

4.6 Architecture Used

The CNN model being used here has a convolutional layer followed by a pooling layer. These two layer are repeated in sets of four. The output of the convolutional layer is first passed on to the rectified linear unit function. The flow runs in a manner where we have first conv layer then RELU function and then pooling layer. And then the output of the last pooling layer is passed on to the fully connected layer. Then the output is further passed on to fully output layers.

Basic Structure:

Convolution Layer and Max Pool Layer Stack
Convolution Layer 1 > RELU > Max Pool Layer 1:
The first layer uses 30 filters with the size of 20×20
Convolution Layer 2> RELU > Max Pool Layer 2:
The second layer uses 50 filters with the size of 20×20
Convolution Layer 2 > RELU > Max Pool Layer 2:
The third layer uses 70 filters with the size of 20×20
Convolution Layer 2 > RELU > Max Pool Layer 2:
The fourth layer uses 90 filters with the size of 20×20

Fully Connected Layer Stack: Two fully connected layers have been used. The first layer has a dropout of 25% and the second layer is used has no dropouts.

Output Layer: The output layer maps the output to the 101 classes (0 -100) and feeds the final output to the softmax Unit which converts the final output values to a range of 0 to 1 and sum total of 1 thus representing the outputs as the probabilities for the predicted class.

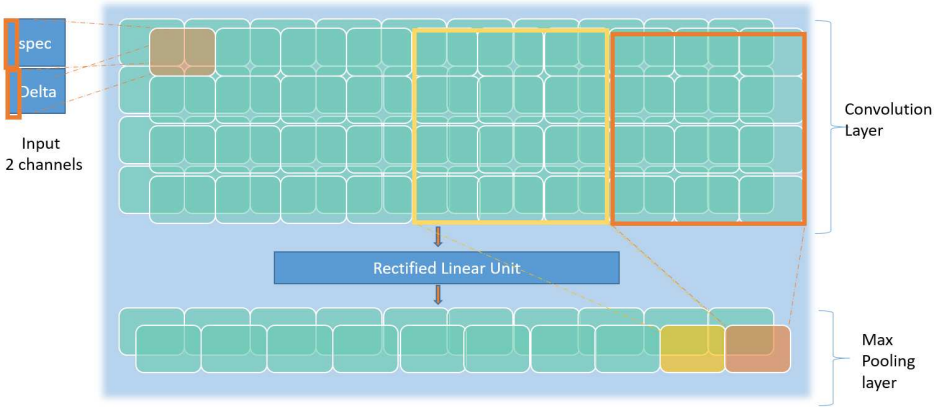


Figure 4.17: Stack of convolutional layers

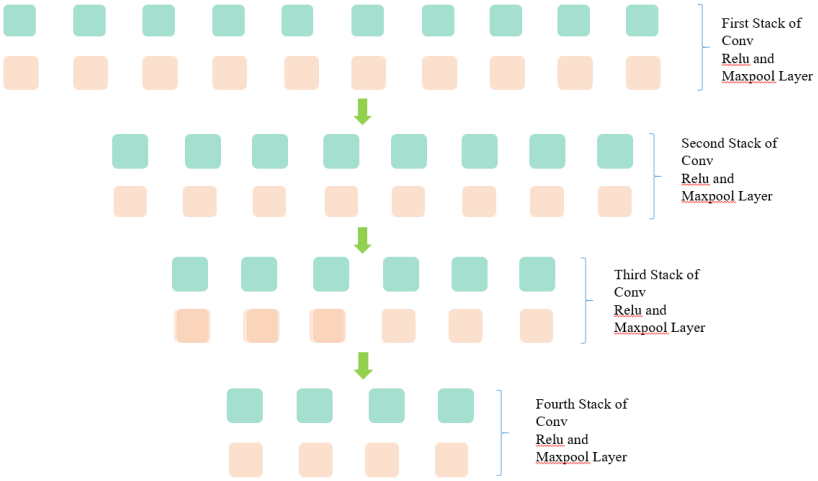


Figure 4.18: Architecture of fully connected and output layer

Chapter 5

EXPERIMENTAL RESULTS

We collected 3030 samples of audio files by recording recital of Punjabi numerals from 0 to 100. In order to obtain an equal balance 15 male and 15 female speakers were selected. Then this data was fed to CNN in order to recognize the speech. The following results were drawn from the study.

- Table 5.1 represents the division of data into training, testing and validation. It also shows the different testing and validation accuracy for different division of data with noise reduction. The same is depicted using a bar graph in Figure 5.1.
- Table 5.2 represents the division of data into training, testing and validation. It also shows the different testing and validation accuracy for different division of data without noise reduction.The same is depicted using a bar graph in Figure 5.2.
- Table 5.3 represents the comparison of the results obtained using CNN (my work) and GMM –HMM (existing work) for the data with noise reduction.The same is depicted using a bar graph in Figure 5.3.
- Table 5.4 represents the result comparison of the results obtained using CNN (my work) and GMM –HMM (existing work) for the data without noise reduction.The same is depicted using a bar graph in Figure 5.4.
- Table 5.5 represents the error improvement rates for the data with and without noise reduction.The same is depicted using a bar graph in Figure 5.6.
- Figure 5.5 represents the comparison of test accuracies obtained with and without noise reduction.

Table 5.1: Results obtained for dataset with Noise Reduction

Training:Validation:Testing Samples	Training Samples	Validation Samples	Testing Samples	Testing Accuracy	Validation Accuracy
60/20/20	1818	606	606	92.43%	94.12%
50/20/30	1515	606	909	88.66%	92.07%
40/20/40	1212	606	1212	84.38%	88.49%

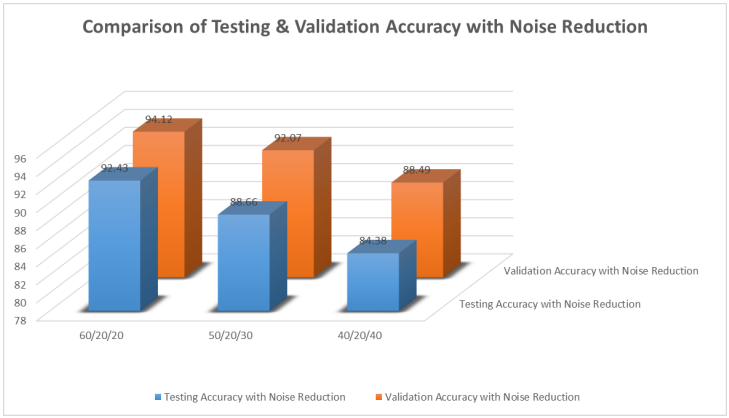


Figure 5.1: Comparison of Testing and Validation Accuracy with Noise reduction

Table 5.2: Results obtained for dataset without Noise Reduction

Training:Validation:Testing Samples	Training Samples	Validation Samples	Testing Samples	Testing Accuracy	Validation Accuracy
60/20/20	1818	606	606	88.56%	85.39%
50/20/30	1515	606	909	83.01%	81.67%
40/20/40	1212	606	1212	78.93%	80.23%

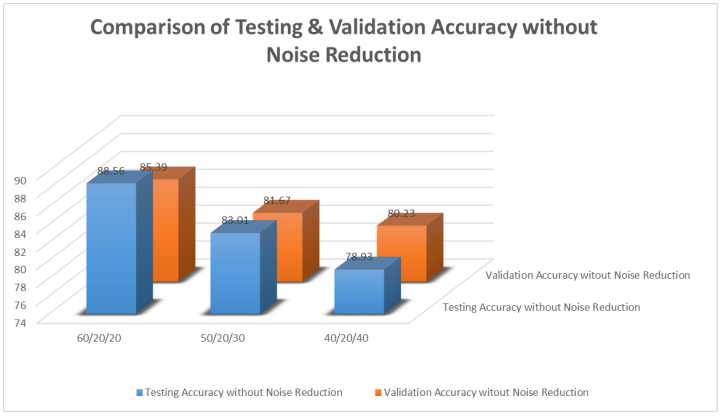


Figure 5.2: Comparison of Testing and Validation Accuracy without Noise reduction

Table 5.3: Comparison of accuracies obtained using CNN and GMM-HMM with Noise Reduction

Model	Accuracy	Error
$GMM - HMM$	89.20%	10.80%
CNN	92.43%	7.57%

Table 5.4: Comparison of accuracies obtained using CNN and GMM-HMM without Noise Reduction

Model	Accuracy	Error
GMMHMM	84.80%	15.20%
CNN	88.56%	11.44%

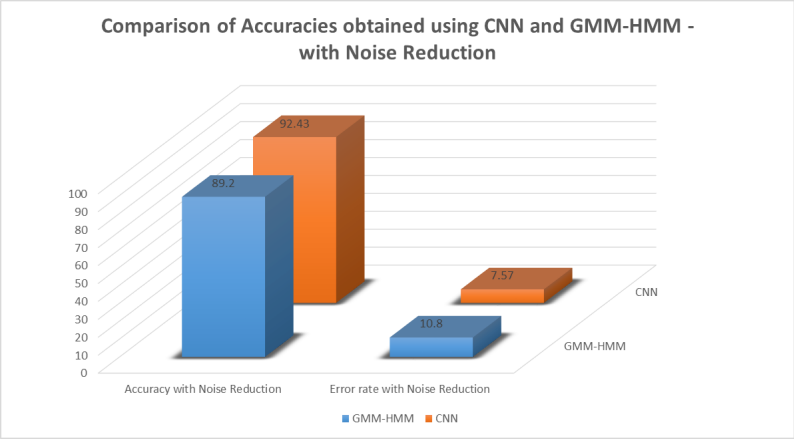


Figure 5.3: Comparison of accuracies obtained using CNN and GMM-HMM with Noise Reduction

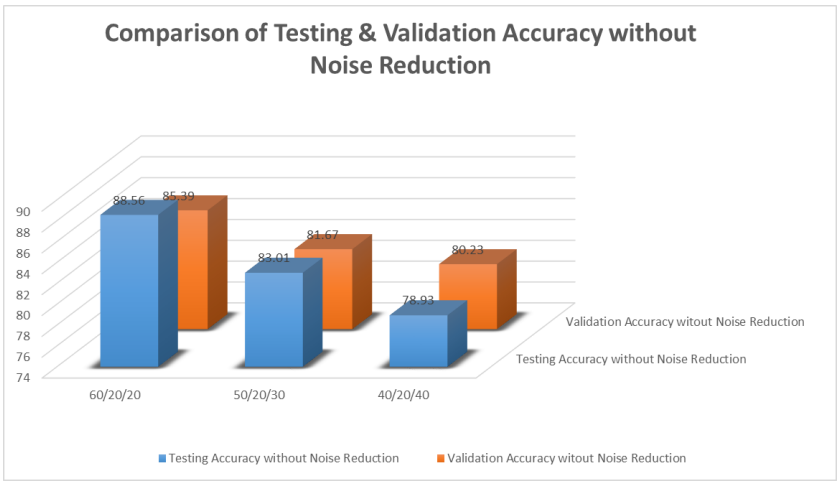


Figure 5.4: Comparison of accuracies obtained using CNN and GMM-HMM without Noise Reduction

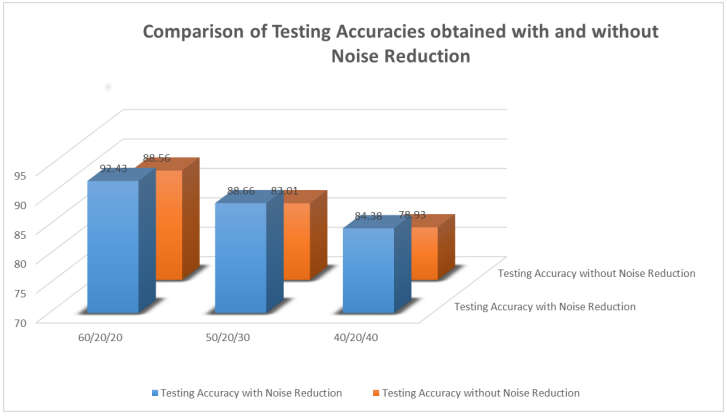


Figure 5.5: Comparison of Testing Accuracies obtained with or without noise reduction

Table 5.5: Reduction in error rate

Recognition	Reduction in error rate
With noise reduction	3.23%
Without noise reduction	3.76%

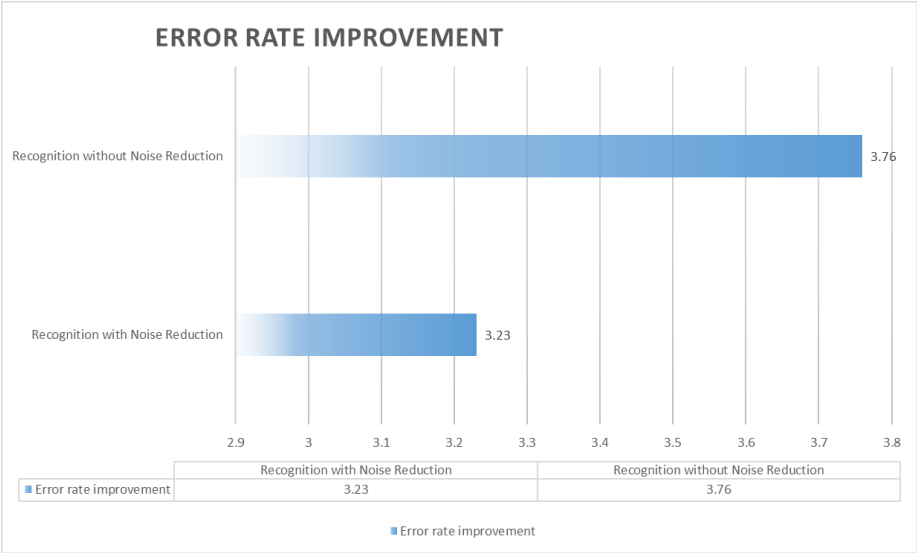


Figure 5.6: Graphical representation of reduction in error rate

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

From the results section we can infer that:

- The CNN model performs better than the GMM –HMM model for the present dataset.
- The reduction in the error rate in case of the dataset without noise reduction is greater than the dataset with noise reduction. So we can conclude that it can perform well with data that is near to real world (without noise reduction).
- When the pattern get really complex the deep net starts to outperform all other algorithms. The extra baggage is that deep nets take much longer to train, but the GPU can train these neural nets faster following the parallel processing. Deep Networks best for pattern recognition.
- To recognize simple patterns a basic classification tool like SVM or logistic regression is good enough ,but when data has moderate patterns neural net over perform others. But when patterns get more complex the number of node required in each layer grows exponentially with the number of possible patterns in the data. Eventually training becomes way too expensive and the accuracy starts to suffer. So for patterns that are intricate basic neural nets are not good enough. The only choice that is practical to use is deep net.
- Convolutional networks capture local "spatial"patterns in data. If the data cannot be made to look like an image Convolutional Nets are less useful. These are great at finding patterns but if the data remains useful after swapping any of the columns with each other than the convolutional net cannot be used as it does not form any pattern. So Convolutional Nets can work well with image and anything that has a pattern or can be represented in that format such as for speech signal processing and sentence processing.

6.2 Scope for Future Work

- Increasing the Accuracy: The gap between the current accuracy and the maximum accuracy can be bridged by expanding the structure of the architecture and deploying it on the system which is more computation power. This can also be done by trying out variation of this model with other deep net model in a hybrid format.
- Increasing the Dataset: This model can be implemented on a greater dataset and thereby tuning the parameters to achieve better results.

References

[1] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[2] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[3] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[4] Y LeOu11, L Jackal, L Bottou, A Brunet, C Cortes, J Denker, H Drucker, I Guyon, U Miiller, E Séckinger, et al. Comparison of learning algorithms for handwritten digit recognition.

[5] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[6] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[7] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[8] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010.

[9] Karol J Piczak. Environmental sound classification with convolutional neural networks. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pages 1–6. IEEE, 2015.

[10] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[11] Ossama Abdel-Hamid, Li Deng, and Dong Yu. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Interspeech*, pages 3366–3370, 2013.

[12] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The

shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[13] Navdeep Jaitly, Patrick Nguyen, Andrew W Senior, and Vincent Vanhoucke. Application of pretrained deep neural networks to large vocabulary speech recognition. In *Interspeech*, pages 2578–2581, 2012.

[14] Vincent Vanhoucke, Andrew Senior, and Mark Z Mao. Improving the speed of neural networks on cpus. In *Proc. Deep Learning and Unsupervised Feature Learning NIPS Workshop*, volume 1, page 4, 2011.

[15] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2012.

[16] Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. Exploring strategies for training deep neural networks. *Journal of Machine Learning Research*, 10(Jan):1–40, 2009.

[17] Dimitri Palaz, Ronan Collobert, et al. Analysis of cnn-based speech recognition system using raw speech as input. Technical report, Idiap, 2015.

[18] Yanmin Qian, Mengxiao Bi, Tian Tan, and Kai Yu. Very deep convolutional neural networks for noise robust speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12):2263–2276, 2016.

[19] Yu Zhang, William Chan, and Navdeep Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 4845–4849. IEEE, 2017.

[20] G Saha, Sandipan Chakroborty, and Suman Senapati. A new silence removal and endpoint detection algorithm for speech and speaker recognition applications. In *Proceedings of the 11th national conference on communications (NCC)*, pages 291–295, 2005.

[21] B Atal and L Rabiner. A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(3):201–212, 1976.

[22] Ekaterina Verteletskaya and Boris Simak. Noise reduction based on modified spectral subtraction method. *IAENG International journal of computer science*, 38(1):82–88, 2011.

[23] Anuradha R Fukane and Shashikant L Sahare. Different approaches of spectral subtraction method for enhancing the speech signal in noisy environments. *International Journal of Scientific & Engineering Research*, 2(5):1, 2011.

[24] Shweta Mittal and Karun Guide Verma. *Speaker Independent Isolated word speech to text conversion using HTK*. PhD thesis, 2014.

[25] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, YangqingJia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[26] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

Appendix

A.1 Tensorflow

TensorFlow is an interface for expressing machine learning algorithms, and an implementation for executing such algorithms [25]. It is compatible with variety of operating systems and various computational devices with CPU and GPU support both. The system is flexible enough and can be used to express a wide variety of algorithms. It supports a wide variety of algorithms. It can not only be used for conducting research but also deploying machine learning systems ML into production across many fields.

Tensorflow can be installed onUbuntu, Mac OS X and Windows. We have installed tensorflow on Ubuntu. There is also a choice available with CPU support and GPU support. There are four mechanisms by which tensorflow can be installed:

- Virtualenv
- native pip
- Docker
- Anaconda

In order to implement this work we installed tensorflow on Ubuntu with CPU support and the selected mechanism was the Native pip method. The prerequisite for this installation are python (2.7 or 3.3+) and pip. Install TensorFlow:

- Pip install tensorflow
- Pip3 install tensorflow
- Pip install tensorflow-gpu
- Pip3 install tensorflow-gpu

If the above command fails install the latest version of tensorflow in the following manner:

- Sudo pip install -upgrade tfBinaryURL
- Sudo pip3 install upgrade tfBinaryURL

Where tfBinaryURL is the URL of the Tensorflow Python package. The value of this URL depends on the various factors mentioned above such as GPU support, Python

version and operating system.

For installing tensorflow in Linux machine with CPU support only and with Python version 2.7 this command can be used.

- **TFPYTHONURL = https://storage.googleapis.com/tensorflow/linux/cpu /tensorflow-1.1.0-cp27-none-linux_x86_64.whl**

A.2 LIBROSA

Librosa is python package for audio and music signal processing[26]. At a high level, librosa provides implementations of a variety of common functions used throughout the field of music information retrieval.

To install librosa the following commands can be used:

- pip install librosa or
- Sudo pip install librosa Or to install System − wide
- pip install u librosa