4-2022

Impact of Noise in Automatic Speech Recognition for Low-Resourced Languages

Vigneshwar Lakshminarayanan
vl1255@rit.edu

Impact of Noise in Automatic Speech Recognition for Low-Resourced Languages

Vigneshwar Lakshminarayanan

Impact of Noise in Automatic Speech Recognition for Low-Resourced Languages

Vigneshwar Lakshminarayanan

April 2022

A Thesis Submitted
in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in
Computer Engineering

RIT | Kate Gleason College of Engineering

Department of Computer Engineering

Impact of Noise in Automatic Speech Recognition for Low-Resourced Languages

Vigneshwar Lakshminarayanan

Committee Approval:

Prof Dr. Emily Prud'hommeaux, *Advisor* Date
Department of Computer Science, Boston College

Prof Dr. Alexander Loui, *Committee Member* Date
Department of Computer Engineering

Prof Dr. Andres Kwasinski, *Committee Member* Date
Department of Computer Engineering

Acknowledgments

I am grateful to Dr.Emily Prud'hommeaux for her guidance, encouragement and continued support. I would like to extend my gratitude to Dr. Alexander Loui and Dr.Andres Kwasinski for being on my thesis committee. I also want to thank Dr. Raymond Ptucha, Robert Jimmerson for their advice and invaluable suggestions.I am also thankful to my family and friends, whose support has been instrumental in my life.

Abstract

The usage of deep learning algorithms has resulted in significant progress in automatic speech recognition (ASR). The ASR models may require over a thousand hours of speech data to accurately recognize the speech. There have been case studies that have indicated that there are certain factors like noise, acoustic distorting conditions, and voice quality that has affected the performance of speech recognition. In this research, we investigate the impact of noise on Automatic Speech Recognition and explore novel methods for developing noise-robust ASR models using the Tamil language dataset with limited resources. We are using the speech dataset provided by SpeechOcean.com and Microsoft for the Indian languages. We add several kinds of noise to the dataset and find out how these noises impact the ASR performance. We also determine whether certain data augmentation methods like raw data augmentation and spectrogram augmentation (SpecAugment) are better suited to different types of noises. Our results show that all noises, regardless of the type, had an impact on ASR performance, and upgrading the architecture alone were unable to mitigate the impact of noise. Raw data augmentation enhances ASR performance on both clean data and noise-mixed data, however, this was not the case with SpecAugment on the same test sets. As a result, raw data augmentation performs way better than SpecAugment over the baseline models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automatic Speech Recognition (ASR) technologies have been progressively used in many modern applications like Amazon Alexa, Apple's Siri, Microsoft's Cortona [1]. This has been possible due to the emergence of deep learning architectures and the advancement in computation methods and the large amounts of data available for languages like English and Mandarin [2]. But even with these advancements, the performance of ASR models have been poor and fragile when exposed to certain factors like noise, acoustic distorting conditions, voice-quality. The performance of these models worsens when trained with low-resource languages.

The motivation for this thesis is to research on the impact of several types of noise like continuous noises, punctuated noises, background noises on various ASR models like the GMM-HMM acoustic model, SGMM acoustic model, and the DNN model using a low-resource Tamil language dataset and find out how much each type of noise impacts the performance of ASR. Additionally, we evaluate whether certain data augmentation approaches, such as raw data augmentation and spectrogram augmentation (SpecAugment), are particularly well suited to various types of noises, thereby minimizing the noise impact on the low-resource dataset. We demonstrate the above models on the low-resource Tamil language dataset provided by SpeechOcean.com and Microsoft for a low-resource ASR challenge in Interspeech 2018 [3]. We observe that all types of noises regardless of the acoustic model architecture impact the ASR

performance. In the case of data augmentation techniques, raw data augmentation outperforms SpecAugment [4] on the clean data as well as the noise-mixed data.

1.1 Objectives

The objectives of the thesis are as follows:

- To investigate how different types of common environmental noises differentially impact ASR quality. We consider two different dimensions: (1) continuous vs punctuated noise (e.g., a running tap vs. a door slamming); and (2) machine-made noise (mechanical) vs. noise produced by a living being(Non-mechanical) [e.g., a running tap vs. background party chatter].
- To explore the relationship between different data augmentation techniques and ASR quality. In particular, we want to understand whether generic data augmentation methods perform as well as targeted data augmentation methods.

1.2 Contributions

The main contributions of this thesis are outlined below:

- Providing researchers with information about how different types of noise differentially impact ASR quality.
- We offer suggestions for future researchers on how to do data augmentation to specifically target the kinds of noise that might be expected in some particular application (e.g., in a car vs. in a personal assistant).

1.3 Document Structure

The reminder of the document is structured as follows: The Chapter 2 reviews the background of ASR in low-resource languages, the general Acoustic and Neural Net-

work models, the Language models, the Kaldi toolkit, evaluation metrics, types of noises, and related works. Chapter 3 provides a detailed description of the speech dataset used for this thesis. Chapter 4 describes about the approach/methodology towards finding the impact of several types of noises in ASR models using a low resource language, data preparation for data augmentation and how data augmentation helps in improving the performance of the ASR models in noisy environments.Chapter 5 discusses the results and various experiments conducted to find the impact of noise and also how data augmentation works on different kinds of noises.Chapter 6 provides the final conclusions and also talks about the future scope.

Chapter 2

Background

Automatic Speech Recognition (ASR) is the process of translation of the audio provided by the user into text via a developed software program. There are four basic components in most ASR frameworks: an acoustic feature extraction method; an acoustic model; a language model, and a decoder.

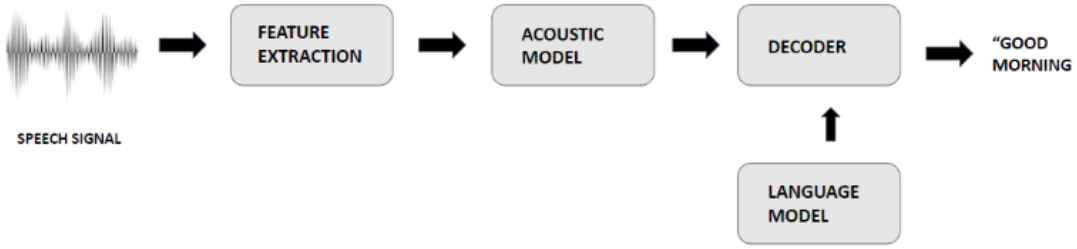

Figure 2.1: This figure shows the general pipeline of an Automatic Speech Recognition System

The speech signal is passed through the feature extraction step where all the features like Mel-Frequency Cepstral Coefficients Features (MFCC) [5], Feature Space Maximum Likelihood Linear Regression (fMLLR) [6], and spectrograms of the raw input waveform are extracted in short time-frames. The feature vectors are passed into the acoustic model which makes sure that these features complement with the phoneme classes and the models are trained and thereby passed to decoding to predict

the series of text or characters. The decoding with the help of a language model which contains the dictionary of the concerned language is used accordingly to predict the final output text/characters.

In the following sections, we explain how each component is trained and used in the pipeline. We then discuss evaluation metrics for ASR. We introduce the various types of noises we will explore in this thesis. Finally, we explore some of the popular works in Automatic Speech Recognition.

2.1 Feature Extraction

Feature extraction is used to extract features from the raw audio input that capture the important aspects of the signal that are associated with the various human speech sounds. These features are extracted in such a way that they are meaningful and also robust to certain conditions. Generally, the feature extraction is done by converting the speech signals into a parametric form for processing and analysis.There are many approaches to acoustic feature extraction and enhancement like Mel-Frequency Cepstral Coefficients (MFCCs) [5], Linear Predictive Coding (LPC), Discrete Wavelet Transforms (DWTs), Feature space Maximum Likelihood Linear Regression (fMLLR) [6]. Here we describe the approach used in this research, which is the default approach used in Kaldi [7] one of the most widely-used ASR toolkit.

Figure 2.2: This figure shows the pipeline for Feature Extraction

Before the features of the speech signal are extracted, a series of pre-processing

work is executed. This is known as Pre-Emphasis. In this step, the low-frequency and the high-frequency components of a voiced sound signal are balanced out by either increasing the amplitude of the high-frequency component or decreasing the low-frequency component.After this step, the audio signals are cut into short frames and a windowing function is applied to each frame of the audio signal.In the final step, the features are extracted and are used to train the acoustic model.

2.1.1 Mel-Frequency Cepstral Coefficients Features

The Mel-frequency cepstral coefficients (MFCC) features are one of the popular features used for speech recognition. The reason for its popularity is due to its ability to imitate the behaviors of the human ear [5]. The complete process for extracting the MFCC features are as follows [8]:

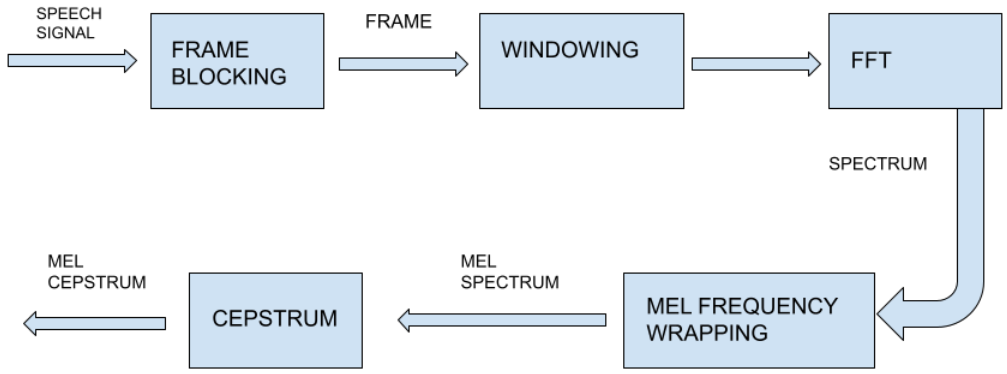

Figure 2.3: This figure shows the complete pipeline of the MFCC Feature Extraction

1. **Short Frames** : The audio signals are cut into short frames of 25-30 ms [5].

2. **Windowing** : Each short frame is multiplied by a windowing function (Hamming) [5].

3. **Fast Fourier Transform (FFT)** : The FFT is used on each frame of N samples to calculate the magnitude of frequency domain [5].

4. **Mel frequency Wrapping** : The magnitude of the frequency domain is used to calculate the the log filter-bank energies of each bypass filter [5].

5. **Discrete Cosine Transform (DCT)** : The DCT is applied on the log filter-bank energies to obtain the Mel-frequency cepstrum and the coefficients are used as features [5].

2.1.2 Feature Space Maximum Likelihood Linear Regression

The Feature Space Maximum Likelihood Linear Regression (fMLLR) [9] features is a type of feature transform techniques where the features are converted into a speaker adaptive form using the transformation matrix. Only the maximum likelihood transformations are considered as features for fMLLR.This type of feature is mainly used in HMM-Based speech recognition [6][10].

Compared with other features, the fMLLR feature transforms tend to perform better than other features techniques like MFCCs, FBANKS, etc., due to the process of speaker adaptiveness [11].

2.1.3 Cepstral Mean and Variance Normalization

Cepstral Mean and Variance Normalization (CMVN) is a normalization technique used for robust speech recognition where all the utterances are normalized into zero mean and unit variance [12]. The CMVN stats are collected once the MFCC features are extracted.

2.1.4 Linear Discriminative Analysis

In ASR, the Linear Discriminative Analysis (LDA) is used to create HMM states using the features in a low-dimensional space. LDA can be used not only for dimensionality reduction but also as a classification as well [13]. The limitation of LDA is that the diagonal gaussians are not represented perfectly in the low-dimensional space and that's the reason LDA features are followed by MLLT transforms [14].

2.1.5 Maximum Likelihood Linear Transform

Maximum Likelihood Linear Transform (MLLT) is a feature transformation technique that transforms the data into a gaussian variated space [14]. The diagonal co-variance property is improved in the features provided they are independent of each other [15]. MLLT takes the low-dimensional space derived from the LDA and creates transforms with well-represented diagonal gaussian.

2.2 Acoustic models

The acoustic model uses large quantities of transcribed audio data to learn the associations between acoustic features (e.g., MFCCs) and specific human speech sounds ("t", "a", "v"). Most of the acoustic models are built using the probability distribution over space [16]. The usage of acoustic models is to find the correlation between the extracted features and the phoneme labels. There are many different approaches to acoustic modeling for ASR. Here we describe only the approaches used in this research: Hidden Markov Models, Gaussian Mixture Models, Subspace Gaussian Mixture Models, and Deep Neural Networks.

2.2.1 Hidden Markov Model

A Markov model is a stochastic method to model randomly changing systems. It provides a way to model the dependencies of the current state as a method of predicting the next state. A Hidden Markov Model contains two parts – hidden and observed, where the states of the model are hidden which are not directly observed [17]. Their presence is observed by the output that each hidden state emit. Owing to the temporal structure of the speech, the HMM's can be able to model through self-loops and predict the speech using dynamic programming algorithms such as Viterbi algorithm which provides an efficient way to make an inference or prediction of the next character or phone. The forward-backward algorithm is used for the probabilistic interferences of each state transition allowing for the HMM's to update the internal weights like the propagation in a DNN[18].

2.2.2 Gaussian Mixture Model

The Gaussian Mixture Model (GMM) is commonly used to estimate of the probability density function used in statistical classification systems [19]. This combined with HMM to estimate the density and maximize the likelihood of the data's distribution. Subspaces are preferred over larger models which are introduced to reduce the parameter estimation issues which thereby reduces the dimensionality of the system. Though GMM's are capable of producing high- quality results, their main disadvantage is that they are inefficient at modeling non-linear data. Thus, speech, with its inflection, tone, and other properties is not the ideal application [18].

2.3 Deep Neural Networks

Deep Neural Networks (DNN) is an extension of Artificial Neural Networks with more hidden layers in between to learn larger sets of data and thereby providing

better predictions. DNN's are feed-forward networks that can be trained both via forward and backward propagation [20]. Due to the large amount of hidden layers and with each hidden layer having many units makes them very complex and this potential is very much required for acoustic modelling [21].The development of end-to-end speech recognition and generating models has been aided by deep neural networks. Various networks like Convolutional Neural Networks (CNN) [22], Recurrent Neural networks (RNN) [23], and transformer [24] models have accomplished excellent results. In CNN's, two-dimensional networks are used where the features are organized in two-dimensions, where the dimensions represent time and frequency respectively [25] whereas in Recurrent Neural Networks, the computations are performed in such a way that the output predicted depends on the previous states [23]. In the transformers, the architecture is based on encoder-decoder [24].

2.4 Language Model

Language models are one of the integral parts of speech recognition. The language model is used to compute the likelihood probability for a sequence of n-words. Most of the typical language models are n-gram based where n=1-5, meaning they compute the probability of the next word given the past n-1 words [26]. The probability is computed using the Maximum Likelihood Estimate.The phonemes and the words are modelled into a n-gram distribution and thereby each n-gram would contain n-words.

2.5 Decoder

The decoding process takes place once the Acoustic model has been trained using the features extracted and is used to recognize the sequence of words from the speech data and metrics are evaluated [27]. In Kaldi, after the acoustic model is trained, a decoding graph is created which contains all the lexicons, and dependencies, and this

graph is used for decoding the sequence of words. There are two types of decoders namely beam search decoders and greedy decoders.

2.6 Kaldi Toolkit

Kaldi [7] is a open-source toolkit used for recreating several speech recognition systems.The training and decoding are based on finite-state transducers (FST) and it supports linear and affine transformations. This toolkit can be used for feature extraction, acoustic modeling, decoding, language modeling. For feature extraction, the features like Mel-Frequency Cepstral Coefficients (MFCC) [5], Perceptual linear prediction (PLP), Linear Discriminant Analysis (LDA) can be extracted using Kaldi. The acoustic models HMM-GMM, DNN, TDNN, Subspace GMM can be easily recreated using this toolkit. N-gram language model is used in Kaldi as it is based on FST frameworks and it has several decoders which can be used for decoding [7].

2.7 Evaluation Metrics

2.7.1 Word Error Rate

The Word Error Rate (WER) will be used as an evaluation metric to estimate the performance of most of the ASR models. The WER is defined by the number of errors between the predicted and the reference sentence. Lower WER implies that the ASR model recognizes speech accurately (high accuracy) whereas Higher WER implies that the ASR model has low accuracy. To calculate the WER, the errors Substitutions (S), Insertions (I), Deletions (D) that occur in the recognition of a sentence are summed up and this value is divided by the number of words spoken in the reference sentence (N).

The equation to calculate the WER is as follows:

$$WER = \frac{(S + I + D)}{N} \tag{2.1}$$

For example, lets say a person speaks a sentence from the transcript file, **"I bought a table"** and the recognized output is: **"I brought the table"**. In this example, the recognized output has **2 substitutions**; **"bought"** changes to **"brought"** and **"a"** changes to **"the"**.

From equation 2.1, the WER in the example is :

$$WER = \frac{(2 + 0 + 0)}{4} = 0.50 \tag{2.2}$$

The Word Error Rate for the above example is 0.50

2.8 Types of Noise in Speech Recognition

Noise is a type of unwanted sound or signal which can corrupt the whole audio signal while transmitting or recording. Most of the ASR models find it difficult to handle noise and may require several methods and approaches to minimize its impact. There are many different types of noises that can impact ASR quality. Here we consider noise on two different dimensions: continuous vs. punctuated, and mechanical vs. produced by a living being.

2.8.1 Continuous vs Punctuated Noise

Continuous noise is a type of noise that is being produced continuously. This type of noise can be produced from heating systems, car engines, running motors or it could be produced from anywhere where the noise is continuous and does not have sudden changes in volume. Noises like running tap, human chatter in the background also come under this category. Punctuated noise is a type of noise that is being produced variably, it consists a combination of quiet and noisy periods. Examples include dog

barking, baby crying, door slamming.

2.8.2 Mechanical vs Non-Mechanical Noise

Mechanical noises are a type of noise that are mostly produced by machines. Examples include:heating and ventilation systems, car engine, vacuum cleaner, factory equipment, running motors, running tap. Non-mechanical noises are a type of noise that are produced by human beings. Examples include: crowd chatter, Ambience in an exhibition hall/airport.

Types of Noise	**Continuous Noise**	**Punctuated Noise**
Mechanical Noise	Car Engine, Vacuum Cleaner	Gun firing, Train Passing Nearby
Non-mechanical Noise	Crowd chatter, Ambience Noise	Dog barking, Baby crying

Table 2.1: Different types of Noises used

Table 2.1 provides a small sample-space as to what categories/kinds of noise does each noise belongs.

The mechanical and non-mechanical noises used in this research are as follows:

- **Mechanical Noises** : Running Tap (continuous noise), Dishes (continuous noise), Door Slamming (Punctuated noise), Truck Horn (Punctuated noise).

- **Non-Mechanical Noises** : Party Chatter (continuous noise), Restaurant Chatter (continuous noise), Dog barking (Punctuated noise), Cat-Meowing (Punctuated noise).

In this research, we have taken this small sample space of noises that are most frequently produced/created when using the ASR applications. Non-mechanical noises like party chatter , restaurant chatter are in the frequency range of the human speech

whereas the mechanical noises except truck horn aren't in the frequency range of the human speech are therefore used to find the impact of noise in ASR.

2.9 Prior Work

In ASR, various forms of Deep Neural Network models have been proposed using several language datasets but most of these models aren't trained using noise-mixed data to make it more robust to noise and other external conditions. The following papers explored the impact of noise and also on creating noise-robust ASR models. Ayesha et.al [27] provided a comparative study on various acoustic and deep learning models and thereby created robust models in a noisy environment. The robust models were trained by noise-augmented training data and testing these models on both clean and noise data. They experimented with several models only by noise augmentation and not by other techniques whereas, in this work, we will be finding out which type of augmentation technique is well suited to different kinds of noise. Hu et.al [28] proposed a noise-robust speech recognition system called interactive feature fusion network (IFF-Net) to learn the missing latent information from the enhanced feature and original noisy feature as a fused representation. This system achieved better results on the RATS Channel-A corpus.

Urmila et.al [29] elaborated on a few problems due to changes in environmental conditions and speaker characteristics and proposed a method to increase the robustness of the ASR systems using speech enhancement techniques like Spectral Normalization and Spectral Subtraction. Giurgiu et.al [30] explained how energy normalization and speech re-synthesis can improve the performance of ASR systems by recognizing speech signals in high-noisy conditions (negative SNR). Kinoshita et.al [31] investigates whether the usage of single-channel time-domain neural networks can help in the reduction of noise and thereby improve the performance. Gupta et.al [32] proposed a Back-propogation Artificial Neural Network with feature compression us-

ing MFCC and thereby producing improved performance with low values of SNR . Liu et.al [33], proposed a noise-resistant speech recognition called Wavoice that merges two unique voice sensing modalities millimeter-wave signals (mmWave) and audio signals. This system is modeled based on the inherent correlation between mmWave and audio signals and has performed really well in various conditions in a range of 7 meters.

2.10 Data Augmentation

Data Augmentation is the process of including additional data into the training set artificially. The purpose of using data augmentation is to increase the performance of the ASR models. There are several techniques by which data augmentation can be done. This includes spectrogram augmentation and raw augmentation.

2.10.1 Spectrogram Augmentation

Spectrogram Augmentation (SpecAugment) [4] is done using spectrograms in which certain sections of the spectral representations of the audio are blocked out. It is performed using the log mel spectrogram of the input speech data. Spec Augment is preferred over other augmentation techniques since it doesn't require any extra data as it is being applied on the log mel spectrograms and its computationally cheap as well. SpecAugment [4] has provided the state-of-the-art results on the LibriSpeech 960h [34] and Switchboard 300h [35] datasets using the Listen Attend Spell model [36]. There are two ways by which spectrogram augmentation can be done [4].

1. **Frequency Mask**: A range of frequencies is randomly erased out by adding horizontal bars in the spectrogram. [4].

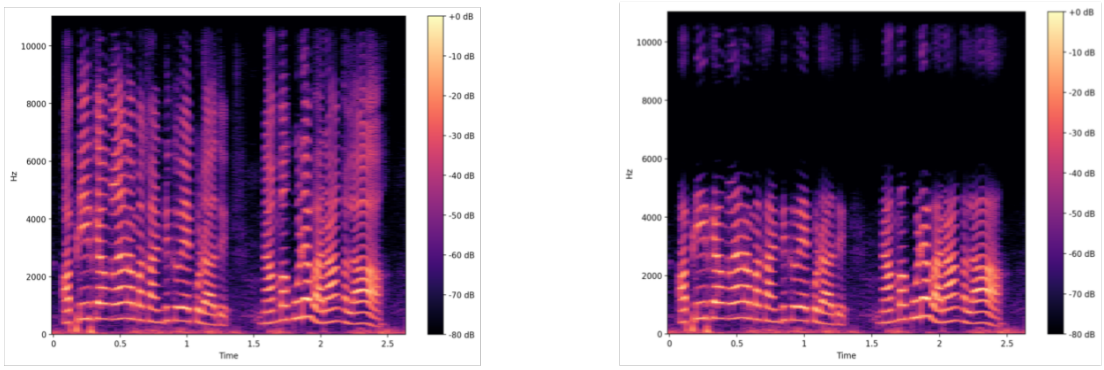

Figure 2.4: Spectrogram Augmentation of a audio signal with frequency masking

Figure 2.4 shows the color map comparison of the original audio file with the frequency masked audio file.

2. **Time Mask**: A range of time blocks is randomly erased out by adding vertical bars in the spectrogram [4].

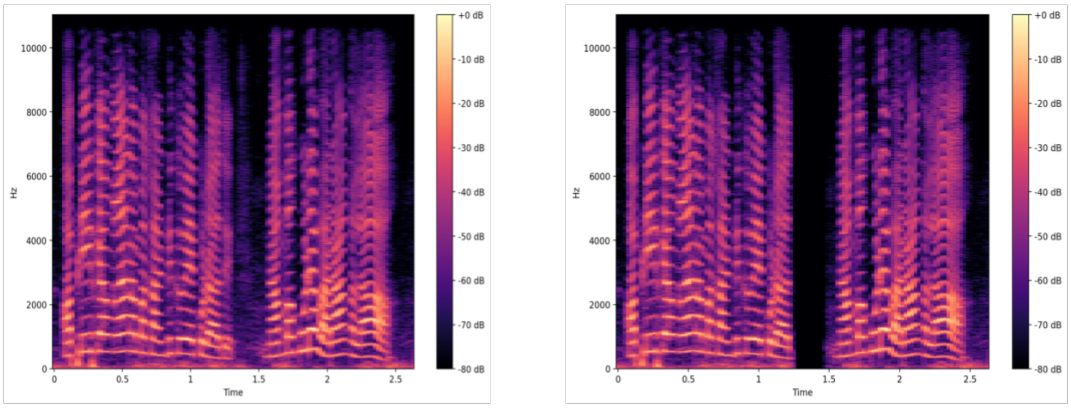

Figure 2.5: Spectrogram Augmentation of a audio signal with time masking

Figure 2.5 shows the color map comparison of original audio file with the time-masked audio file.

2.10.2 Raw Augmentation

Raw augmentation is done using the raw audio data. There are several ways by which raw audio augmentation can be done. These include Time Shift, Pitch Shift, Time Stretch, Adding Noise.

1. **Time Shift** : Shifting the audio signals to either the left or to the right for the given amount of time in seconds.

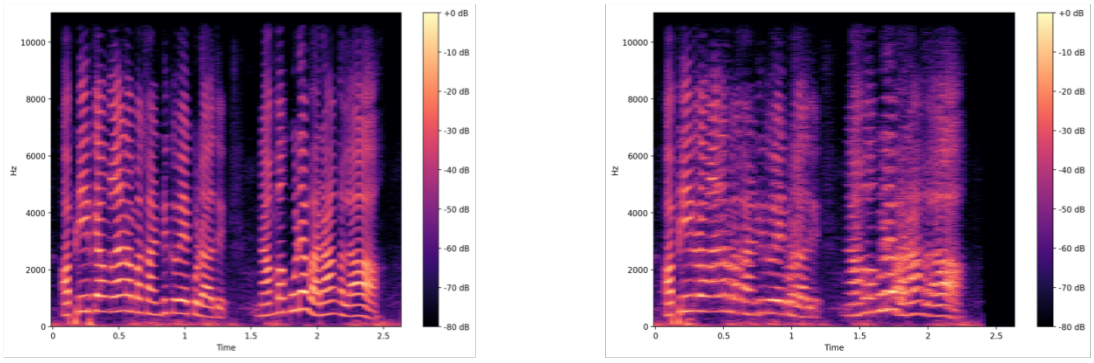

Figure 2.6: Raw Augmentation of a audio signal by shifting the time of the audio signal

Figure 2.6 shows the color map comparison of original audio file with the time-shifted audio file. The audio file is shifted towards the left side for a period of 0.5 seconds.

2. **Pitch Shift** : Changing the pitch of the audio signals by a random amount.

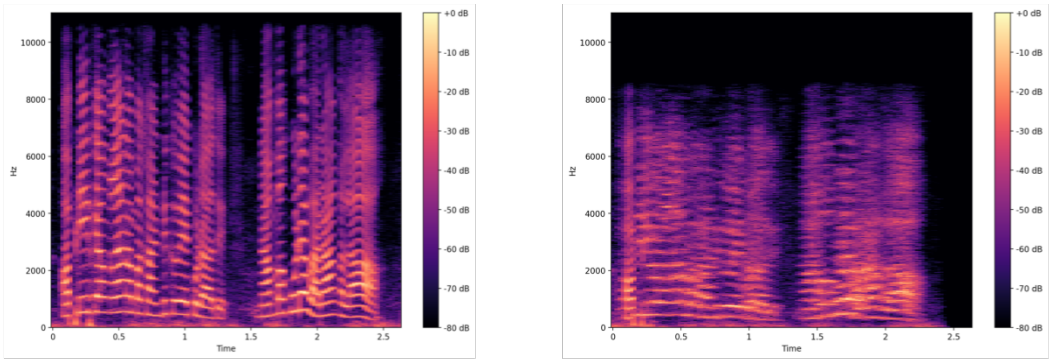

Figure 2.7: Raw Augmentation of a audio signal by changing the pitch of the audio signal

Figure 2.7 shows the color map comparison of original audio file with the pitch-shifted audio file.The audio file is shifts the pitch down without changing the tempo.

3. **Time Stretch** : Stretching the audio signals (speeding up or slowing down) for a given amount of time.

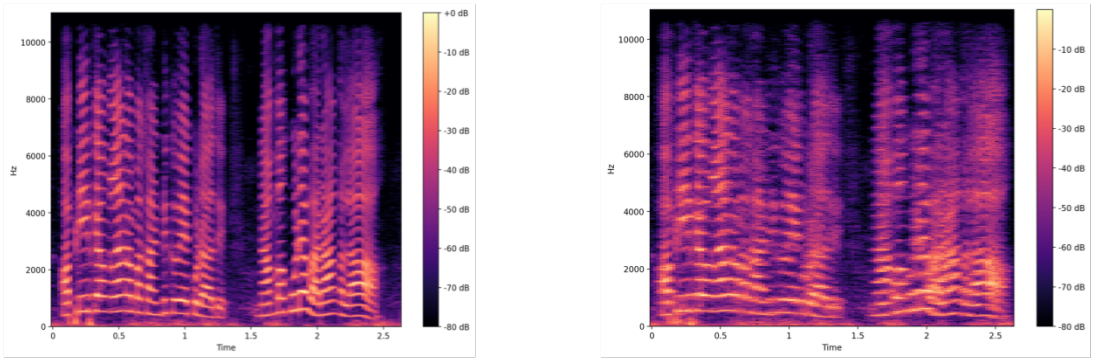

Figure 2.8: Raw Augmentation of a audio signal by stretching the pitch of the audio signal

Figure 2.8 shows the color map comparison of original audio file with the time-stretched audio file (slowing down).The audio file is stretched without changing the pitch.

4. **Adding Noise** : Adding some random noise into the audio signals.In this research, we will be specifically using this method of augmentation by experimenting with various kinds of noises.

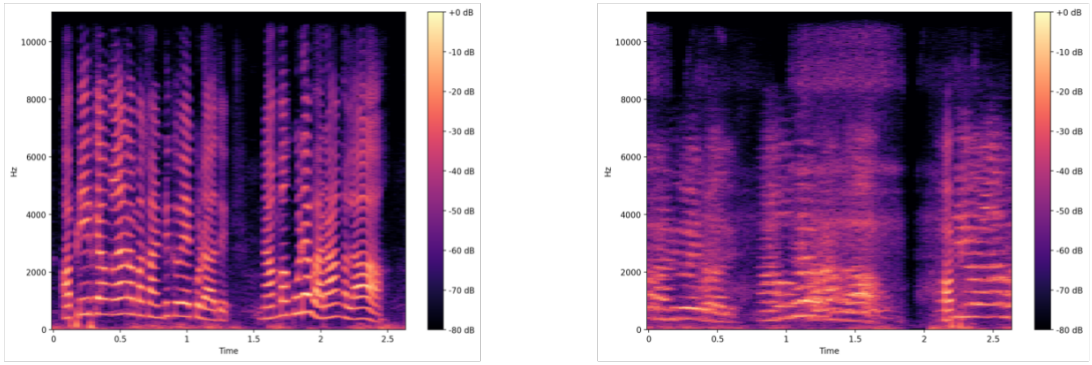

Figure 2.9: Raw augmentation of a audio signal by adding noises to the audio data

Figure 2.9 shows the original audio file with the noise-mixed audio file.The noise sample used in this figure is dog-barking.The noise-sample with a sample rate of 16kHz is mixed with the audio file continuously till the end of the audio file.

Chapter 3

Dataset

The data used for the approach is a low-resource Tamil language dataset and it has been provided by SpeechOcean.com and Microsoft for a low-resource ASR challenge in Interspeech 2018 [3]. The dataset consists of read speech and speech conversations that have been split into utterances and these have been transcribed as well. The dataset contains a total of 50 hours of recorded speech data in a clean noise-free environment. The dataset is split into the train and test splits where the train split consists of 40 hours of speech data, the test split consists of 5 hours of speech data.

Description	**Tamil**	
	Train set	**Test set**
Size (hours)	40	5
Unique utterances	39131	3081
Speakers	1780	118

Table 3.1: Tamil low-resource language dataset description

Table 3.1 provides the description of the Tamil low-resource language dataset like the unique utterances, number of speakers used. All the audio files are sampled with a sample rate of 16kHz and consists of 1 channel. A total of 1900 speakers used for the whole dataset. The length of each of the unique utterances are in the range of 3000ms - 10000ms.There is a total of 42212 unique utterances in the dataset.

The description of the audio data in the dataset is as follows:

- Channels : 1

- Sample Rate : 16000 Hz
- Precision : 16-bit
- Bit Rate : 256k
- Sample Encoding: 16-bit Signed Integer pulse code modulation (PCM)

Chapter 4

Methodology

The following sub-sections provide a description of the proposed method for this thesis and the objectives to be achieved.

4.1 Data Preparation

All the experiments are conducted on the low-resource Tamil language dataset where all the audio data files are sampled at 16kHz. The training dataset split contains 40 hrs of recorded speech data and the test data split contains 5 hrs of speech data.

4.1.1 Adding Noise for Raw Data Augmentation

Several kinds of noises files are mixed with the training speech data files to create the noisy dataset. All the noise files were sampled at 16kHz. The sound level of these noise samples were reduced by 20dB (Noise - 20dB) and are mixed with the speech data signals. For the continuous noises, the noises were mixed directly with each of the training data files whereas, for the punctuated noises, the noises were mixed at a fixed time interval. The time interval is selected from a random value ranging between 2000ms to 5000ms to each of the audio files in the training dataset. All the experiments are conducted where the sound level is reduced by 20dB (Noise-20dB) and is mixed with the speech data.

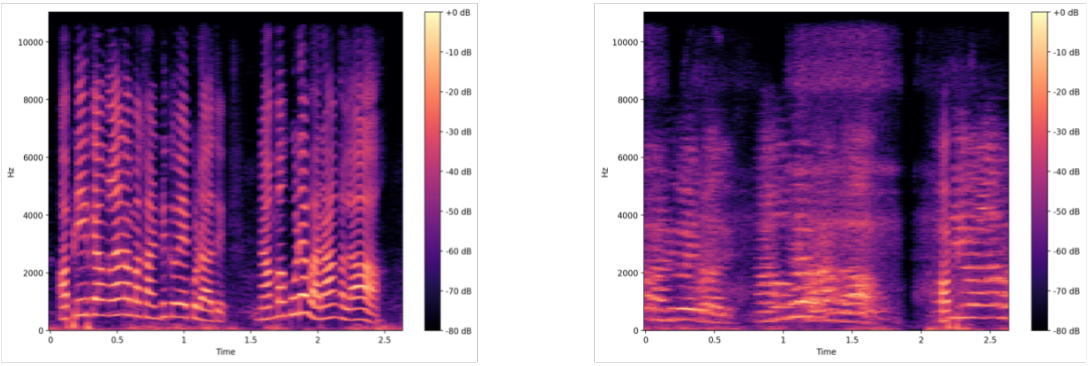

Figure 4.1: Adding noise to audio file for raw data augmentation

Figure 4.1 provides the colour wave comparison of a audio file from the dataset and the noise-mixed audio file. The noise sample used in the above figure is dog-barking. The noise level of the dog-barking noise was reduced by 20dB and were mixed with audio file.

4.1.2 Frequency Masking for Spectrogram Augmentation

For SpecAugment [4], the frequency of the audio data is masked for a block of the frequency channels. The frequency is masked for all the audio files in the training data. The clean data along with the spec-augmented data forms the new training data set and this was used to train the baseline models.

4.2 Baseline Models

The baseline models are built using the Tamil language dataset via the Kaldi toolkit [7]. The Kaldi recipes are used to build these models with some variations in the data points and parameters. All the acoustic baseline models analyze the data using the features that are extracted and their probabilities are calculated and these values are combined. The ASR models used for baseline are GMM-HMM, SGMM, SAT Triphones, and DNN.

4.2.1 GMM-HMM Acoustic Model

For training the GMM-HMM Model, a series of feature extraction and acoustic models training and alignments are done sequentially. After each stage of training the acoustic model, the model can be used for decoding and thereby calculating the performance. The transcription files of the speech dataset, and the language dictionary containing the lexicons, silent and non-silent phonemes are required for the training process. The whole training procedure for the GMM-HMM Model [27] is provided below:

1. The first step is to extract the MFCC acoustic features from the training dataset and the test dataset is extracted. After extracting the features, the cepstral mean and variance normalization (CMVN) statistics are calculated for both the training and test datasets MFCC acoustic features.

2. The monophone acoustic model (Mono) is trained using the MFCC features extracted. This monophone model does not contain any information about the next phonemes. After training, the monophone model is aligned in such a way that the transcripts of the speech files are aligned. The alignments are done for the model so that the preceding/other algorithms can use this model to improve the performance of the recognition.

3. The delta-based triphone model (tri1) is trained using the alignments and the features of the monophone model. Extra parameters like the HMM States and the Number of Gaussians are required for training the model. We used 4200 HMM States and 40000 Gaussians for training the tri1 model. These parameters depend on the training data and phonemes. After training, the alignments of the triphone model are done.

4. The delta + delta-delta triphone model (tri2a) is trained using the first and second-order derivatives of the audio signal. We used 4200 HMM States and

40000 Gaussians for training the tri2a model.

5. The LDA-MLLT Triphone model (tri2b) (GMM-HMM) is trained using the LDA - MLLT features extracted for the speech data (Triphone + Delta Delta + LDA and MLLT). We used 4200 HMM States and 40000 Gaussians for training the tri2b model. The alignment of the LDA-MLLT triphone model is done using the fMLLR features.

6. The Speaker Adaptive Training (SAT) [37] Triphone model (tri3b) is trained using the SAT features along with the delta-delta and LDA-MLLT features. We used 4200 HMM States and 40000 Gaussians for training the tri3b model. The SAT triphone model is used for speaker and noise normalization by adapting to each speaker. It is also used to calculate the variance in the phonemes.

The decoding can be done after each step to analyze the performance of each acoustic model. In Kaldi, the decoding is done by creating a decoding graph once the model is trained. The decoding graph (HCLG) is used for decoding and also calculating the metrics. The graph consists of lexicons (L), HMM definitions (H), an acceptor that encodes the language model (G), and the context (C) containing the phonemes.Figure 4.3 shows the GMM-HMM acoustic model and how the model is being trained sequentially.

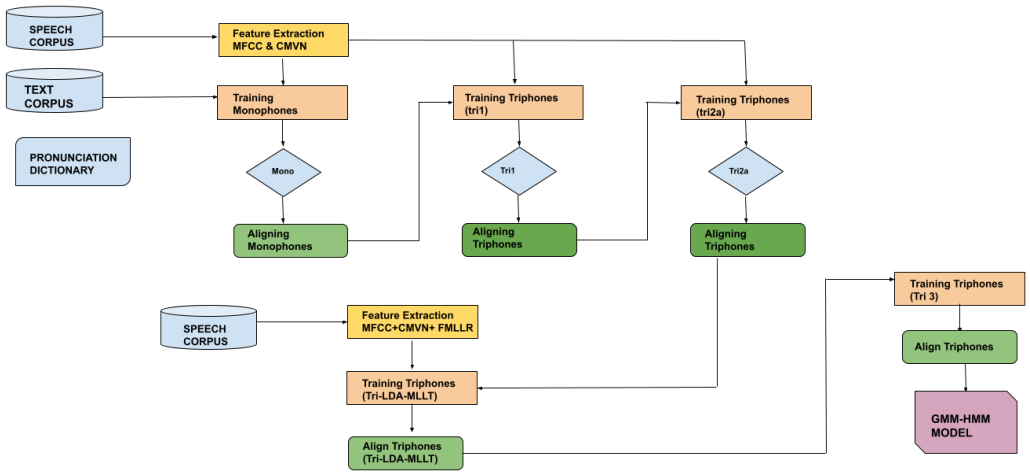

Figure 4.2: GMM-HMM Acoustic model

4.2.2 Subspace GMM Acoustic Model

Subspace GMM is a type of acoustic model where all the phoneme states use a common Gaussian Mixture Model structure [38]. The SGMM model is trained by using clustering the Gaussians from the GMM-HMM model that has been trained with the HMM states and the Gaussians. The first step of training the SGMM is by clustering Gaussians using the Universal Background Model (UBM). The UBM model is a speaker-independent high order GMM model [39]. The next step is the training of SGMM models using the UBM model having the state probability distribution functions as identical. The final step of the training process is to use the EM algorithm to train the SGMM model using the alignments from the GMM-HMM and also from the SGMM model as well [38].

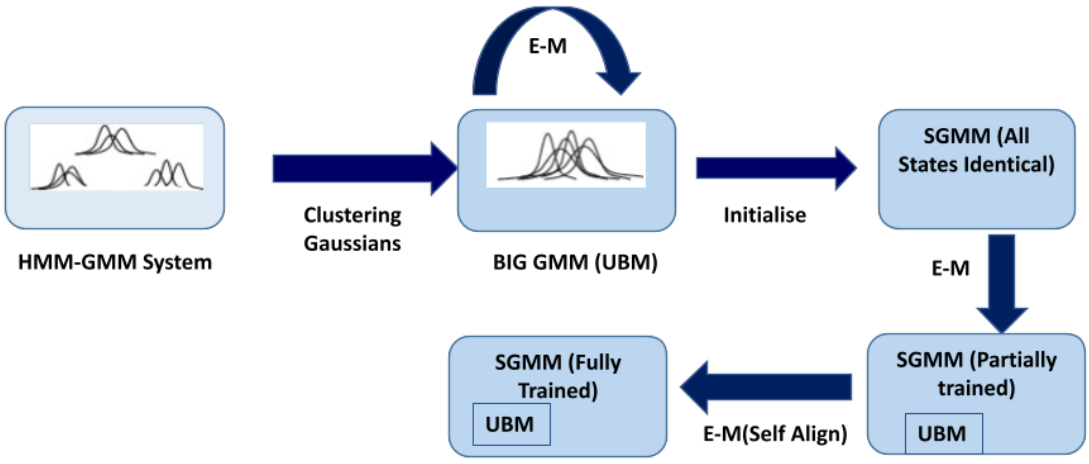

Figure 4.3: SGMM Acoustic model

Figure 4.4 shows the pipeline in which the SGMM acoustic model is being trained sequentially.

4.2.3 Deep Neural Networks (DNN)

Deep Neural Networks (DNN) have been producing state-of-the-art results in speech recognition. Karel's implementation of DNN is re-created in this research [40].Here, the sequence discriminative training of the deep neural networks is performed. The DNN model consists of 6 hidden layers with the output layer where each hidden layer has 2048 nodes [41]. The DNN model can be trained using the features extracted in the GMM-HMM acoustic model. The input to the DNN model is an 11 frame window of the 40-dimensional feature map. The DNN training can be done in several stages sequentially namely: Restricted Boltzmann Machines (RBM) pre-training, Frame cross-entropy training, sequential discriminative training. The initial stage is the extraction of the 40-dimensional feature map for training the DNN. This 40-dimensional feature map includes MFCC with CMVN stats, LDA-MLLT, and fMLLR.The first stage is pretraining of stacked Restricted Boltzmann Machines (RBM). The first RBM in the stack uses the Gaussian-Bernoulli units and the following ones

Bernoulli-Bernoulli units. The next stage is frame cross-entropy training. In this stage, the DNN is trained to classify the frames into the probability distribution functions (PDF's). This process is completed by using mini-batch stochastic gradient descent [41]. The final stage of the training is the state-level Minimum Bayes Risk (sMBR) sequence discriminative training. In this stage, the neural network is trained using stochastic gradient descent to optimize the sentences and maximize the accuracy of the labels derived from the reference alignments. Initially, the lattices and alignments are generated before the neural network training starts. The decoding for the DNN can be performed both after RBM pretraining stage as well during the sMBR DNN training. The decoding is done using the HCLG decoding graph that is used for the GMM-HMM model and the SGMM model.

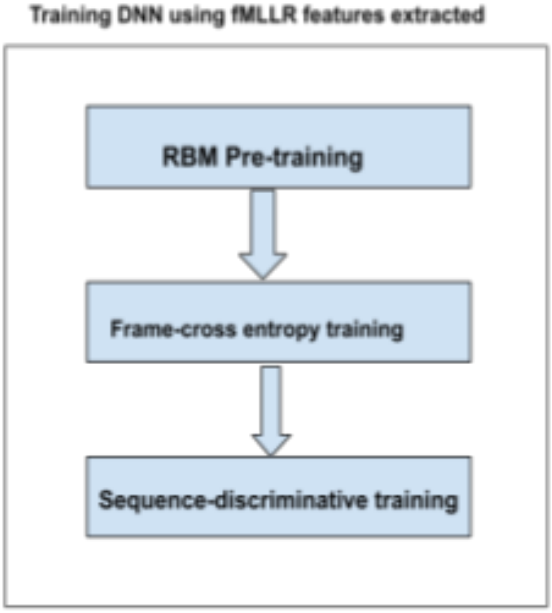

Figure 4.4: Pipeline of the DNN model

Chapter 5

Results and Discussions

The above-mentioned baseline models were trained using the 40 hrs of the training set and then tested using the test set of 5 hrs and the Word Error Rate is calculated for the Tamil Language. Several experiments were conducted without noise (clean data) and also with different kinds of noise to find the impact of noise and also how it affects the performance of the Baseline models.

The baseline results without the presence of noise for the Tamil language were performed using the Kaldi toolkit [7]. Table 5.1 provides the results of the top performing acoustic models, GMM-HMM, SGMM, SAT Triphones, DNN (Karel's implementation) [40] trained and tested using the clean data and the best WER from each model is tabulated.

Baseline	**WER**
GMM-HMM	44.66
SGMM	39.22
SAT-Triphones	36.15
DNN	32.58

Table 5.1: ASR Baseline Results for Tamil Language

From the table 5.1, we can infer that the DNN architecture has produced the best WER out of all the other acoustic models. The GMM-HMM acoustic model has the highest WER compared to the other acoustic models.

The baseline results will be used as a reference to compare the impact of noise in

the baseline models and data augmentation techniques (Raw Augmentation, SpecAugment [4]).

5.1 Impact of Noise

To find the impact of noise in ASR models, a noise sample from each kind of noise (mechanical/non-mechanical, continuous/punctuated) was taken and this noise sample was mixed with all the audio files in the test set thereby creating a noise-mixed test set. This modified test set is then tested with the baseline models like GMM-HMM, SGMM, and DNN that are trained on clean/unaugmented data and the results were tabulated. The noise sample used for this research are as follows:

- **Mechanical Noises** : Running Tap (continuous noise), Dishes (continuous noise), Door Slamming (Punctuated noise), Truck Horn (Punctuated noise).

- **Non-Mechanical Noises** : Party Chatter (continuous noise), Restaurant Chatter (continuous noise), Dog barking (Punctuated noise), Cat-Meowing (Punctuated noise).

For the continuous noises, each of the noises were directly mixed with all the test audio files. Each of the punctuated noises are added at a fixed time interval. The time interval is selected from a random value ranging between 2000ms to 5000ms.

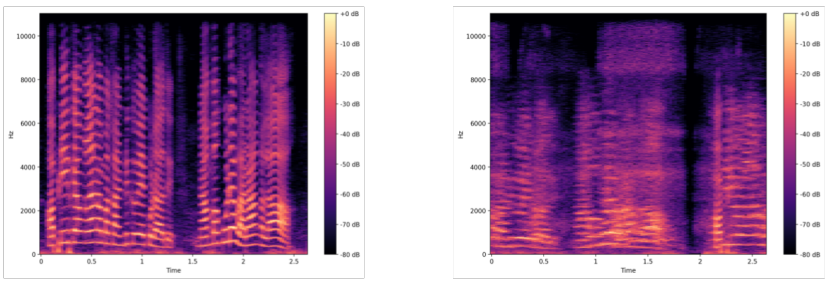

Figure 5.1: Original audio vs Noise-mixed audio

Figure 5.1 shows the colour maps of the original audio file and the noise-mixed audio data. The noise sample used is the restaurant chatter noise that is mixed continuously with the original audio file.

The results with the presence of several mechanical and non-mechanical noises for the Tamil language were performed using the Kaldi toolkit [7]. Table 5.2 provides the results of baseline models like GMM-HMM, SGMM, and DNN that are trained on clean/unaugmented data and tested on the mechanical noise-mixed test sets and the best WER from each of the model is tabulated.

Acoustic Model	**Clean Data**	**Continuous Noise**		**Punctuated Noise**	
		Run Tap	**Dishes**	**Truck Horn**	**Door Slam**
GMM-HMM	44.66	61.2	55.1	50.52	50.21
SAT Triphones	39.22	69.12	61.46	48.77	47.21
SGMM	36.15	54.4	48.18	45.24	44.94
RBM Pretrained	32.88	47.89	42.98	40.75	40.55
DNN	32.58	47.18	41.54	39.44	39.12

Table 5.2: ASR results with the presence of different mechanical noises

The baseline results from Table 5.1 are used as a reference to find out the impact of noise on the baseline models. From the table 5.2, we can infer that all the mechanical noises regardless of their type impact WER and improving the architecture is not enough to overcome the impact of adding noise. The DNN architecture has produced the best WER while the GMM-HMM acoustic model has the highest WER as compared to the other acoustic models for all the test sets. We can see that the continuous noises (Running Tap, Dishes) have a bigger impact on the performance of the baseline models when compared to the punctuated noises (Truck Horn, Door Slamming). Out of all the mechanical noises, running tap has the biggest impact on the ASR performance whereas door slamming noise has the least impact on the ASR performance for all the baseline models. The SAT-triphone model with the running tap noise has produced the highest WER of 69.12 and thereby has the biggest impact

whereas the DNN model with the door slamming noise has produced the lowest WER of 39.12.

Table 5.3 provides the results of baseline models like GMM-HMM, SGMM, and DNN that are trained on clean/unaugmented data and tested on the mechanical noise-mixed test sets and the best WER from each of the model is tabulated.

Acoustic Model	**Clean Data**	**Continuous Noise**		**Punctuated Noise**	
		Party	**Restaurant**	**Dog-bark**	**Cat-meow**
GMM-HMM	44.66	66.23	53.20	52.20	48.00
SAT Triphones	39.22	76.24	60.57	47.27	52.38
SGMM	36.15	66.05	47.75	45.19	41.26
RBM Pretrained	32.88	59.03	42.57	41.54	36.67
DNN	32.58	56.88	41.18	39.91	35.91

Table 5.3: ASR results with the presence of different non-mechanical noises

The baseline results from Table 5.1 are used as a reference to find out the impact of noise on the baseline models. From the table 5.3, we can infer that all the non-mechanical noises regardless of the baseline model impact the WER and improving the architecture of these models wont help to overcome the impact of adding noise. The DNN baseline model has given the best WER (lowest) whereas the GMM-HMM baseline model has produced the highest WER when compared to other acoustic models for all the test sets. The continuous noises (Party and Restaurant chatter) have a huge impact on the ASR performance of all the baseline models when compared to the punctuated noises (Dog-bark, cat-meow). Out of all the non-mechanical noises, party chatter has the highest impact on the WER whereas cat-meow noise has the least impact on the WER for all the baseline models except the SAT-triphone model. The SAT-triphone model with the party chatter noise has produced the highest WER of 76.24 and thereby has a huge impact on the model whereas the DNN model with the cat-meowing noise has produced the lowest WER of 35.91.

From table 5.2 and table 5.3, we can infer that the cat-meowing noise has a least impact on the ASR performance and the party chatter noise has the highest impact on the ASR performance out of all the noise test sets. Both mechanical and non-mechanical noises have a considerable impact on the performance of the baseline models.

The impact of noise on the baseline models can be reduced by performing data augmentation techniques like raw data augmentation and spectrogram augmentation.

5.2 Data Augmentation

Data Augmentation is the process of including additional data into the training set artificially. The purpose of using data augmentation is to increase the performance of the ASR models. A noise sample from each kind of noise was taken and this noise sample was mixed with all the audio files in the training set thereby creating a noise-mixed augmented training set. This training set was used to train the baseline models and the same noise-mixed test set that was used to find the impact of noise was used to test the baseline models that were trained on augmented train set. The raw data augmentation was performed specifically for each type of noises. In total, there were 6 augmentation training sets that were created and trained on baseline models.The 6 augmentation models are:

1. **Run Tap-Dishes Augmentation**: Each of the audio files from the training set were mixed with the running tap noise and also mixed with the dishes noise thereby creating the augmented data. The clean data along with the augmented run-tap data and the augmented dishes data formed the new augmented training set and this was used to train the baseline models.

2. **Dog bark-Cat meow Augmentation**: Each of the audio files from the training set were mixed with the dog-barking noise and also mixed with the cat-

meowing noise separately thereby creating the augmented data. The clean data along with the augmented dog-barking data and the augmented cat-meowing data formed the new augmented training set and this was used to train the baseline models.

3. **Truck horn - Door Slamming Augmentation**: Each of the audio files from the training set were mixed with the truck door noise and also mixed with the door slamming noise thereby creating the augmented data. The clean data along with the augmented truck door data and the augmented door slamming data formed the new augmented training set and this was used to train the baseline models.

4. **Party-Restaurant chatter Augmentation**: Each of the audio files from the training set were mixed with the party chatter noise and also mixed with the restaurant chatter noise thereby creating the augmented data. The clean data along with the augmented party chatter data and the augmented restaurant chatter data formed the new augmented training set and this was used to train the baseline models.

5. **Mechanical noise Augmentation**: Each of the audio files from the training set were mixed with the all the mechanical noises like run-tap, dishes, truck horn, door slamming separately thereby creating the augmented data. The clean data along with each of the augmented mechanical noise data formed the new augmented training set and this training set was used to train the baseline models.

6. **Non-mechanical Augmentation**: Each of the audio files from the training set were mixed with the all the non-mechanical noises like party, restaurant chatter, dog-barking, cat-meowing separately thereby creating the augmented data. The clean data along with each of the augmented non-mechanical noise

data formed the new augmented training set and this training set was used to train the baseline models.

For SpecAugment [4], the frequency of the audio data is masked for a block of the frequency channels. The frequency is masked for all the audio files in the training data thereby creating the augmented train set. The clean data along with the spec augmented data formed the new training data set and this was used to train the baseline models.

5.2.1 Raw Data Augmentation

The results with various raw data augmentations for the Tamil language were performed using the Kaldi toolkit [7]. Tables 5.4 and 5.5 provides the results of baseline models like SGMM, and DNN that are trained on different types of augmentations and tested on the mechanical noise-mixed test sets and the best WER from each of the model is tabulated.

Augmentation Type	Acoustic Model	Clean Data	**Continuous Noise**	
			Running Tap	**Dishes Noise**
No Data Augmentation	DNN	32.58	47.18	41.54
	SGMM	36.15	54.4	48.18
Running Tap & Dishes	DNN	31.88	38.03	36.21
	SGMM	36.16	48.81	44.23
Truck Horn & Door Slam	DNN	31.47	35.88	39.17
	SGMM	35.19	52.88	46.67
Party & Restaurant	DNN	31.85	38.41	36.91
	SGMM	35.84	50.71	44.48
Dog & Cat	DNN	31.72	38.11	39.83
	SGMM	35.49	53.18	46.8
SpecAugment	DNN	35.75	50.63	43.97
	SGMM	43.06	64.62	54.53

Table 5.4: ASR results for different augmentation types tested on mechanical - continuous noises

The baseline results from Table 5.2 (SGMM and DNN) are used as a reference to find out the performance of different augmentation types on the baseline models.

From the table 5.4, we can infer that all the raw data augmentation types improves the ASR performance regardless of the architectures. The spectrogram augmentation [4] doesn't improve the ASR performance when compared to the raw-data augmentation types. The DNN baseline model has given the best WER (lowest) for all the test sets. Both the continuous noises (running tap and dishes noises) have improved the ASR performance (WER) on all the raw data augmentation types when compared with no augmentation and both the continuous noises perform better with the running-tap and dishes augmentation for both the baseline models than with other raw data augmentation types. The running tap noise has the improved the ASR performance by a big margin when compared to the dishes noise.SpecAugment [4] doesn't improve the ASR performance (WER) for both the continuous noises when compared with the raw-data augmentation types.

Augmentation Type	Acoustic Model	Clean Data	**Punctuated Noise**	
			Truck horn	**Door Slam**
No Data Augmentation	DNN	32.58	39.44	39.12
	SGMM	36.15	45.24	44.94
Running Tap & Dishes	DNN	31.88	38.16	38.27
	SGMM	36.16	44.38	42.87
Truck Horn & Door Slam	DNN	31.47	36.06	35.88
	SGMM	35.19	40.97	40.2
Party & Restaurant	DNN	31.85	35.26	38.41
	SGMM	35.84	41.18	44.03
Dog & Cat	DNN	31.72	38.24	38.11
	SGMM	35.49	43.89	43.25
Spec Augment	DNN	35.75	44.91	42.74
	SGMM	43.06	52.32	51.31

Table 5.5: ASR results for different augmentation types tested on mechanical - punctuated noises

From the table 5.5, we can infer that all the raw data augmentation types improves the ASR performance (WER) regardless of the architectures. Karel's implementation of DNN [40] outperforms the SGMM [38] due to the initial pre-training of RBM's (involving the labeled frames) and also due to the early stopping of the training

thereby providing better results. The spectrogram augmentation [4] doesn't improve the ASR performance (WER) of both the baseline models when compared to the raw-data augmentation types. The DNN baseline model produces the best WER (lowest) for all the test sets. Both the punctuated noises (truck horn and door slam) provides a small improvement in the ASR performance (WER) on all the raw data augmentation types when compared with no augmentation and both the punctuation noises perform better with the truck horn-door slamming augmentation for both the baseline models when compared with other raw data augmentation types. Both the punctuated noises improve the ASR performance by a small margin. SpecAugment [4] doesn't improve the ASR performance (WER) for both the punctuated noises when compared with the raw-data augmentation types.

Tables 5.6 and 5.7 provides the results of baseline models like SGMM, and DNN that are trained on different types of augmentations and tested on the non-mechanical noise-mixed test sets and the best WER from each of the model is tabulated.

Augmentation Type	Acoustic Model	Clean Data	**Continuous Noise**	
			Party chatter	**Restaurant**
No Data Augmentation	DNN	32.58	56.88	41.18
	SGMM	36.15	66.05	47.75
Running Tap & Dishes	DNN	31.88	52.94	37.81
	SGMM	36.16	63.42	45.87
Truck Horn & Door Slam	DNN	31.47	55.79	39.91
	SGMM	35.19	63.47	46.15
Party & Restaurant	DNN	31.85	46.16	35.18
	SGMM	35.84	57.27	41.7
Dog & Cat	DNN	31.72	54.85	38.97
	SGMM	35.49	64	45.73
Spec Augment	DNN	35.75	58.86	45.02
	SGMM	43.06	68.56	54.37

Table 5.6: ASR results for different augmentation types tested on non-mechanical - continuous noises

The baseline results from Table 5.2 (SGMM and DNN) are used as a reference

to find out the performance of different augmentation types on the baseline models. From the table 5.6, we can infer that all the raw data augmentation types improves the ASR performance regardless of the architectures. Karel's implementation of DNN [40] outperforms the SGMM [38] due to the initial pre-training of RBMs (involving the labeled frames) and also due to the early stopping of the training thereby providing better results. The spectrogram augmentation [4] doesn't improve the ASR performance when compared to the raw-data augmentation types. The DNN baseline model has given the best WER (lowest) for all the test sets. Both the continuous noises (party chatter and restaurant chatter) provides a small improvement in the ASR performance (WER) on all the raw data augmentation types when compared with no augmentation and both the continuous noises perform better with the Party and restaurant augmentation for both the baseline models than with other raw data augmentation types.The party chatter noise has the improved the ASR performance relatively by a big margin when compared to the restaurant chatter noise.SpecAugment [4] doesn't improve the ASR performance (WER) for both the continuous noises when compared with the raw-data augmentation types.

Augmentation Type	Acoustic Model	Clean Data	**Punctuated Noise**	
			Dog bark	**Cat meow**
No Data Augmentation	DNN	32.58	39.91	35.91
	SGMM	36.15	45.19	41.26
Running Tap & Dishes	DNN	31.88	38.84	35.09
	SGMM	36.16	44.34	41.02
Truck Horn & Door Slam	DNN	31.47	38.81	33.23
	SGMM	35.19	44.13	40.9
Party & Restaurant	DNN	31.85	37.35	33.87
	SGMM	35.84	43.1	39.35
Dog & Cat	DNN	31.72	34.26	32.07
	SGMM	35.49	38.25	36.59
Spec Augment	DNN	35.75	46.27	39.53
	SGMM	43.06	51.68	47.11

Table 5.7: ASR results for different augmentation types tested on non-mechanical - punctuated noises

From the table 5.7, we can infer that all the raw data augmentation types improves the ASR performance (WER) regardless of the architectures. The spectrogram augmentation [4] doesn't improve the ASR performance (WER) of both the baseline models when compared to the raw-data augmentation types. The DNN baseline model produces the best WER (lowest) for all the test sets. Both the punctuated noises (dog-barking and cat-meowing) provides a slight improvement in the ASR performance (WER) on all the raw data augmentation types when compared with no augmentation. Both the punctuation noises perform better with the dog-barking-cat-meowing augmentation for both the baseline models when compared with other raw data augmentation types and spec augmentation [4]. Both the punctuated noises improve the ASR performance by a small margin. SpecAugment [4] doesn't improve the ASR performance (WER) for both the punctuated noises when compared with the raw-data augmentation types.

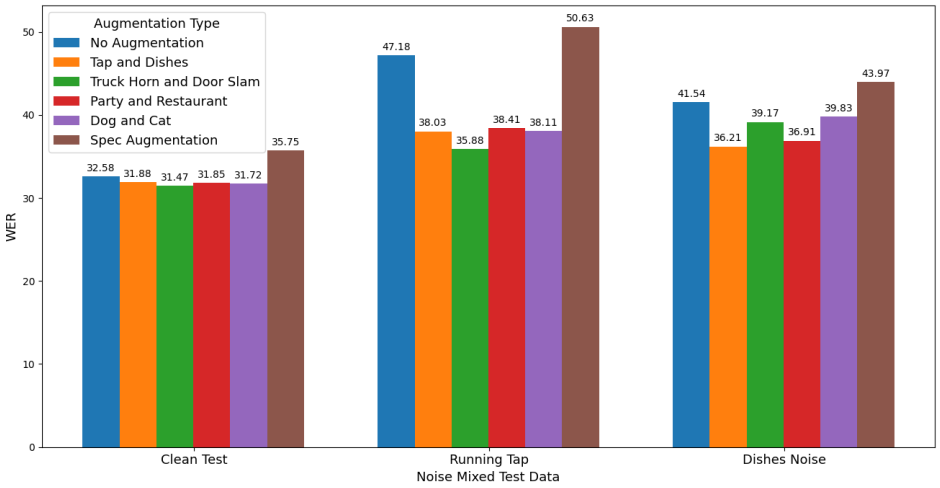

Figure 5.2: WER Comparison of raw data augmentations for the DNN Baseline model

Figure 5.2 visualizes the WER comparison of the raw data augmentations for the DNN Baseline model on the running tap and dishes noises. In this graph, we can infer

that the run-tap noise provides better performances in all of the raw augmentation types when compared with the dishes noises.Both the noises perform better with the run- tap and dishes augmentation when compared with other data augmentation types and also these noises provides a better improvement in the ASR performance (WER) on all the raw data augmentation types when compared with no augmentation. This graph shows that targeted raw data augmentation (Running tap-dishes augmentation) improves ASR performance. The spectrogram augmentation [4] doesn't improve the ASR performance when compared to the raw-data augmentation types.

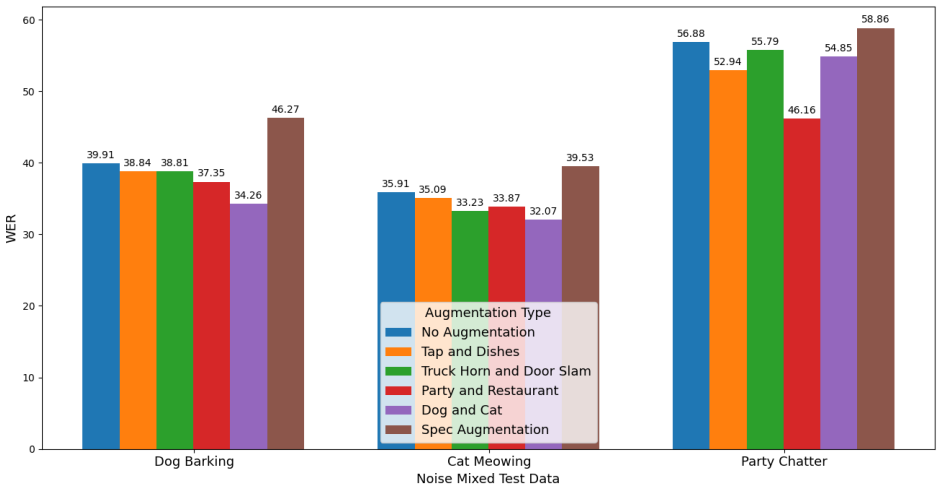

Figure 5.3: WER Comparison of raw data augmentations for the DNN Baseline model on the dog-barking, cat-meowing, and, party chatter noises

Figure 5.3 visualizes the WER comparison of the raw data augmentations for the DNN Baseline model on the dog-barking, cat-meowing, and, party chatter noises. In this graph, we can infer that the dog-barking and cat-meowing noises perform better with the dog-barking and cat-meowing augmentation when compared with other data augmentation types whereas,the party chatter noise perform better with the party chatter and restaurant chatter augmentation. This graph shows that targeted raw data augmentation improves ASR performance on all the three noises. All the three noises provides a better improvement in the ASR performance (WER) on all the raw

data augmentation types when compared with no augmentation.The spectrogram augmentation [4] doesn't improve the ASR performance when compared to the raw-data augmentation types.

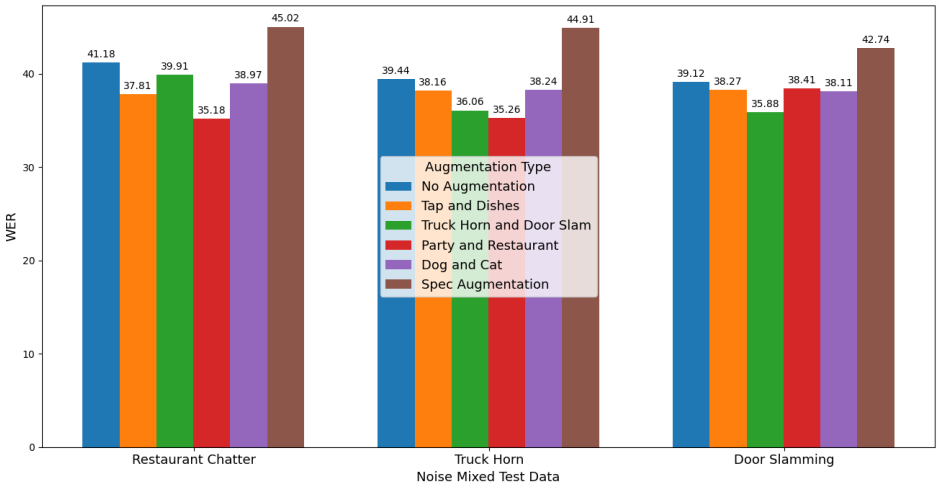

Figure 5.4: WER Comparison of raw data augmentations for the DNN Baseline model on the restaurant chatter, truck horn, and, door slamming noises

Figure 5.4 visualizes the WER comparison of the raw data augmentations for the DNN Baseline model on the dog-barking, cat-meowing, and, party chatter noises. In this graph, we can infer that the truck horn and door slamming noises perform better with the truck horn and door slamming augmentation type when compared with other data augmentation types whereas,the restaurant chatter noise has a huge improvement in the ASR performance with the party chatter and restaurant chatter augmentation type. All the three noises provides a better improvement in the ASR performance (WER) on all the raw data augmentation types when compared with no augmentation. The spectrogram augmentation [4] doesn't improve the ASR performance when compared to the raw-data augmentation types.

5.2.2 Mechanical and Non-Mechanical Data Augmentation

The results with mechanical and non-mechanical raw data augmentations for the Tamil language were performed using the Kaldi toolkit [6]. Tables 5.8 and 5.9 provides the results of baseline models like SGMM, and DNN that are trained on different types of augmentations and tested on the noise-mixed test sets and the best WER from each of the model is tabulated.

Augmentation Type	Acoustic Model	Clean Data	**Continuous Noise**		**Punctuated Noise**	
			Run-Tap	**Dishes**	**Truck horn**	**Door-Slam**
No Data Augmentation	DNN	32.58	47.18	41.54	39.44	39.12
	SGMM	36.15	54.4	48.18	45.24	44.94
Mechanical	DNN	31.88	36.25	36.47	35.74	36.25
	SGMM	36.16	48.84	44.43	40.56	40.52
Non-mechanical	DNN	31.47	37.41	37.88	34.76	37.41
	SGMM	35.19	51.38	45.23	40.81	42.94

Table 5.8: ASR results for the mechanical and non-mechanical augmentations tested on mechanical noises

From the table 5.8, we can infer that both augmentation type improves the ASR performance (WER) regardless of the architectures but the mechanical augmentation types performs better than the non-mechanical augmentation type due to the baseline models being trained on mechanical augmented data. The DNN baseline model produces the best WER (lowest) for all the test sets in the table. Both the punctuated noises (truck horn and door slamming) provides a better improvement in the ASR performance (WER) than the continuous noises on both these augmentation types. Both the punctuated noises improve the ASR performance by a small margin. All the non-mechanical noises provides a better improvement in the ASR performance (WER) on all the augmentation types in the table when compared with no augmentation.

From the table 5.9, we can infer that both augmentation type improves the ASR performance (WER) regardless of the architectures but the non-mechanical augmen-

Augmentation Type	Acoustic Model	Clean Data	**Continuous Noise**		**Punctuated Noise**	
			Party	**Restaurant**	**Dog-bark**	**Cat-Meow**
No Data Augmentation	DNN	32.58	56.88	41.18	39.91	35.91
	SGMM	36.15	66.05	47.75	45.19	41.26
Mechanical	DNN	31.88	52.94	37.81	38.01	33.94
	SGMM	36.16	63.42	45.87	42.19	39.59
Non-mechanical	DNN	31.47	55.79	39.91	33.46	31.94
	SGMM	35.19	63.47	46.15	38.06	36.46

Table 5.9: ASR results for the mechanical and non-mechanical augmentations tested on non-mechanical noises

tation types performs better than the mechanical augmentation type due to the baseline models being trained on non-mechanical augmented data. The DNN baseline model produces the best WER (lowest) for all the test sets in the table. Both the punctuated noises (dog-barking and cat-meowing) provides a better improvement in the ASR performance (WER) than the continuous noises on both these augmentation types. Both the punctuated noises improve the ASR performance by a small margin.All the non-mechanical noises provides a better improvement in the ASR performance (WER) on all the augmentation types when compared with no augmentation.

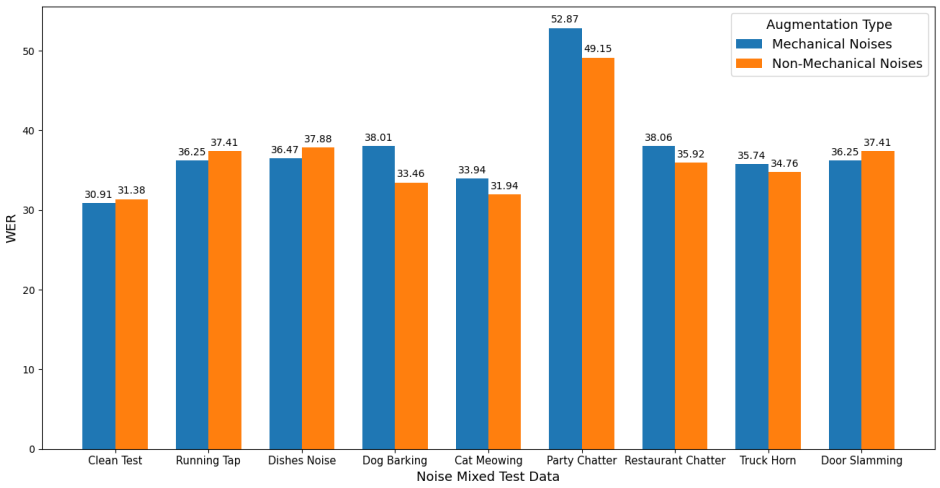

Figure 5.5: WER Comparison of mechanical and non-mechanical data augmentations for the DNN Baseline model

Figure 5.5 visualizes the WER comparison for the mechanical and non-mechanical augmentations for the DNN baseline model.In this graph, we can infer that the mechanical noises (run-tap, dishes, truck-horn, door-slam) provides better performances in mechanical augmentation and similarly non-mechanical noises perform better in non-mechanical augmentations.

Chapter 6

Conclusions

The goal of this research is to investigate the impact of noise on Automatic Speech Recognition models using a low-resource Tamil language dataset. We also look into whether specific data augmentation techniques are better suited to different types of noise (For example, in a car vs. a personal assistant). The baseline models like GMM-HMM, SGMM, SAT-Triphone, DNN were trained using the low-resource dataset. We investigated the impact of noise by mixing several kinds of noise to all the audio files in the test data and evaluated the performance on the trained baseline models. We discovered that all noises, regardless of kind, had an impact on ASR performance, and that upgrading the architecture alone was unable to mitigate the impact of noise. To reduce the impact of noise in ASR models, we implemented few of the data augmentation techniques like Raw data augmentation and spectrogram augmentation (SpecAugment)[4] using the low-resource dataset. The noise sample was mixed with all of the audio files in the training set for raw data augmentation, resulting in a noise-mixed augmented training set that was utilized to train the baseline models. In the case of Spectrogram augmentation (SpecAugment)[4] , the frequency was masked for all the audio files in the training data and this set was utilized for training. We discovered that raw data augmentation improves the WER and thereby reduced the impact of noise considerably when compared with the SpecAugment [4]. Raw data augmentation improves ASR performance on the clean test data and the noise-mixed

data with the DNN model. SpecAugment [4] produces higher WER on both clean and noise-mixed data even without any augmentation type. We also found that targeted raw data augmentation improves ASR performance in general, as evident from most of the raw data augmentation types. Both mechanical augmentation and non-mechanical augmentation type are helpful in recognizing data that contains any of the mechanical noises and non-mechanical noises respectively. Therefore data augmentation techniques would be a better approach to improve the ASR performance and also reduce the impact of several kinds of noises.

6.1 Future Scope

- We can explore the data augmentation techniques on a larger dataset to find out the ASR performance with the addition of noise.
- Investigating with a new set of different kinds of noises on the ASR architectures as to how it affects the ASR performance.
- Evaluating additional types of data augmentations to check if there is any improvements with the ASR performance in the presence of noise.
- To explore SpecAugment with time-masking and a combination of time-masking and frequency masking.

Bibliography

[1] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann, "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 114–126, 2012.

[2] S. Ruan, J. O. Wobbrock, K. Liou, A. Ng, and J. A. Landay, "Comparing speech and keyboard text entry for short messages in two languages on touchscreen phones," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–23, 2018.

[3] B. M. L. Srivastava, S. Sitaram, R. Kumar Mehta, K. Doss Mohan, P. Matani, S. Satpal, K. Bali, R. Srikanth, and N. Nayak, "Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages," *Proc. 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pp. 11–14, 2018.

[4] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019.

[5] H. M. Fayek, "Speech processing for machine learning: Filter banks, mel-frequency cepstral coefficients (mfccs) and what's in-between," 2016. [Online]. Available: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

[6] B. Varadarajan, D. Povey, and S. Chu, "Quick fmllr for speaker adaptation in speech recognition," pp. 4297 – 4300, 05 2008.

[7] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[8] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition," *Speech Communication*, vol. 54, pp. 543–565, 05 2012.

[9] T. N. Sainath, B. Kingsbury, A. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," *CoRR*, vol. abs/1309.1501, 2013. [Online]. Available: http://arxiv.org/abs/1309.1501

[10] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98,

1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230898900432

[11] M. Ravanelli, T. Parcollet, and Y. Bengio, "The pytorch-kaldi speech recognition toolkit," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6465–6469.

[12] N. V. Prasad and S. Umesh, "Improved cepstral mean and variance normalization using bayesian framework," *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 156–161, 2013.

[13] A. Martinez and A. Kak, "Pca versus lda," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[14] H. Erdogan, "Subspace kernel discriminant analysis for speech recognition," 2004.

[15] A. Yuliani, R. Sustika, R. Yuwana, and H. Pardede, "Feature transformations for robust speech recognition in reverberant conditions," pp. 57–62, 10 2017.

[16] M. Sarma and K. K. Sarma, "Acoustic modeling of speech signal using artificial neural network: A review of techniques and current trends," *Intelligent Applications for Heterogeneous System Modeling and Design*, pp. 282–299, 2015.

[17] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[18] M. Fabien, "Introduction to automatic speech recognition (asr)," 2020. [Online]. Available: https://maelfabien.github.io/machinelearning/speech-reco/

[19] D. A. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*, 2009.

[20] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 1209–1213, 01 2014.

[21] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[22] Y. Wang, X. Deng, S. Pu, and Z. Huang, "Residual convolutional CTC networks for automatic speech recognition," *CoRR*, vol. abs/1702.07793, 2017. [Online]. Available: http://arxiv.org/abs/1702.07793

[23] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *CoRR*, vol. abs/1303.5778, 2013. [Online]. Available: http://arxiv.org/abs/1303.5778

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[25] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[26] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st ed. USA: Prentice Hall PTR, 2000.

[27] A. Pervaiz, F. Hussain, H. Israr, M. A. Tahir, F. R. Raja, N. K. Baloch, F. Ishmanov, and Y. B. Zikria, "Incorporating noise robustness in speech command recognition by noise augmentation of training data," *Sensors (Basel, Switzerland)*, vol. 20, 2020.

[28] Y. Hu, N. Hou, C. Chen, and E. S. Chng, "Interactive feature fusion for end-to-end noise-robust speech recognition," *arXiv preprint arXiv:2110.05267*, 2021.

[29] U. Shrawankar and V. M. Thakare, "Adverse conditions and ASR techniques for robust speech user interface," *CoRR*, vol. abs/1303.5515, 2013. [Online]. Available: http://arxiv.org/abs/1303.5515

[30] M. Giurgiu and A. Kabir, "Improving automatic speech recognition in noise by energy normalization and signal resynthesis," in *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, 2011, pp. 311–314.

[31] K. Kinoshita, T. Ochiai, M. Delcroix, and T. Nakatani, "Improving noise robust automatic speech recognition with single-channel time-domain enhancement network," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7009–7013.

[32] S. Gupta, K. M. Bhurchandi, and A. G. Keskar, "An efficient noise-robust automatic speech recognition system using artificial neural networks," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 1873–1877.

[33] T. Liu, M. Gao, F. Lin, C. Wang, Z. Ba, J. Han, W. Xu, and K. Ren, "Wavoice: A noise-resistant multi-modal speech recognition system fusing mmwave and audio signals," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 97–110.

[34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.

[35] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," p. 517–520, 1992.

[36] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, 2016.

[37] D. Povey, H.-K. Kuo, and H. Soltau, "Fast speaker adaptive training for speech recognition." 01 2008, pp. 1245–1248.

[38] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, "Subspace gaussian mixture models for speech recognition," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4330–4333.

[39] D. Povey, S. M. Chu, and B. Varadarajan, "Universal background model based speech recognition," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 4561–4564.

[40] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 2345–2349, 01 2013.

[41] P. Cosi, "A kaldi-dnn-based asr system for italian," 07 2015, pp. 1–5.