Speech Enhancement and Speaker Separation for Distant Speech Recognition

M.Tech Dissertation

Submitted in partial fulfillment of the requirements for the degree of

Master of Technology

by

Pratheek Suresh

(Roll No. 173074009)

Under the guidance of

Prof. Rajbabu Velmurugan

Department of Electrical Engineering

Indian Institute of Technology Bombay

June 2019

Approval Sheet

This is to certify that the dissertation titled **Speech Enhancement and Speaker Separation for Distant Speech Recognition** by **Pratheek Suresh (173074009)** is approved for the degree of **Master of Technology** in Electrical Engineering with a specialization in **Electronic Systems**.

Examiner1 : Signature: Ashish Panda.

Examiner2 : Signature: Preeti Rao

Chairperson : Signature: Preeti Rao

Supervisor : Signature: V. Ri B

Declaration

I declare that this written submission represents my ideas in my own words. I have adequately cited and referenced the sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke punitive action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Pratheek Suresh
1743074009
Department of Electrical Engineering
IIT Bombay

Date: 19/06/2019

Acknowledgement

I want to express my deep sense of gratitude to Prof. Rajbabu Velmurugan, Department of Electrical Engineering, IIT Bombay, for giving me the opportunity to work in this project. His invaluable guidance and support by providing necessary information regarding the topic have helped me progress in the right direction in completing my thesis. His valuable advice and suggestions have always been a motivation for my work. I would also like to express my indebtedness to Prof. Preeti Rao for her constant guidance, insightful comments, and encouragement.

I want to extend my heartfelt gratitude to my friends Nikhil Mohan and Sachin Nayak for their valuable suggestions, discussions, and encouragement throughout my project. I would also like to thank all my friends and all those who directly or indirectly helped me in my project.

Last but not least, I owe my gratitude to my family for their constant support and prayers throughout these two years of study.

Pratheek Suresh
Electrical Engineering
IIT Bombay

Abstract

In the age of technology and connected devices, ways in which humans interact with these devices have changed from having physical buttons to touch sensors, to voice commands. To understand and interpret commands such devices need to understand speech. Speech recognition systems play an essential role in how humans interact with these devices. There are also applications where this speech needs to be converted to text and a multi-speaker transcription system used for meetings is one such.

In such applications, meetings recorded using a microphone / an array of microphones is processed to get the transcripts. The system is envisaged to process data offline from the recordings in non-real time to produce the transcripts. The processing involves speech enhancement, separating speakers in the recording and speech-to-text conversion of each individual speakers. Speech enhancement is essential for improving the signal quality by removing noise such as noise from Air Conditioners and de-reverberation (to remove room reflections). Speaker separation is used to separate simultaneous conversation from the enhanced speech. This work focuses on unsupervised speech enhancement using weighted prediction error (WPE) [1] and speaker separation using a supervised approach. We achieve this by using a multi-channel speech enhancement followed by a Fully connected Deep Neural Network(DNN) for separation. This work comprises multi-channel signal enhancement followed by supervised speech separation using Deep Neural Networks(DNN's). Speech Enhancement results are shown on TCS meeting dataset [2] and Speaker separation results are presented using open source dataset GRID corpus [3]. The WPE algorithm was effective in suppressing reverberation. The DNN based approach for separating individual speakers performed well in terms of objective measures and also improved the word error rate.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Speech recognition first introduced in 1952 [4], has since evolved to recognize sentences from multiple languages and speakers better. There is a multitude of devices which we interact through voice nowadays, such as mobile phones, voice assistants, and car audio systems. There are commercial systems that convert speech into text so that it can be archived for later reference. The main challenges faced by these systems are ambient noise, reverberation due to the closed environment of a room and multiple speakers speaking simultaneously [5]. One approach to minimize these external disturbances is to keep the microphones as close to the speaker and then use a filter to remove unwanted noise. However, this solution may not be feasible in all scenarios. An alternate technique is to place a microphone(s) away from speakers and using array processing techniques to reduce noise and suppress interference. The above step is referred to as a distant speech recognition system (DSR).

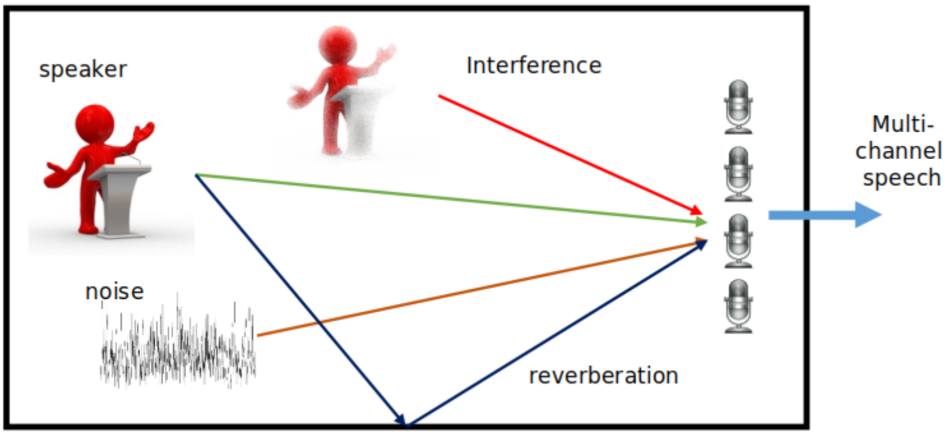

Figure 1.1: Typical speech recognition scenario using an array of microphones with multiple speakers, ambient noise and reverberation

One such multi-microphone setup, as shown in Figure 1.1 is well suited for the multi-speaker separation. Using the speech data from the microphones, using signal processing techniques, we can remove noise and reverberation. In this work, we use a multi-channel based de-reverberation technique WPE for reducing the effect of room reflections. In this approach, we train a Deep Neural Network to suppress the interfering speaker. The challenge of using DNN's are the need for extensive data set for training the network to achieve reliable performance. This work uses speaker mixtures created from single-channel open source speech data sets to train the network to achieve excellent performance. Generally, in a meeting scenario, not more than two speakers talk simultaneously. The systems can be designed to output the individual speaker channels from the speech mixture received by the microphones. Additionally, by using the multi-microphone arrangement, the spatial information provided by the arrangement aids in separating the individual speaker.

1.1 Block Diagram

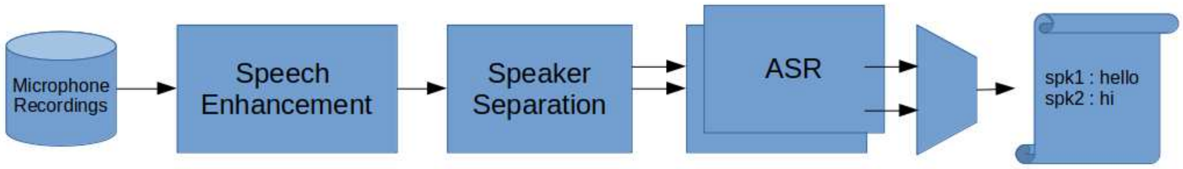

Figure 1.2: System block diagram showing speech enhancement of microphone(s) recordings, speaker separation and conversion of speech-to-text

The final goal of our system is to create an automatic meeting/ conversation transcription system 1.2. This system involves identifying and separating the speaker from the speech recorded using an array of microphones. The main modules envisaged for the project are speech enhancement module, speaker identification, speaker separation, automatic speech recognition. This report concentrates on the speaker separation problem involving two speakers who are simultaneously speaking, which is similar to the scenario in a conversation or meeting scenario.

1.2 Report Outline

The final objective of the project is briefly described with datasets and the processing. The following chapter 2 discusses the microphone signal enhancement for de-noising and de-reverberation. The chapter also describes in detail the implementation of a system to evaluate the performance of these techniques on real meeting recordings. The problem of speaker separation is described in chapter 4. As a step towards this, we attempted a relatively easier problem of separating vocals from music mixtures (containing vocals and instruments). This was continued with a study to explore the configuration of the neural network required, its training procedure. The study is described the next section 3.1 in detail and results. The problem of separating simultaneous speakers and our approach towards this is described in section 3.2. In chapter 4, we summarize the proposed architecture and discuss the overall results. The final chapter 5 points to future improvements and suggestion for continuing the research.

Chapter 2

Speech Enhancement using Weighted Prediction Error

Speech enhancement intends to improve speech quality by using audio signal processing techniques so that that overall intelligibility and perceptual quality of degraded speech signal is bettered [6]. This technique is used in almost all by applications, e.g., mobile phones, teleconferencing system, hearing aids, and ASR systems.

Speech de-noising

Ambient noise affects the perceived quality of the speech signal. Hence, it is essential to improve quality by reducing ambient noise [7]. Multiple techniques have been employed for this purpose, such as signal subspace method, spectral subtraction, Wiener Filtering, and adaptive noise canceling. The performances of speech enhancement techniques are evaluated by the quality and intelligibility of the processed speech signal. Speech signal-to-noise ratio (SNR) improvement is considered as one of the measures of improvement.

Speech de-reverberation

Reverberation is a phenomenon where the signal received consists of both direct sound and the reflected sound from the boundaries of the room or object present in the room, as shown in Figure 2.1. Reverberation effects are desired for concert and music. However, it has an undesirable effect in DSR systems as it can degrade the speech quality due to erroneous DOA estimates and hence the source localization performance [8].

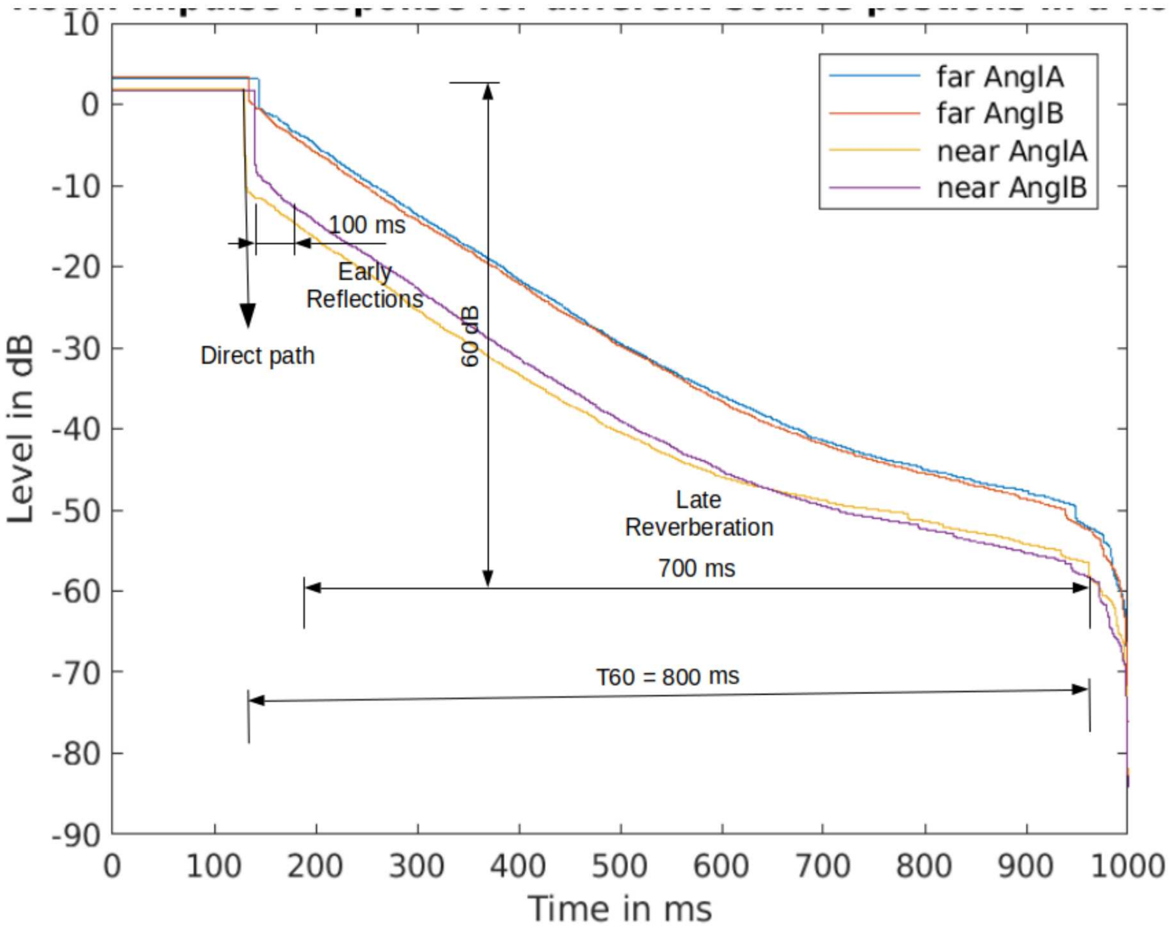

Figure 2.1: Plot of recorded reverberation signal energy in a room for a sound impulse, with 2 distances between a speaker and microphone array (near = 50 cm and far = 200 cm) for different source positions, shows direct path, early and late part of reverberation

Reverberation is characterized by a gradual decay of the signal. Reverberation is specified in terms of Reverberation Time T_{60}; the time taken by the signal to decay by 60 dB of its original level. There is a good rationale behind this constant since the loudest sound for most music is about 100 dB and a typical room background level is about 40 dB. The T_{60}, as shown in Figure 2.1, depends on the size of the room, the frequency of the signal, and various parameters like absorption coefficient and temperature. T_{60} of 200 ms and above degrades the performance of ASR systems [9]. So de-reverberation plays an vital role in improving ASR. There have been multiple approaches to suppress reverberation; one of the widely used algorithms is WPE.

2.1 Speech de-reverberation using Weighted Prediction Error (WPE)

Reverberant data degrades the performance of source localization and hence affect the ASR results. The WPE algorithm [10] [1] uses a statistical approach to remove the late part of reverberation using the multi-microphone signal, without any prior information of the RIR. The speech signal is assumed to be generated using a Gaussian modelled process and the estimate is achieved using a delayed linear prediction with Maximum Likelihood Estimation (MLE). The time-varying characteristic of the speech is compensated in the estimate to an extent by normalizing each speech frame. The algorithm estimates an inverse system to cancel the effects of late reverberation. The estimator is robust such that the convergence is achieved within a few seconds of utterance.

$$x_m(n) = \sum_{k=0}^{L_h-1} h(k,m)s(n-k) \tag{2.1}$$

$$d_m(n) = \sum_{k=0}^{D-1} h(n,m)s(n-k) \tag{2.2}$$

$$r_m(n) = \sum_{k=D}^{L_h-1} h(k,m)s(n-k) \tag{2.3}$$

$$x_m(n) = d_m(n) + r_m(n) \tag{2.4}$$

$$\hat{d}_m(n) = x_m(n) - (\hat{\mathbf{C}})^T x_m(n-D) \tag{2.5}$$

The degradation and enhancement obtained using this approach are briefly discussed next. The observed signal at the m-th channel $x_m(n)$ can be modelled as (2.1) where m, L_h correspond to microphone index and RIR length, respectively. $h(n,m)$ and $s(n)$ represent the time domain RIR for m-th channel and clean speech, respectively. $d_m(n)$ in (2.2) corresponds to the received clean speech plus the early reverberation part. $r_m(n)$ in (2.3) is the undesirable late reverberation and $x_m(n)$ in (2.4) expresses the observed signal as the sum of early to late. The early and late part of the reverberation is separated by using a D sample index, which splits the impulse response into two parts. (2.5) shows that the desired signal can be estimated from the previously observed samples, where $(\hat{\mathbf{C}})^T$ is the estimated regression coefficients using maximum likelihood estimation (MLE). This process of estimating the signal is referred to as WPE. The WPE algorithm can be applied on both single channel and multi-channel data for signal enhancement.

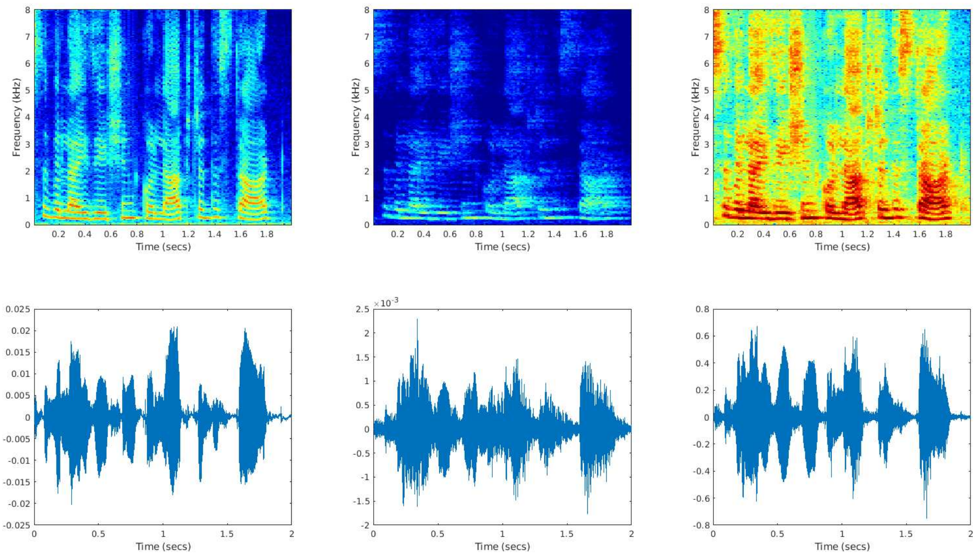

Figure 2.2: Figure showing two second data for Room of dimension 6.67 m x 6.14 m x 6.57 m, $T_{60} = 600$ ms, source angle of 0° degree from the microphone array. Clean speech spectrogram is on the left, reverberate signal is at the centre and WPE enhanced signal is shown on right.

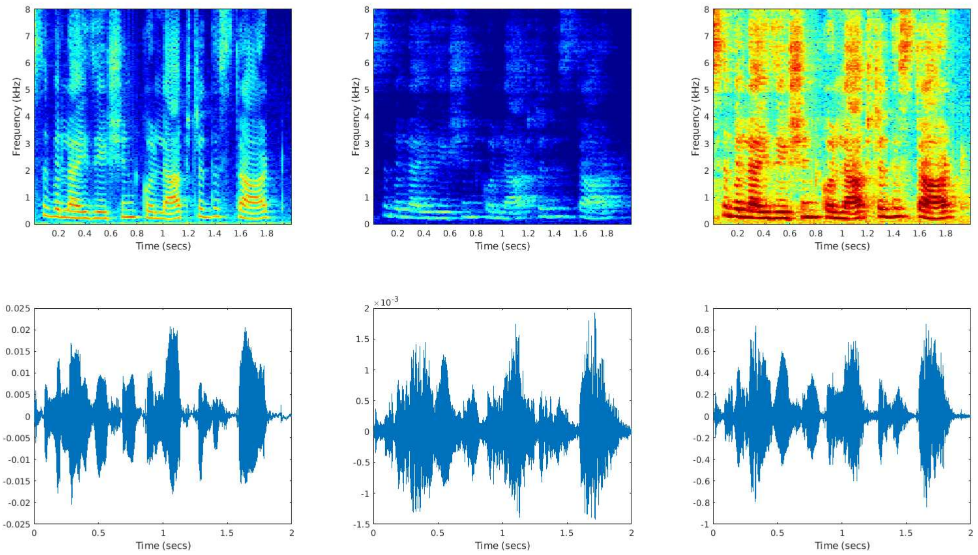

Figure 2.3: Figure showing two second data for Room of dimension 6.67 m x 6.14 m x 6.57 m, $T_{60} = 600$ ms, source angle of 45° from the microphone array. Clean speech spectrogram is on the left, reverberant signal is at the centre and WPE enhanced signal is shown on right.

The Figures 2.2 and 2.3 shows the de-reverberation using WPE on a reverberant speech signal. On the left shows clean speech signal, center shows signal with reverberation and the right shows signal enhanced by WPE. We can clearly see the removal of spectral smearing due to reverberation. The Reverberation time and the room size for both the experiment are same, but the source spatial location is changed from 0° to 45° respectively concerning a microphone array to add variability in terms of reverberation effects as shown in Figure 2.1.

2.2 Performance Evaluation of WPE based enhancement

The performance of WPE was analyzed based on the WER of a trained ASR system used for generating text from multi-microphone speech recording of a meeting. The ASR system used ASpIRE [11] chain model(TDNN and BLSTM) available as Kaldi recipe with a custom dictionary and language model.

Three original recordings [2] were available which had three speakers speaking English sentences one after the other, details of which are shown in Table 2.1. The speakers were stationary though their natural movements were not restricted. Each recording was acquired at 48 kHz sampling rate, and for processing the recordings were downsampled to 16 kHz. These recordings were done in a room of dimension 4 m x 4.5 m x 3 m size, with four microphones circular array of radius 5 cm as shown in Figure 2.4 , kept at the center of the room on a circular table and microphone at the right angles to each other on the periphery of the circle.

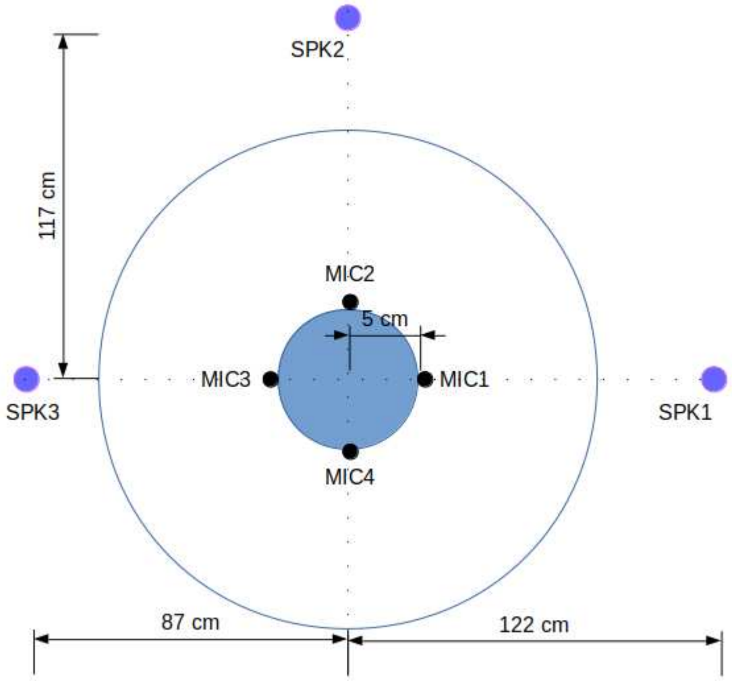

Figure 2.4: Figure shows the arrangement of microphones and positioning of speakers for recordings in a room of dimension 4 m x 4.5 m x 3 m size

Table 2.1: Table showing details of TCS dataset

Recordings	**No. of speaker**	**Gender details**	**Duration**
Recording 1	3	spk1: Female, spk2: Female, spk3: Male	4 min 25 sec
Recording 2	3	All Male	1 min 52 sec
Recording 3	3	spk1: Female, spk2: Female, spk3: Male	4 min 45 sec

The figure 2.5 shows the block diagram of the processing techniques used for the evaluation of WPE performance. Being multi-channel recordings we use beamforming techniques such as Generalized Delay Sum Beamforming(GDSB), Minimum variance Distortion Less Response (MVDR) beamformer [5] and Beamformit [12] to improve the Signal-to-Noise Ratio (SNR) of the speech signal. The WPE can be applied directly to the single channel (SWPE) of multi-channel recordings (MWPE) also; here, both the combinations were analyzed. For a comparison with the single channel performance, we also consider a single channel with maximum SNR for ASR computation.

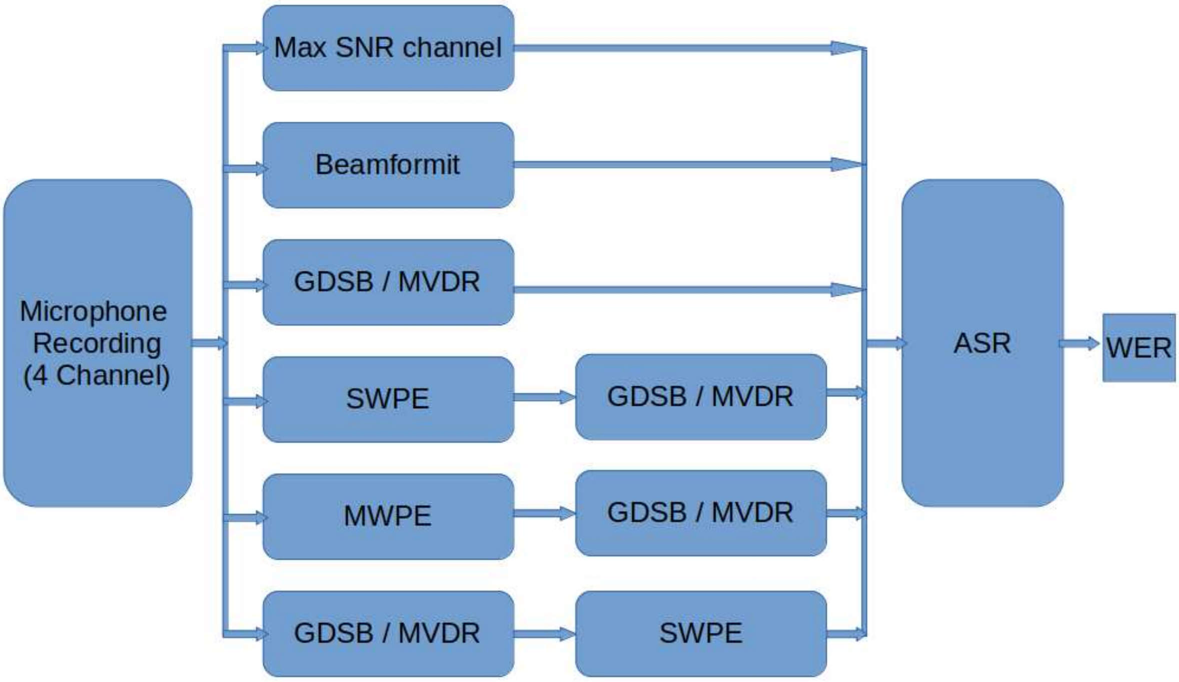

Figure 2.5: Block diagram of the processing employed to evaluate the performance of speech enhancement schemes, where SWPE is Single Channel WPE and MWPE is Multi Channel WPE

Each of these three recordings were processed using different techniques, and a single processed stream of speech was created. This stream is passed into the ASR module to produce the text output. The ASR system used a pre-trained ASpIRE chain model available as the part of TCS dataset [2]. ASpIRE model trained on Fisher English that has been augmented with impulse responses and noises to create multi-condition training [11]. The language model was trained using the text spoken by the user. The text output is compared with the original reference text (actual words uttered by the speaker) to compute the Word Error Rates (WER's).

Table 2.2: Table showing the comparison of WER for different recordings and processing schemes

Room Size = 4 m x 4.5 m x 3 m			
	Word Error Rate		
Processing Techniques	**Recording 1**	**Recording 2**	**Recording 3**
Maximum SNR channel	67.9	33.1	57.9
Beamformit	**21.2**	**24.6**	**28.6**
GDSB	27.4	30.1	30.0
MVDR	**21.2**	26.5	29.6
Single Channel WPE (SWPE) + GDSB	23.7	29.4	31.3
SWPE + MVDR	24.5	28.7	30.8
Multi Channel WPE (MWPE) + GDSB	26.4	27.9	31.5
MWPE + MVDR	24.4	25.0	30.1
GDSB + SWPE	26.2	30.5	31.6
MVDR + SWPE	24.4	25.7	**28.6**

2.2.1 Discussion

As shown in the above Table 2.2, WPE enhancement did not produce significant improvement in WER than other processing techniques. The above can be attributed to the data being reverberation free or less reverberant and also noise in the recording is minimal, which is in agreement with the previous reported results [13].

Chapter 3

Speaker Separation

Speaker separation is the task of separating a target speaker from another speaker or background interference [14]. Speaker separation traditionally is studied as a signal processing problem. Speaker separation can also be formulated as a supervised learning problem, where the distinctive patterns of individual speakers are learned from training data. In recent years, many supervised separation algorithms have been tried for solving this problem. In very recent times, deep learning based approaches have improved the performance [14].

Training a DNN involves identifying the right network and parameters for learning, choosing the appropriate input feature vector and target for learning. For speaker separation problems, there are multiple features available as input and targets [14]. we have chosen log-magnitude spectrogram of the mixture as the input and log-magnitude spectrogram of individual speakers as targets. The commonly used cost functions for regression problems are are mean square error (MSE) as in (3.1), mean absolute error (MAE) as in (3.2).

$$L_{mse} = \frac{1}{N} \sum_{i\ =\ 1}^{N} (y_i - \widehat{y_i})^2 \tag{3.1}$$

$$L_{mae} = \frac{1}{N} \sum_{i\ =\ 1}^{N} |y_i - \widehat{y_i}| \tag{3.2}$$

where $\widehat{y_i}$ and y_i are the predicted output and desired output for neuron i, respectively [14].

The following section discusses the different architecture followed and input/targets used for training for two types of separation problems.

3.1 Study on vocal separation from music

As a precursor to the problem of separating two simultaneous speech source from its mixture, a similar but simpler problem was attempted to separate singing vocals from music recordings comprising of vocal, drums, bass and other instruments [15].

3.1.1 Dataset

The MUSDB18 [16] dataset contains a total of 150 songs of distinctive genres. It has stereo channels for all streams, and the songs are divided between a training and test subsets. This dataset can be used for estimating sources from the mixture, e.g., karaoke generation and has been widely used in competitions where source separation algorithms are evaluated [17]. The train set consists of 100 songs and test set consists of 50 songs. Files are in Native Instruments stems format (.mp4), which composes of 5 stereo streams. These signals correspond to the mixture(stream 1), drums (stream 2), bass (stream 3), rest of the accompaniment (stream 4), vocals (stream 5). The mixture is the sum of vocal, drums and bass. The sampling rate is 44.1 kHz or CD quality.

The problem of separating vocal from music was done using a Deep Neural Network (DNN) [18] trained on mixture spectrogram as input and the vocal spectrogram as targets [19]. The mixture and the vocal spectrograms of all the 150 songs were computed. The parameters used for spectrogram are window size = 46.4 ms with a hop size of 23.2 ms (50 % overlap). The spectrograms are quantized to 8-bit values to reduce the memory requirement for storage.

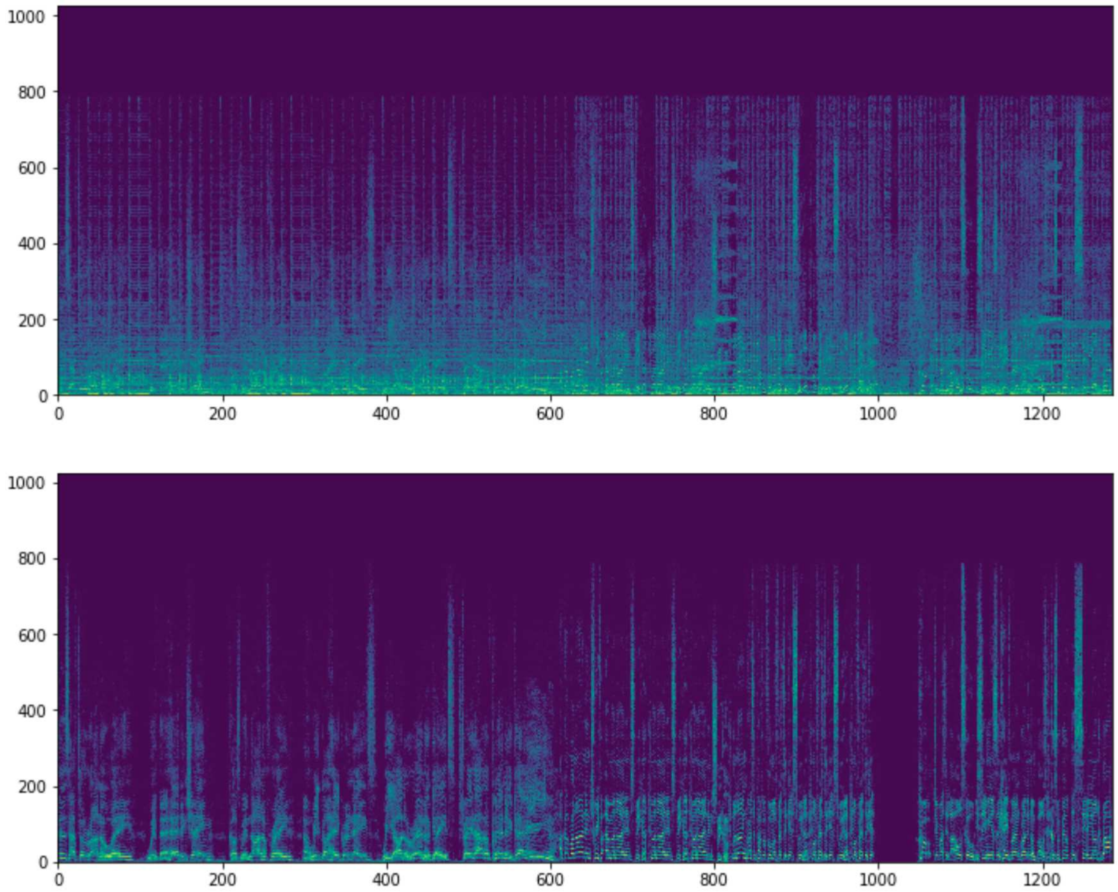

Figure 3.1: Figure shows the 1285 frames (6 seconds) of mixture spectrogram on top and target vocal spectrogram at bottom of a songs from the MUSDB18 dataset.The mixture spectrogram is the input to the DNN and vocal spectrogram is the target used to train the network

3.1.2 DNN & Training

The network [20] used for separating the vocal from the mixture consists of an input layer, followed by three layers of Long Short-term Memory (LSTM) [21] and a fully connected layer, as shown in Figure 3.2. A data loader creates a batch of 16, 128 frames spectrogram randomly as in Figure 3.1 from the train set for the input and target to the network being trained. The data loader also de-quantize the spectrograms while creating the batch. The learning rate was at 1e-3, optimizer used was Root Mean Square Propagation (RMSprop) as shown in (3.3) and 3.4 and Loss function used was Root Mean Square Error(RMSE) as shown in Table 3.1. The network was trained for 1000 iterations. The whole system was developed in python using pyTorch [22] packages for neural networks. PyTorch is a Python package that provides a way to compute Tensor using Graphics Processing Units (GPU's) and to build DNN's using tape-based auto grad system.

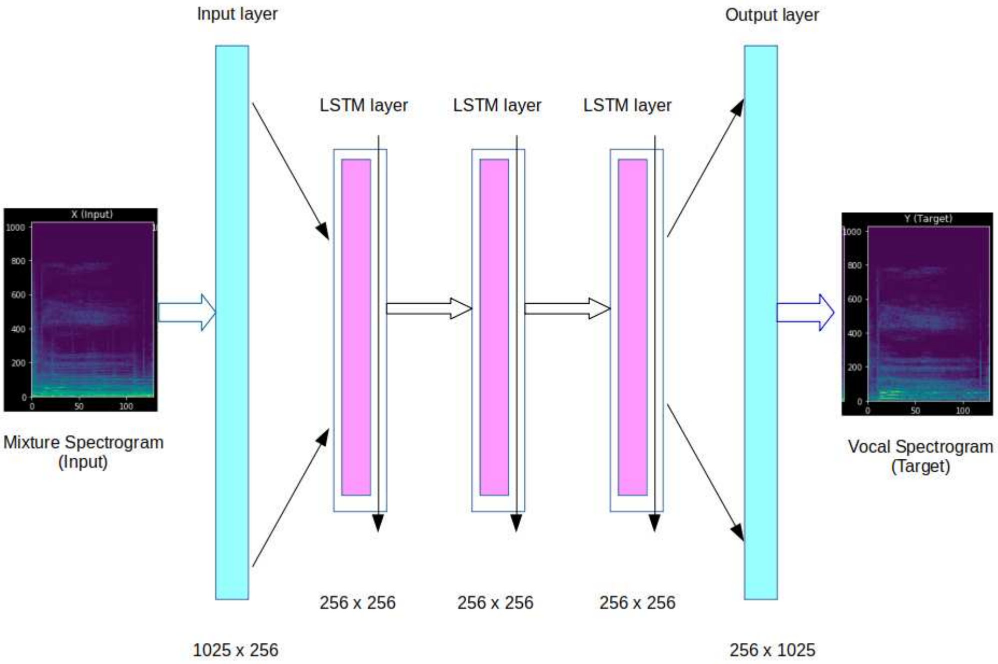

Figure 3.2: Figure shows the DNN architecture for vocal separation using MUSDB18 corpus having an input layer of size 1025, 3 stacked LSTM layer of size 256 and output layer of size 1025

RMSProp [23] is a technique by which the learning rate changed adaptively. The algorithm divides the learning rate for a weight by the running average of the magnitudes of recent gradients for that weight. At first, the running average of the magnitudes of recent gradients for a weight is calculated in terms of means square.

$$v(w,t) = \gamma v(w,t-1) + (1-\gamma)(\nabla Q_i(w))^2 \tag{3.3}$$

where, γ is the forgetting factor and $\nabla Q_i(w)$ is the gradient [23].

Moreover, the parameters are updated as [23],

$$w = w - \frac{\eta}{\sqrt{v(w,t)}} \nabla Q_i(w) \tag{3.4}$$

Table 3.1: Parameters used for training & validation of MUSDB18 dataset

Parameter Name	**value**
Sampling Frequency	44.1 kHz
Analysis Window	2048 samples (46.4 ms)
Hop size	1024 samples (23.2 ms)
Number of Features	1025 bins
LSTM Layer Size	256
Fully Connected Layer Size	1025
Batch Size	2048 frames
Maximum Iterations	1000
Optimizer	RMSprop
Learning Rate	1e-3
Batch Creation	Randomize

Once the network gets trained, the test data set is used to evaluate the performance of the estimates. The Figure 3.3 shows the results of the spectrogram estimates on the vocal from the mixture. The phase of the signal is recovered from the phase of the mixture. Also, the accompaniment is recovered from the original mixture using Wiener filters.

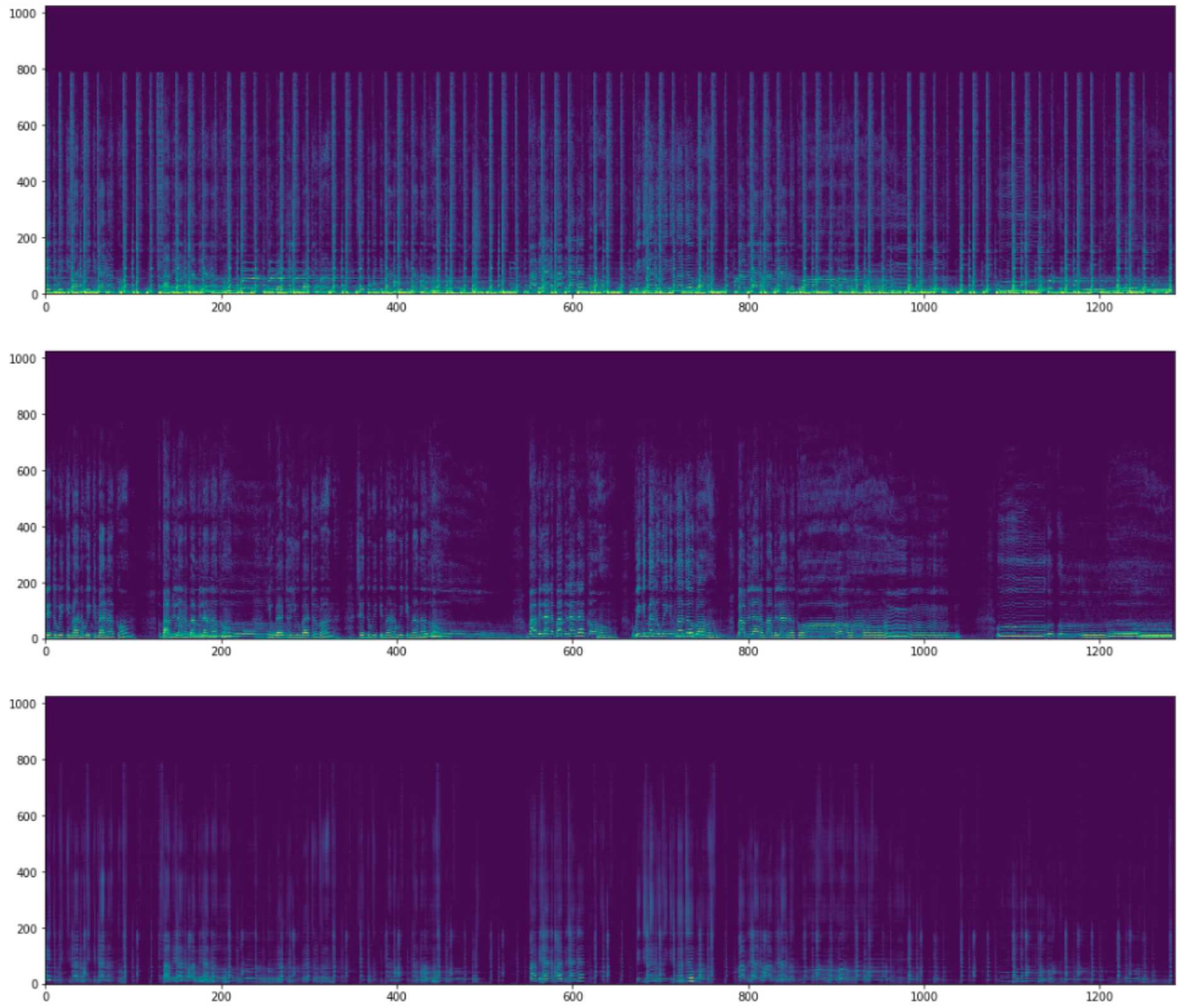

Figure 3.3: Figure shows shows the 1285 frames (6 seconds) the mixture spectrogram on top, vocal spectrogram in middle and the estimated vocal spectrogram at the bottom of a song from the test dataset.

There are other objective measures which can be effectively used to quantify the quality of the signal estimated to the actual signal. BSS Eval [24] is a MATLAB toolbox to measure the performance of source separation algorithms where the source signals are available. The estimated source signal is decomposed to signals corresponding to the target source, interference from unwanted sources, and artifacts such as "musical noise." The measures are "source to distortion ratio" (SDR), "sources to interferences ratio" (SIR), and "sources to artifacts ratio" (SAR) [25] are computed based on the decomposed signal. The higher the value of these estimates, the better the estimate.

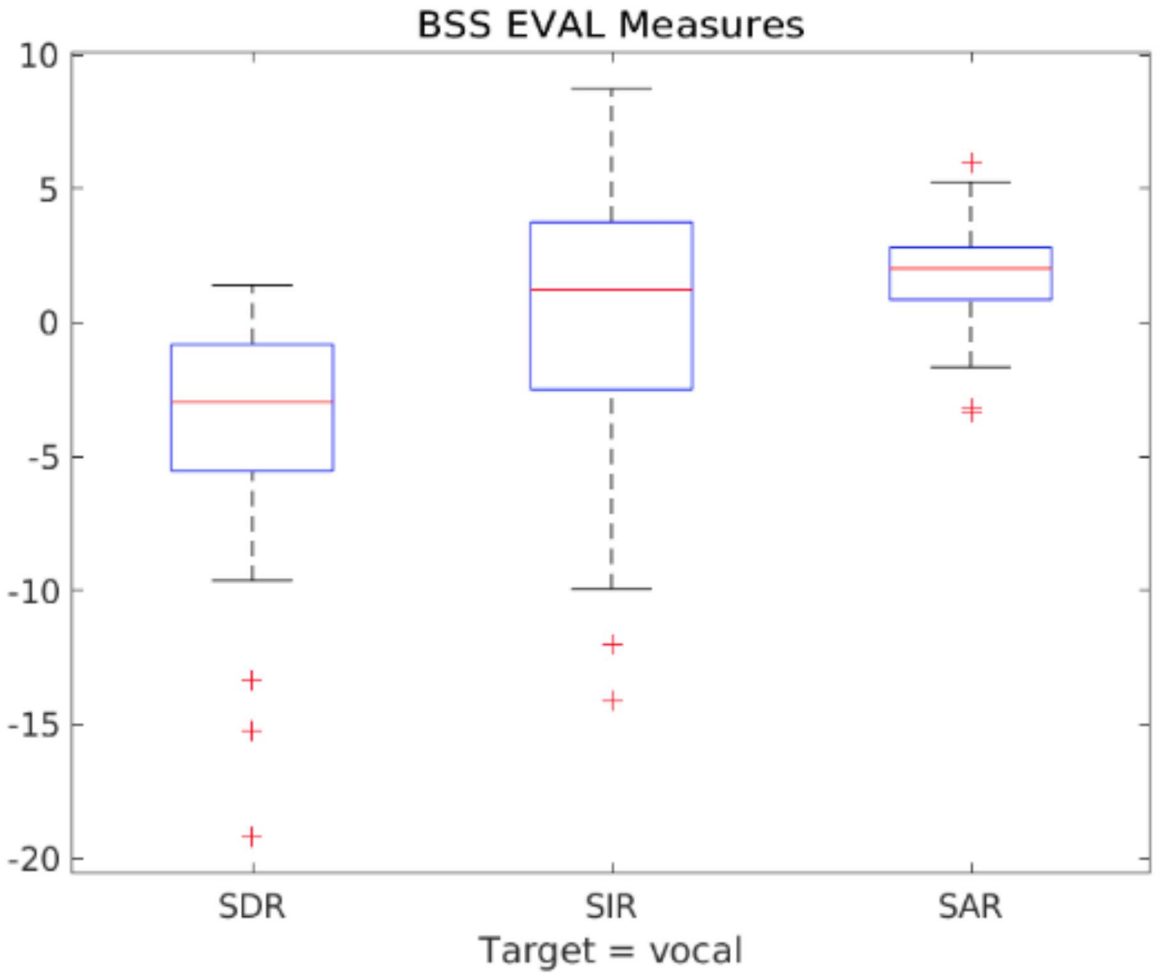

Figure 3.4: Box plot showing the objective measures evaluated for test dataset comprising of 50 songs

The above Figure 3.4 shows that various measures evaluated on 50 songs in the test dataset. The separated vocal had the instruments also present, even though it was highly suppressed. The work on vocal separation has provided insights into the data preparation, batch creation, DNN architectures, and various parameters used for training a neural network. The techniques learned will be applied to solve the problem of speaker separation(Section 3.2) from simultaneous speaker mixtures.

3.2 Speaker Separation

There were several papers in this field which tried to model the wide nonlinear variation between speech features of a target speaker and interferer contained in a mixed signal of both [26] using DNN's. The supervised training approach is to provide the network with input spectrogram of the mixture and give the speaker clean spectrogram as targets. The method is to design a DNN with two output targeting the target speaker and the integer, which provides generalization than only training for target speaker.

3.2.1 Dataset

For the work, the audio-visual corpus [3] created for speech perception and automatic speech recognition studies was used. The corpus consists of audio and video recordings of 1000 sentences spoken by each of 34 individual speakers in the English language. The utterances are simple English phrases like ”place blue at C 3 again”. Each sentence consists of a six-word sequence, as indicated in Table 3.2. Of the parts, three are color, letter, and digits. For alphabets; Only multi-syllabic letter ”W” was omitted. Each speaker spoke all combinations of the three keywords, making to a total of 1000 sentences per speaker. The remaining part command, preposition, and adverb were ”fillers.” Filler positions had four available options. Fillers were chosen to have some variation in contexts for the neighboring keywords. Distinctive gross phonetic classes (nasal, vowel, fricative, plosive, liquid) were used as the initial or final sounds of filler words in each position [3].

Table 3.2: Grid corpus dataset Sentence structure. Keywords are identified with *

command	**colour***	**preposition**	**letter***	**digit***	**adverb**
bin	blue	at	A - Z	1 - 9, 0 (zero)	again
lay	green	by	W excluded		now
place	red	in			please
set	white	with			soon

A total of 16 female and 18 male speakers contributed to the corpus. All the participants spoke with British English accent. The audio corpus was collected using a Bruel & Kjaer (B & K) type 4190 microphone. The corpus was available with a sampling rate of 25 kHz.

The dataset is organized as folders with names ”s1” to ”s34” indicating 34 speakers, each folder has 100 recordings. The table 3.3 below shows the recordings and the gender of the participants. This information will be further used to create the instantaneous mixture for training the DNN’s.

Table 3.3: Gender information of speakers in GRID Corpus dataset

Male			**Female**		
s1	s9	s19	s4	s20	s29
s2	s10	s26	s7	s21	s31
s3	s12	s27	s11	s22	s33
s5	s13	s28	s15	s23	s34
s6	s14	s30	s16	s24	
s8	s17	s32	s18	s25	

3.2.2 Dataset creation

The aim was to capture the speaker variation between the speaker and the interferer using the model to estimate the speaker from an unseen mixture; For this purpose, a dataset was created by mixing a single speaker (e.g., s1) with randomly selected speakers of the opposite gender. Speaker s1 being male, 16 other female speakers are picker randomly for dataset creation. The maximum number of mixtures possible is 16 speaker times 1000 utterances times 1000 utterance of s1. Each utterance is of two seconds long. To make the data more realistic in case of a conservation scenario; the mixture is created by adding both the utterances at varying amplitude level in the range of 0 to 10 dB. A total of 12 hours of data is created; ten hours of the dataset is for testing, 1 hour each dataset is for validation and testing.

3.2.3 DNN Training

The DNN for the speaker separation is developed using Matlab® Deep Learning Toolbox [27]. This toolbox ensured a quick way to prototype the network and train the network. It provides tools necessary to see the training progress, configure various training parameters, and built-in validation options.

The network has three fully connected layers, with an input image layer and regression output layer Figure 3.5. The input layer is of the size 513 (frequency bins) x 7 (past three frames + current frame + future three frames), each fully connected layer has 2048 size, the output layer has a size of 1026 (513 for spk1 + 513 for interfere). The network is trained on mixture log magnitude spectrogram, and the target as the log magnitude spectrogram of the desired speaker and the interfere respectively. The network is trained using an adaptive learning rate and Adaptive Moment Estimation (Adam) as in 3.5 to 3.9 as optimizer. Table 3.4 shows the audio analysis parameter and training parameters used for the network.

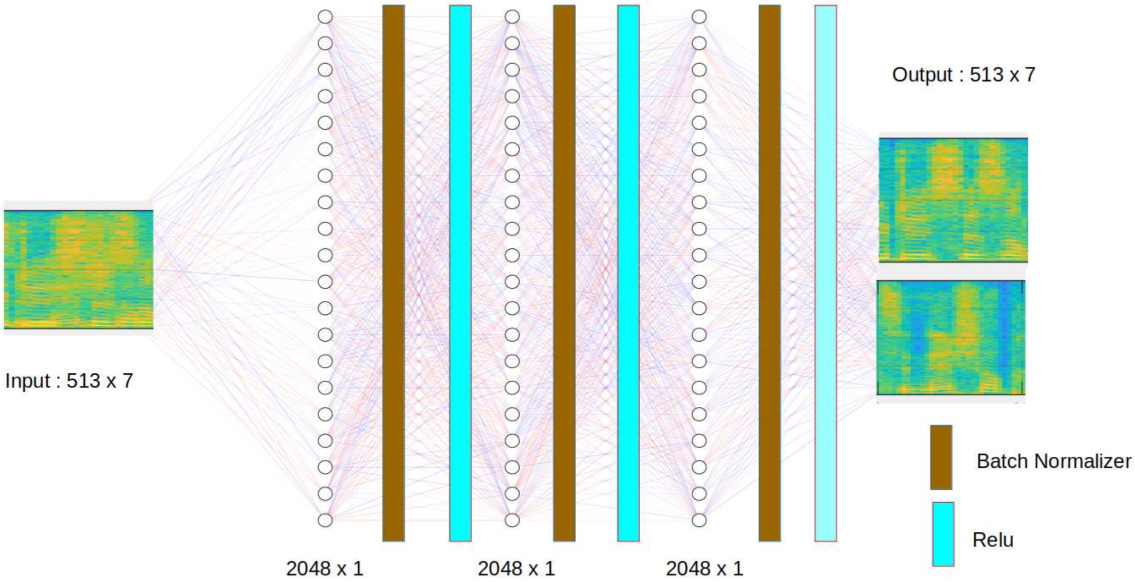

Figure 3.5: Figure showing the DNN architecture used for speaker separation using GRID corpus

Adam [28] algorithm uses running averages of both the gradients and the second moments of the gradients. In the following equations $w^{(t)}$ is the weights, $L^{(t)}$ is the Loss function and t indexes the current training iteration.

The algorithms update is given by [28],

$$m_w^{(t+1)} \leftarrow \beta_1 m_w^{(t)} + (1-\beta_1)\nabla_w L^{(t)} \tag{3.5}$$

$$v_w^{(t+1)} \leftarrow \beta_2 v_w^{(t)} + (1-\beta_2)(\nabla_w L^{(t)})^2 \tag{3.6}$$

$$\hat{m}_w = \frac{m_w^{(t+1)}}{1-(\beta_1)^{t+1}} \tag{3.7}$$

$$\hat{v}_w = \frac{v_w^{(t+1)}}{1-(\beta_2)^{t+1}} \tag{3.8}$$

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w}+\epsilon} \tag{3.9}$$

where ϵ is a tiny scalar to prevent division by 0, and β_1 and β_2 are the forgetting factors for gradients and second moments of gradients.

Table 3.4: Parameters used for Training & Validation

Parameter Name	**value**
Sampling Frequency	16 kHz
Analysis Window	1024 samples (64 ms)
Hop size	512 samples (32 ms)
Number of Features	513 bins
Number of Segments	7
Layer Size	2048
Batch Size	1000 frames with context
Maximum Epoch	50
Optimizer	Adam
Learning Rate	1e-4 (decreasing 10 % after every 2 epoch)
Batch Creation	Randomize every epoch
Validation Frequency	Every Epoch
Early Stopping	after 10 epoch (validation loss does not reduce)

The network took approx 30 minutes to train. Figure 3.6 shows the training progress.

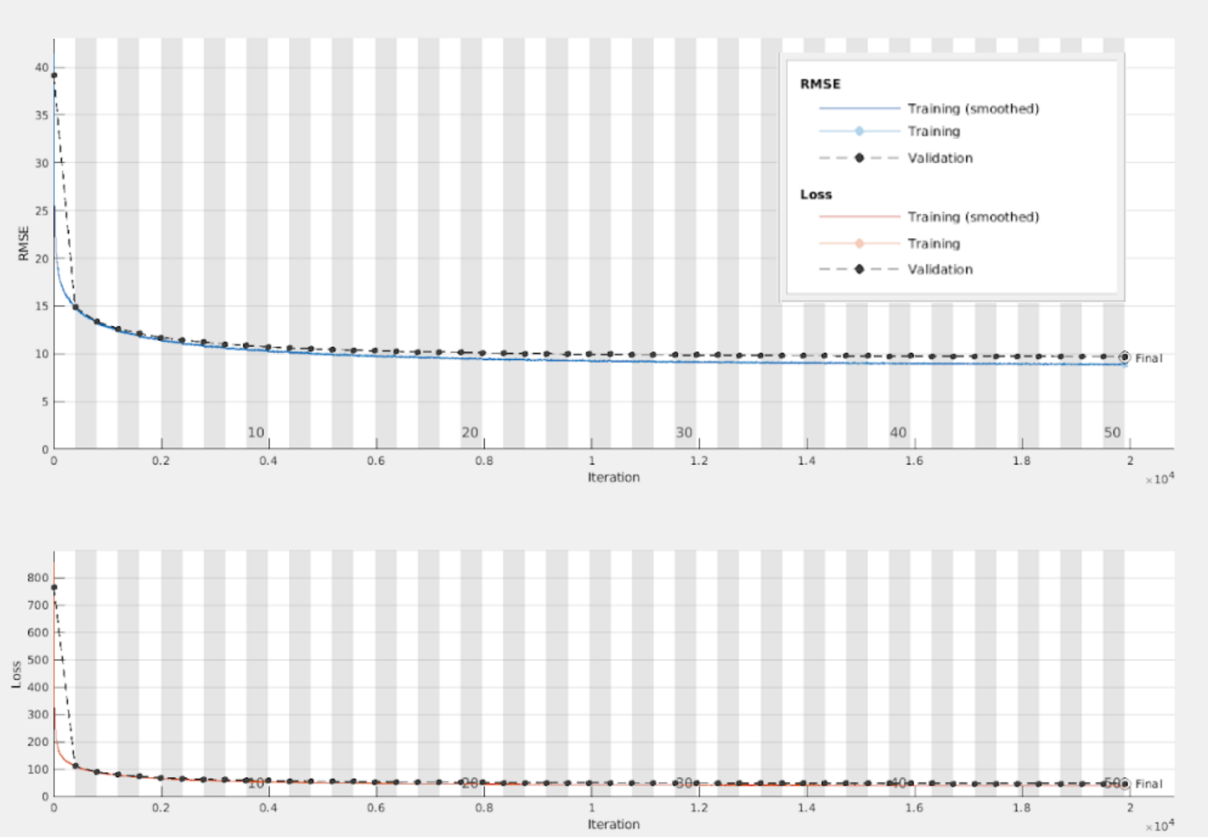

Figure 3.6: Figure shows the training and validation error on top and the corresponding loss at the bottom. The alternate white and grey shades indicates each epochs

3.2.4 Testing & Observations

The testing of the model was done using an hour of utterance from the dataset previously created in section 3.2.2. From the log magnitude spectrogram output, the phase information is added to the utterance by using the phase information recovered from the corresponding mixture files. The separation performance were evaluated using several measures like source to distortion ratio(SDR), sources to interference's ratio(SIR), and sources to artifacts ratio (SAR) [25] shown in Figure 3.7, short-time objective intelligibility (STOI) [29] which assumed to be correlated to speech intelligibility, perceptual evaluation of speech quality (PESQ) [30] shown in Figure 3.8 with a high correlation to subjective scores and the corresponding recognition accuracy.

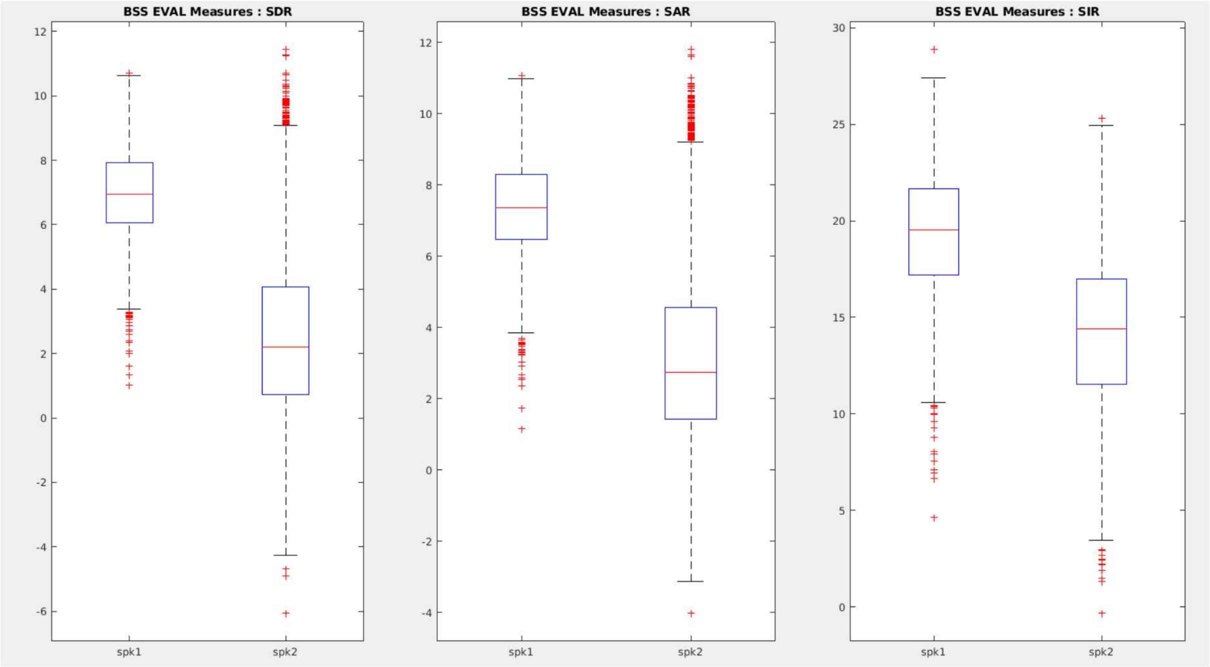

Figure 3.7: Figure shows SDR, SAR and SIR measure on the test data

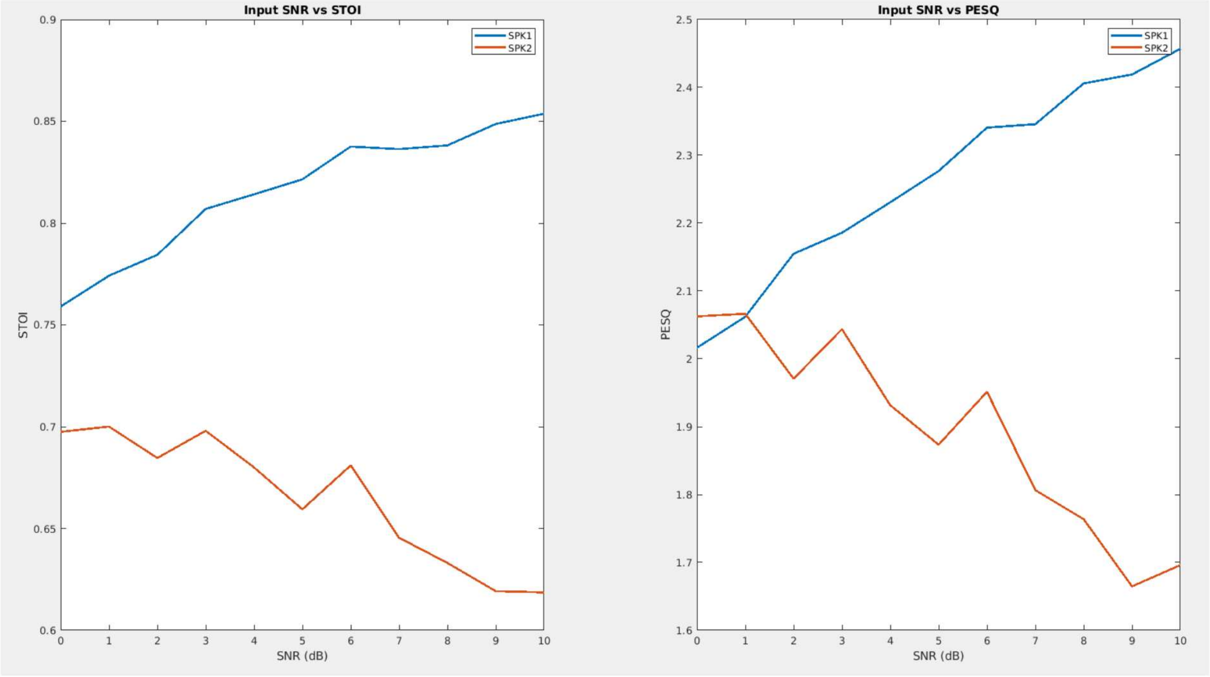

Figure 3.8: Figure shows PESQ and STOI measure on the test data for varying SNR

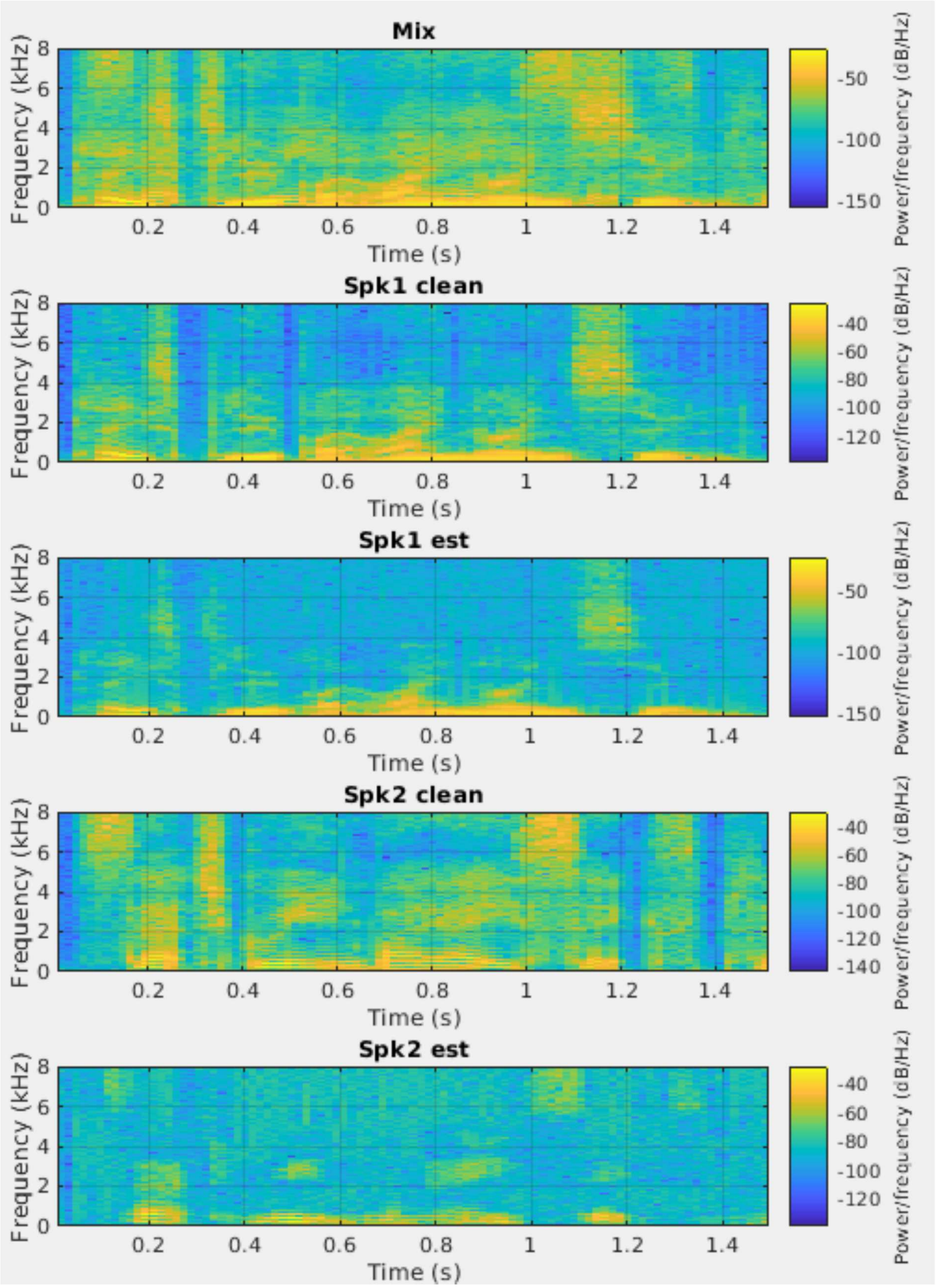

Figure 3.9: Figure shows spectrograms of mixture, speaker 1, speaker 1 estimate, speaker 2 and speaker 2 estimate of a single file used for testing

ASpIRE Chain Model was used to compute the WER's for the dataset. It is a chain model trained on Fisher English that has been augmented with impulse responses and noises to create multi-condition training [11]. The estimated utterance which was computed from the model was forwarded to the speech to text engine running in the cloud, which convert it into text. The resulting transcript is compared to the original text uttered by the speakers. The results are shown in figure 3.10. The table 3.5 shows a 74% improvement in WER for the target speaker.

Table 3.5: WER on GRID corpus dataset in percentages

	Mixture	**Speaker Clean**	**Speaker Estimate**
Speaker 1 (Target)	38.6	0.84	**10.0**
Speaker 2 (Interferer)	65.7	12.4	65.7

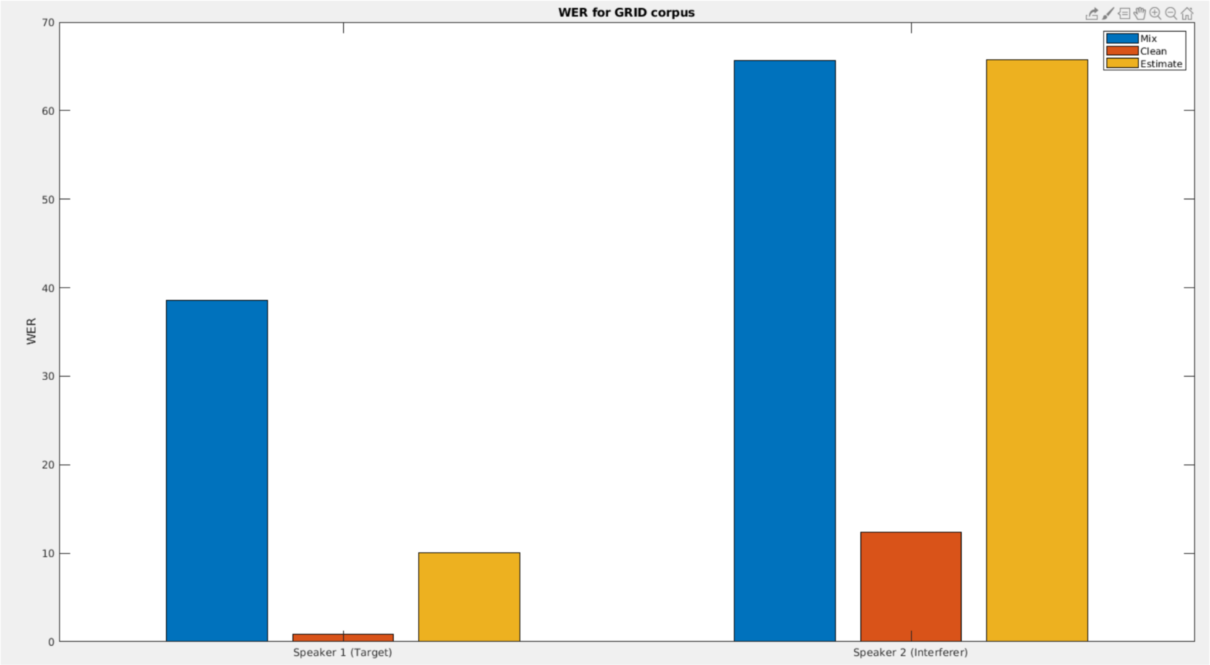

Figure 3.10: Figure shows WER for mix and the estimated speaker utterances using ASpIRE Chain Model

Chapter 4

Summary of Results

The work mainly concentrated on studying the speech enhancement and speaker separation problem from speaker mixture. Speech enhancement results were promising based on the objective measures obtained using simulated speech conditions. On real recording due to the unavailability of degraded data,; the improvement shown were minimal. GRID corpus dataset was used to create the mixture, and the DNN's were trained using multiple datasets and parameters. The speaker separation results showed promising results on GRID corpus datasets, which is a limited vocabulary dataset. The intelligibility score and other objective measures were indicating improved WER's. The WER obtained using the ASpIRE Chain Model showed excellent results for the target speaker. The WER improved by 74 %. The network architecture was able to successfully capture the variation of the speaker in a small vocabulary dataset.

Chapter 5

Future Work

Architectures of DNN's like Convolutional Network and Generative Adversarial Networks need to be explored to get reliable results for real-world data. The training needs to be done on a larger dataset to enable better modeling of the parameters required for the problem. For improving the performance of the system, speaker-specific cues like pitch could be augmented as features during training. The use of i-vectors/ x-vectors can also be incorporated into the dataset for training. Another feature vector like Mel Frequency Cepstral Coefficients (MFCC) can also be added as input features for the model to improve its performance. The single channel based model can be extended to amalgamate multi-channel recordings; which should improve the performance as the network would utilize the spatial information also to localize and separate the speaker.

References

[1] T. Yoshioka and T. Nakatani, “Generalization of multi-channel linear prediction methods for blind mimo impulse response shortening,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.

[2] TCS and DAP Lab IITB, “The tcs meeting corpus,” Aug. 2016. [Online]. Available: Private

[3] M. Cooke, J. Barker, S. Cunningham, and X. Shao, “An audio-visual corpus for speech perception and automatic speech recognition,” *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[4] D. Yu and L. Deng, *Automatic Speech Recognition.* Springer, 2016.

[5] G. Jose, “DISTANT SPEECH RECOGNITION USING MICROPHONE ARRAYS,” Master’s thesis, Indian Institute of Technology Bombay, India, 2017.

[6] I. Cohen, J. Benesty, and S. Gannot, *Speech processing in modern communication: Challenges and perspectives.* Springer Science & Business Media, 2009, vol. 3.

[7] K. Kumatani, T. Arakawa, K. Yamamoto, J. McDonough, B. Raj, R. Singh, and I. Tashev, “Microphone array processing for distant speech recognition: Towards real-world deployment,” in *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific.* IEEE, 2012, pp. 1–10.

[8] T. H. Falk, C. Zheng, and W.-Y. Chan, “A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1766–1774, 2010.

[9] T. Gustafsson, B. D. Rao, and M. Trivedi, “Source localization in reverberant environments: Modeling and statistical analysis,” *IEEE Transactions on Speech and Audio Processing*, 2003.

[10] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[11] D. Povey, "Aspire chain model," 2019(accessed on 12.05.2019). [Online]. Available: http://kaldi-asr.org/models/m1

[12] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[13] A. Cohen, G. Stemmer, S. Ingalsuo, and S. Markovich-Golan, "Combined weighted prediction error and minimum variance distortionless response for dereverberation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on.* IEEE, 2017, pp. 446–450.

[14] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[15] P. Chandna, M. Miron, J. Janer, and E. Gómez, "Monoaural audio source separation using deep convolutional neural networks," in *International conference on latent variable analysis and signal separation.* Springer, 2017, pp. 258–266.

[16] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The musdb18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[17] F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *International Conference on Latent Variable Analysis and Signal Separation.* Springer, 2018, pp. 293–305.

[18] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel music separation with deep neural networks," in *2016 24th European Signal Processing Conference (EUSIPCO).* IEEE, 2016, pp. 1748–1752.

[19] A. Liutkus and F.-R. Stöter, "Music source separation with dnns, making it work," 2019(accessed on 14.02.2019). [Online]. Available: https://sigsep.github.io/tutorials/

[20] Z. Rafii, A. Liutkus, F. R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, "An overview of lead and accompaniment separation in music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1307–1335, Aug 2018.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[23] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude. coursera neural netw," *Mach. Learn*, 2012.

[24] E. Vincent, "Bss eval toolbox version 3.0 for matlab," 2007.

[25] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[26] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2014.

[27] Matlab, "Deep learning toolbox," 2019(accessed on 10.03.2019). [Online]. Available: https://in.mathworks.com/products/deep-learning.html

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2010, pp. 4214–4217.

[30] I.-T. Recommendation, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.