ISSN No:-2456-2165

Speech De-Noising Using Ideal Binary Masking and Deep Neural Networks

Subhrajit Barui
Electronics and Communication
SRM IST
Chennai, India

Barsali Das
Electronics and Communication
SRM IST
Chennai, India

Dr.S Krithiga
Electronics and Communication
SRM IST
Chennai, India

Ishani Saha
Electronics and Communication
SRM IST
Chennai, India

Aditi Samanta
Electronics and Communication
SRM IST
Chennai, India

Abstract:- Noise reduction which is also known as speech enhancement algorithm improves one or more perceptual aspects of noisy speech, most notably, quality and intelligibility. Speech quality is a measure of how clear, natural and free of distortion the speech is. This project deals with two noise reduction algorithms: 1) Binary Masking Algorithm (BMA) 2) Deep Neural Networks (DNN). Binary Masking Algorithm (BMA):This algorithm is used to identify speech dominated and noise dominated units, using which a binary mask is calculated and applied to the noisy input spectrum to get the noise suppressed spectrum. Deep Neural Networks (DNN) containing multiple secluded layers of non-linearity having great potential to capture the complex relationships between noisy and clean pronouncement across various speakers. This project uses a DNN that is operating on the spectral domain of speech signals, and predicts the clean speech spectra when presented with a noisy input spectra.

Keywords:- *Ideal Binary Masking, Deep Neural Network, Noise Quality, Oracle, Intelligibility*

I. INTRODUCTION

In situations where signals from various sources are mixed, source separation may be relevant. Source separation will divide the sound mixture into one or more target sounds and one or more infiltrator sounds, and in some systems, the distinct sources will be further processed. The competence of the human auditory system, to focus upon and follow one particular speaker in such a mixed sound environment, has been termed "the cocktail party phenomenon", and the issue of replicating this capability is called the "cocktail party problem". The cock-tail party phenomenon doesn't create any difficulty for people with normal hearing unless the party takes place in a large room with high reverberation or loud background music. This problem is more evident for hearing impaired listeners.[1]The focus in this project has been to work with two speech enhancement algorithms to mitigate this cock-tail party problem. The two algorithms are Binary Masking Algorithm and Deep Neural Networks. A subjective analysis is performed on different noises with different SNR levels. Depending on those values a performance analysis would be done which can further be used to check for better results and improvements.

II. IDEAL BINARY MASKING(IDBM)

In binary masking, sound sources are assigned as either target or interferer in the time-frequency domain. When different sources are mixed, the identification of parts belonging to the aimed signal and parts belonging to the interferer can be difficult. But using the time-frequency(TF) representations can make this assignment easier because the change from time domain to time-frequency domain can determine which parts of the sound belong to the aimed signal and which parts belong the interferer. The time-frequency representation can reveal characteristics of the sound sources that are invisible in the time domain or frequency domain. Binary masking algorithms can be executed in two steps: Binary Mask estimation and application of the computed mask to carry out the source separation. In the estimation of the binary-mask, the time-frequency parts are assigned to either the target or interferer using, e.g., information about the dimensional location of the sources or speech models. When the binary mask is multiplied to the noisy signal, the TF regions assigned to the aimed source are kept, whereas the regions assigned to the interferer sources are removed or attenuated. To check the feasibility of an estimated binary mask, it is compared to an available reference mask. This reference mask makes it feasible to compare different estimated binary masks and measure their precision, e.g., by estimating the number of errors in the binary mask compared to the reference mask. This reference mask is created under ideal situations. Ideal situations translate into nothing but the beforehand knowledge of the noise in the noisy signal taken as an input. It is also called Oracle Binary Mask.

A. Steps involved in Ideal Binary Masking

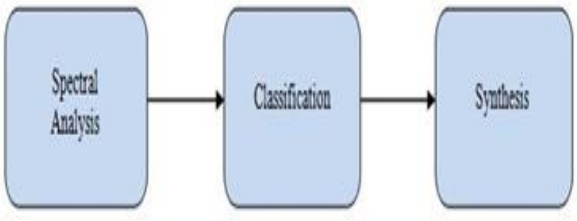

Fig 1:- Block Diagram for Ideal Binary Mask Algorithm

It has three stages: spectral analysis, classification, and synthesis, as we can see in Fig. 1. The spectral analysis step uses the fast Fourier transform (FFT) or a filter bank to trace the original, noisy signal from the time domain to the time-frequency (TF) domain. In the grouping stage, each TF unit is either classified as belonging to class '1' (clean speech, a.k.a. "target"), or class '0' (noise). This designs a binary mask. In the fusion stage or the synthesis stage, the TF-domain form of the original, noisy signal is computed by the binary mask removing all the portions containing the noise of the signal. After the binary mask is multiplied, the TF units are then combined to form a speech signal that is cleaner with a higher SNR. [2]

B. Ideal Binary Masking Flowchart

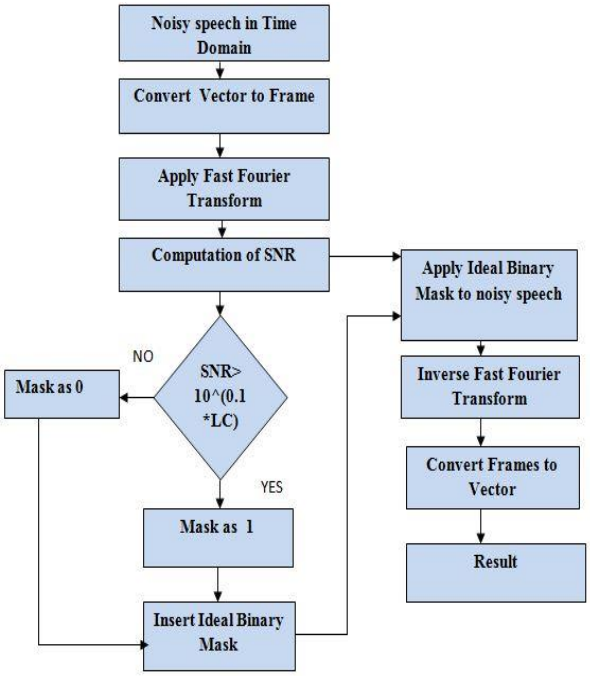

Fig 2:- IDBM Flow-Diagram

➢ *Noisy Speech Creation:*

In this section we have introduced several types of background noises to the clean speech signal. Our main target is to set the SNR of noisy signal as low as possible (ideally zero). We have taken background noises with different SNR values. The formula to measure SNR is as follows

$$SNR_{dB}=20\log_{10}[\text{clean noise/noise}] \quad (1)$$

➢ *Vector To Frame:*

Speech or audio signal contains sound amplitude that varies in time. Vector is a time vector of the noisy signal in which each element correlates to the time of each sample. Vector to Frame conversion splits signal into overlapped (75 percent) frames using indexing and it forms a matrix of input vector. The noisy speech is divided into parts or frames of 32ms with a shift of 4ms in each frame. The matrix created consist rows of sections of length 256, taken at every 32 samples along the input vector and windowed using hamming window.

$$Windowlength = fs * frtime \quad (2)$$

Where fs = 8000Hz (Sample Frequency), fr(time) = 32ms (Frame time) Window length = 256

$$W(n)= 0.5[1-\cos((2/N)] \quad 0 <= n <= N \quad (3)$$

Where *W(n)* is a function to determine the hanning window coefficients. The hanning window length is L=N+1.

The noisy signal is manifold with a hanning window function of fixed length form of the noisy signal. Then we check if padding is needed but for exact division we have to zero padding.

➢ *Fast Fourier Transform:*

Fast Fourier Transform is for transformation of signal from time to frequency domain. The speech is first windowed for FFT analysis. FFT process are placed one after another and processed frames are placed systematically to form output signal.

$$X(k)=\sum_{n=0}^{N+1} x(n)W(n)_N^{kn} \quad k=0,1...N \quad (4)$$

Where *X(k)* is the FFT output, *x(n)* is the time domain signal and *W(n)* is the hanning window function. In MATLAB we can use the *Y = fft(X)* function to perform the FFT on the noisy signal.

➢ *Binary Masking:*

Ideal Binary Mask(IBM) This identifies the speech dominated and the noise dominated units, a binary mask is measured and applied to the noisy input spectrum. The ideal binary mask is calculated. From an oracle SNR by thresholding with local SNR benchmark stated by LC. The units with a local SNR higher than the threshold are determined as target-dominated units, while others are defined as masker-dominated units. Speech dominated unit is labeled as 1 and noise dominated units are labeled as 0. The binary mask is then multiplied to the noisy signal to get the enhanced signal.

$$MASK = '1' \quad SNR>10^{(0.1*LC)} \quad (5)$$

$$MASK = '0' SNR<10^{(0.1*LC} \quad (6)$$

$$Y(k) = X(k) * \text{MASK} * \exp(j2\pi * X(k)) \quad (7)$$

Where X(k) is the input noisy FFT signal and Y(k) is the output masked signal.

➢ *Inverse Fast Fourier Transform*:

In this process we convert the enhanced signal from frequency to time domain. The formula for IFFT is as follows

$$Y(n)=(1/N)\sum_{n=0}^{N+1} Y(k)W_N^{nk} \quad \text{n=0,1,2....N+1} \quad (8)$$

Identify applicable funding agency here. If none, delete this text box.

ISSN No:-2456-2165

Where *Y(n)* is the enhanced signal in time domain and is the enhanced speech in frequency domain. We can use *Y = ifft(x)* in MATLAB here.

➢ *Frame To Vector Conversion:*

The frame to vector conversion is crucial to convert frames of signal into vector form. After this process we get continuous enhanced signal at the output. The algorithm use weighted overlap-and-add synthesis method for conversion and uses synthesis filter. Overlap-and-add method segments the input into block of length L. If e take M as the length of impulse response then M-1 zeros, the last M-1 points from each output portion must be overlapped-and-added to first M-1 points of next block. Hence this method is called as overlap-and-add method. At the end of conversion, overlap-and-add method combines the entire blocks to obtain the continuous output signal.

$$y(n) = (\sum_{k} x_k[n - kL]) * h(n) \quad (9)$$

Where the *y(n)* is the continuous time-domain signal and is the impulse response of the linear filter. The advantage of high computation is directly associated with the circular convolution.

$$y_k(n) = IFFT(FFT(x_k(n)) * FFT(h(n))) \quad (10)$$

So the final expression is

$$y(n) = \sum_{k} y_k[n - kL] \quad (11)$$

III. DEEP NEURAL NETWORKS

With the development of deep learning, neural networks were used in many forms of speech recognition such as phoneme categorization, remote word recognition, audio visual speech recognition, audio-visual speaker recognition and speaker adjustment. Deep learning enabled the evolution of Automatic Speech Recognition (ASR) systems. These ASR systems need separate models, namely acoustic- model (AM), a pronunciation-model (PM) and a language-model (LM). There are different types of neural networks.

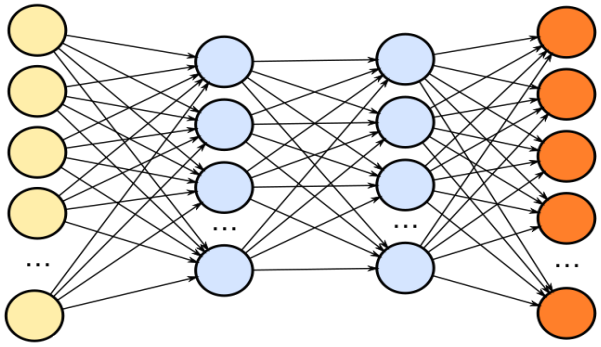

Fig 3:- Fully Connected Network

Different types of neural networks use different convention in defining their own rules. Adaptive Linear Neuron Network (ADALINE) is designed as the single layer neural network. It is established on the principle of Multilayer Perceptron (MLP).[4]There have been many advancements in this area and multi-layer neural networks have come into existence. Some of them are Artificial Neural Networks(ANN), Recurrent Neural Networks(RNN), Fully-Connected Neural Networks(FNN), Convolutional Neural Networks(CNN) etc.[3] In this work we have used Fully-Connected Neural Network for speech enhancement. A fully-connected neural network consists of an array of fully-connected layers. Each output magnitude depends on each input magnitude .Each neuron in a fully-connected layer is connected to all activation from the previous layer.[4]

A. Steps Involved in Deep Neural Network

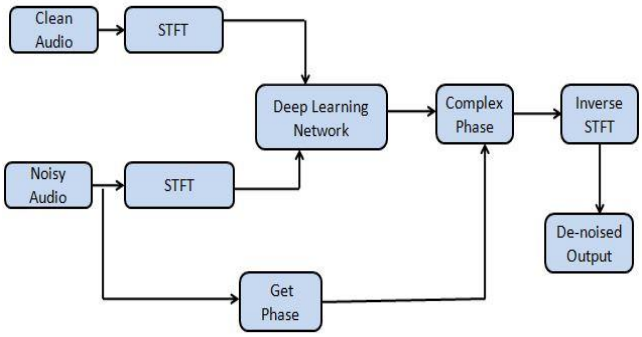

Figure 4:- Block Diagram for Neural Networks

The basic deep learning training scheme is displayed in Fig 4. To decrease the estimation load of the structure first down-sampling is done on the clean and noisy audio signals to 8KHz. The predictor and target network signal are the magnitude-spectrum of the noisy and clean signals respectively. The networks output is the magnitude-spectrum of the de-noised signal. The regression network uses the predicted input to minimize the mean square error between the input and the output target. The output magnitude-spectrum and the phase of the noisy signal are used to reform from the de-noised audio to the time-domain.

ISSN No:-2456-2165

B. Flow Diagram of Deep Neural Network

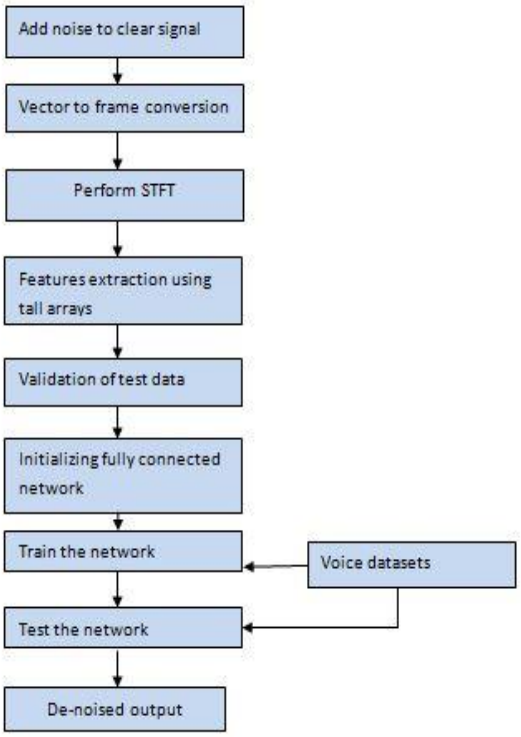

Fig 5:- Flow Chart of Deep Neural Network

The methodology explains how to de-noise speech signals. The aim of speech de-noising is to discard noise from the speech signal. While minimizing undesired artifacts in output speech. The process starts with examining the data sets. This data sets are used to train and test the structure. The deep learning networks need to be trained on a subset of thousand files per turn.

$SNR_{dB}=20log_{10}[clean\ noise/noise]$ (12)

➢ *Downsampling :*

The signal needs to be downsampled to decrease the estimation load. The predictor and target network signal are the magnitude-spectrum of noisy and clean audio signals respectively. The networks output is the magnitude spectrum of the de-noised signal.

$$Y(e^{j\omega}) = {}^{1}/_{D} \sum_{k=0}^{D-1} X(e^{\frac{j(\omega-2\pi k)}{D}}) \quad (13)$$

$X(e^{jw})$ is the spectrum of the input signal. $Y(e^{jw})$ is the spectrum of the output signal. D is the sampling rate reduction factor.

➢ *Computation of STFT:*

To start the training program the initial step is computation of stft of predictor and noisy signals. The final dimension of both variables correlates to the number of discrete predictor or aim pairs created by the audio file. Each predictor is 129/8 and each aim signal is 129/1.

$$STFT\{x(t)\}(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)\omega(t-\tau)e^{-j\omega t}\,dt \quad (14)$$

where $x(t)$ the signal to be transformed $w(t)$ is the window function $X(\tau,\omega)$ is essentially the Fourier-Transform of $x(t)w(t-\tau)$ a complex function representing the phase and magnitude of the signal over time and frequency.

➢ *Fully connected:*

De-noising network comprised of fully-connected layers where each neuron is connected to all activation from the previous layers as we can see in the figure. In a fully connected layer the input is multiplied by a weight matrix and then a bias vector is added. Determination of the dimensions of the weight matrix and bias vector are done by the number of neurons in the layer and the number of activations from the previous layer. Each time index m, the output signal $y(m)$ and the error $a(m)$ can be calculated as

$$y(m) = w(m) * x(m) + b(m) \quad (15)$$
$$a(m) = t(m) - y(m) \quad (16)$$

where t(m) is the target signal or the clean signal x(m) is the input signal or the noisy signal b(m) is the bias parameter. w(m) is the weight parameter. The weights and biases of the network are adapted as

$$w(m)_{new} = w(m) + la(m) * x(m) \quad (17)$$
$$b(m)_{new} = b(m) + la(m) \quad (18)$$

➢ *Activation Energy:*

It gives a \say{"weighted sum"} of the input, then a bias is added and then it decides whether it should be \say{"fired"} or not. Here we are using the ReLU(Also known as Rectifier Neural Networks) function as the activation-function. The graphical representation of the ReLU function is represented in Figure 6.

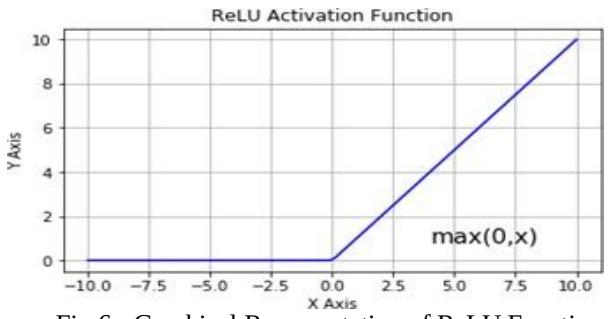

Fig 6:- Graphical Representation of ReLU Function

The mathematical Equation is

$F(x)=max(0,x)$ (19)

➢ *Training :*

Training options are specified for the network. Epochs(cycle) size is set to three so that the structure makes three passes through the training data. The dimensions are set to to 128 so that the structure looks at 128 training signals at a time. The plots are specific as "training-progress" to create plots that show the training progress with the increasing number of iterations. The training arrays are

made to shuffle at the initial stage of each epoch. The L factor or the learning rate factor in this case is set to 0.9. Validation Data is set to the validation predictors and targets. Validation Frequency is set to such that the validation mean square error is computed once per epoch. The Adaptive Moment Estimation (ADAM) solver is used by this training phase.

➢ *Testing:*

To test the performance of the trained network, speech signals from the \say{test} folder are used . \say{"AudioDatastore"} is used to generate a data store for the files in the "test" folder. The files in the data store are shuffled. Then the matter of the file are gathered from the data store. Audio signal is converted to 8 kHz. A random noise segment is created from the noise vector. Noise is added to the speech signal such that the SNR is 0 dB. STFT of the noisy audio is performed. From the noisy STFT,- segment training predictor signals are generated. The consecutive predictors have an overlap of 7 segments. Predictors which are normalized by mean and standard deviation is calculated in the training stage . The de-noised speech signals are computed by using Istft() function which performs the inverse STFT. The phase of the noisy STFT vectors is used to reconstruct the time-domain signal.

IV. RESULTS

A. *Ideal Binary Masking*

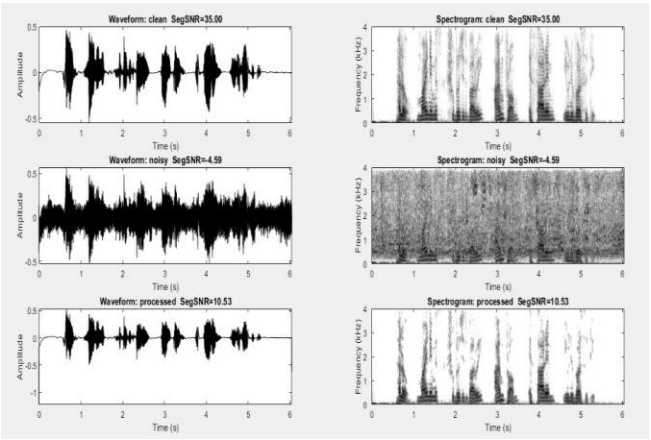

Fig 7:- Spectrogram Representation of Babble Noise

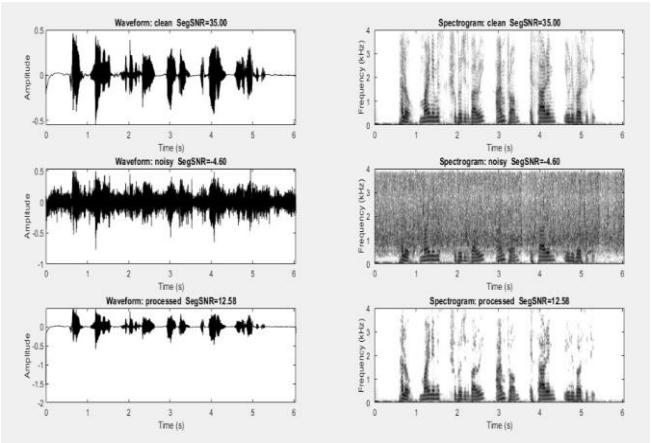

Fig 8:- Spectrogram Representation of Ceiling Noise

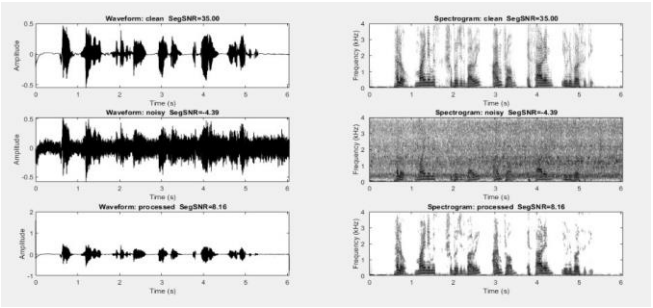

Fig 9:- Spectrogram Representation of Washing Machine Noise

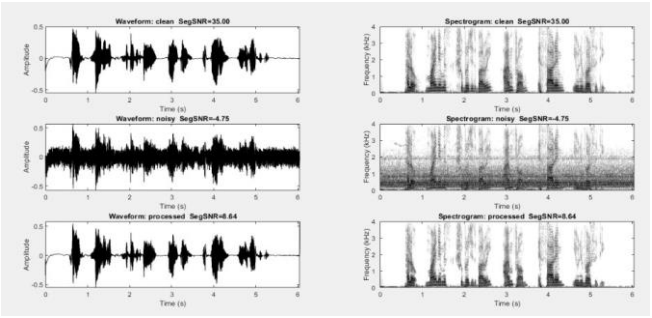

Fig 10:- Spectrogram Representation of Tap Water Noise

Noise Types	***Clean SNR(dB)***	***Noisy SNR(dB)***	***Processed SNR(dB)***
Washing Machine	35	-4.39	8.16
Tap Water	35	-4.74	8.64
Babble	35	-4.59	10.53
Ceiling	35	-4.6	12.68

Table 1

The above table shows the comparison of different types of noises and their SNR is compared. From the above table it can be inferred that Ceiling fan processed signal is having the highest value of SNR 12.68 and hence the noise is reduced to the maximum extent in this type of signal whereas the lowest processed signal is Ceiling fan noise which is having SNR of 8.16, so here the noise is least reduced. While for the other types of processed signals are between the maximum and minimum SNR values of the nine types of signals ,hence noise is reduced moderately.

B. *Deep Neural Networks*

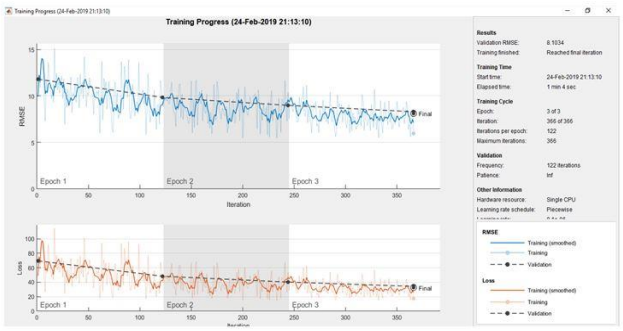

Fig 11:- Graphical Representation of Training

The dataset is too large hence it took several minutes to complete the process. The number of weights in the connected network are 2237440. The Root Mean Squared Error(RMSE) value is sufficiently low. Since the number of datasets are increasing it is seen that there is successive improvement in the RMSE value. But this increase the computational load and computational time of the process. The loss in the processed signal is lower than 40dB and this value can also be improved by increasing number of data-sets.

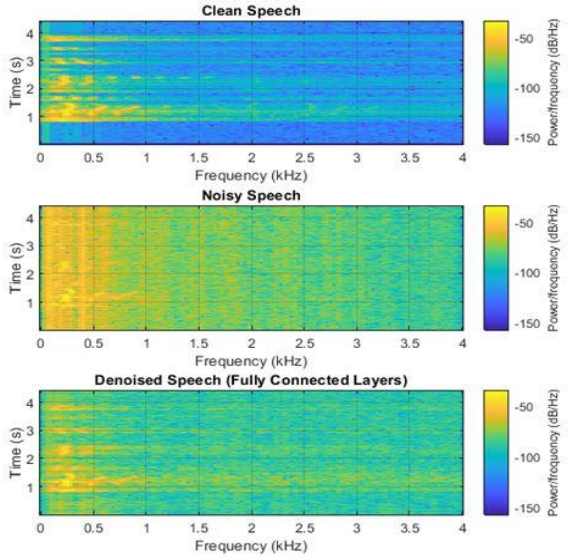

Fig 12:- Spectrogram Representation of DNN Output

For testing purpose testing data-set is available. But sample input signal will be used to analyze the testing process. This is because our main target is to observe enhancement in the quality.Next the testing is done using the data-set to check if the network is computationally efficient. As from the plot Figure:11 it is observed that the nature of the de-noised speech is quite good and there can be some minor noise signals are in the background. From the spectrogram analysis also Figure:12 it is observed that there is very minimum loss due to the noise signal.

V. CONCLUSION

Speech enhancement method by using Ideal Binary Mask is performed on different speech to enhancement the nature as well as intelligibility of speech. Performance is assessed on the grounds of speech quality and intelligibility. SNR of clean, noisy and enhanced speech is measured to determine the nature of processed speech. The proposed method eliminates the additive noise present in speech signal and restores the speech signal to its original form. Speech enhancement method in proposed method attenuates the additive background noise without adding speech distortion.

DNN is used for different hidden layers that can establish the effectiveness of speech enhancement in the area of data mining. Though the time consumption is more in DNN, speech enhancement is better. The de-noised output is obtained and the background noises are reduced to the maximum extent.

REFERENCES

[1]. Philipos C. Loizou and Gibak Kim. ``Reasons why Current Speech-Enhancement Algorithms do not Improve Speech Intelligibility and Suggested Solutions" \textit{ IEEE Transactions on Audio, Speech, and Language Processing}, vol. 19, No. 1, January 2011.

[2]. HANSON, K. ODAME, ``REAL-TIME EMBEDDED IMPLEMENTATION OF THE BINARY MASK ALGORITHM FOR HEARING PROSTHETICS", \TEXTIT{IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS}, VOL. 8, NO. 4, AUGUST 2014.

[3]. Ding Liu , Paris Smaragdis,Minje Kim ``Experiments on Deep Learning for Speech Denoising" \textit{ University of Illinois at Urbana-Champaign, USA Adobe Research, USA Interspeech} 2014

[4]. Se Rim Park,Jin Won Lee ``A Fully Convolutional Neural Network for Speech Enhancement`` \textit{ Carnegie Mellon University, USA Qualcomm Research, USA Interspeech 2017} August 20–24, 2017

[5]. Morten Kolbaek, Zheng-Hua Tan (2017) ``Speech Intelligibility Potential of General and Specialized Deep Neural Network based Speech Enhancement Systems" \textit{ IEEE Transactions on audio, speech and Language processing }, vol. 25, No. 4, November 2015.