A Study on Effectiveness of Deep Neural Networks for Speech Signal Enhancement in Comparison with Wiener Filtering Technique

Vijay Kumar Padarti, Gnana Sai Polavarapu, Madhurima Madiraju, V. V. Naga Sai Nuthalapati, Vinay Babu Thota, and V. D. Subramanyam Veeravalli

1 Introduction

The five fundamental senses, i.e., hearing, sight, smell, taste, and touch, perceive the information from the environment, and the human brain processes this information to create a precise response. Sound acts as an information provider to these senses. The information that is transmitted has to be free of noises to get a better understanding of the external environment. Noise can be described as any unwanted information which hinders the ability of the human body to process the valuable sensory information. Hence, an uncorrupted sound becomes essential for proper interaction of humans with their external world. The primary focus is on speech signals which are information providers in various communication systems. During the transfer of signals, distortion by some unwanted signals causes loss of useful data and information stored in the signals. There are many real-world noise signals such as the noise of a mixer grinder, washing machine, and vehicles which have to be reduced to retrieve the wanted information. The frequency of speech signals ranges from 85 to 255 Hz. Typical male voice ranges in between 85 and 180 Hz, whereas the female voice ranges in between 165 and 255 Hz. Babies have even higher ranges of frequency reaching up to 1000Hz in a few cases [1].

Speech denoising refers to the removal of background content from speech signals. The goal of speech denoising is to produce noise-free speech signals from noisy recordings while improving the perceived quality of the speech component and increasing its intelligibility [2]. Speech denoising can be utilized in various applications where we experience the presence of background noise in communica-

V. K. Padarti (✉) · G. S. Polavarapu · M. Madiraju · V. V. Naga Sai Nuthalapati · V. B. Thota · V. D. Subramanyam Veeravalli
Department of Electronics and Communications Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India
e-mail: vijayakumar.padarti@vrsiddhartha.ac.in

A. Biswas et al. (eds.), *Advances in Speech and Music Technology*, Signals and Communication Technology, https://doi.org/10.1007/978-3-031-18444-4_6

tions, e.g., hearing aids, telecommunications, speech recognition applications, etc. [3].

A number of techniques have been proposed based on different assumptions on the signal and noise characteristics in the past, but in this chapter, we shall compare two main methods, Wiener filtering technique and neural network method. For neural network technique, we will consider two types of networks, fully connected network and convolutional neural network. We compute PSNR and SNR values for these three techniques to compare the denoised signal quality.

2 Background

2.1 Wiener Filtering Technique

One of the notable techniques of filtering that is widely used in signal enhancement methods is Wiener filtering. The key principle of Wiener filtering, essentially, is to take a noisy signal and acquire an estimate of clean signal from it. The approximate clean signal is acquired by reducing the mean square error (MSE) between the estimated signal and desired clean signal [4].

The transfer function of the Wiener system in frequency domain is

$$H(w) \ = \frac{P_s(w)}{P_s(w) + P_v(w)}. \tag{1}$$

where

$$P_s(w) = power\ spectral\ density\ of\ clean\ signal. \tag{2}$$

$$P_v(w) = power\ spectral\ density\ of\ noise\ signal. \tag{3}$$

Here, the signal s and noise v are considered to be uncorrelated and stationary.

The signal-to-noise ratio (SNR), which is used to detect the quality of a signal, is defined as

$$SNR \ = \frac{P_s(w)}{P_v(w)}. \tag{2}$$

Substituting SNR in the above transfer function, we obtain

$$H(w) \ = \left(1 + \frac{1}{\mathrm{SNR}}\right)^{-1}. \tag{3}$$

One of the popular applications of the Wiener filtering technique is the Global Positioning System (GPS) and inertial navigation system. Wiener filter, which is also used in geodesy to denoise gravity records, is used in GPS to model only those time variabilities that are significant when adapted to noise level of data [5].

Signal coding applications is a field where Wiener filter is widely used in. In signal processing and broadly engineering applications too, Wiener filter is considered to be a great tool for speech applications due to its accurate estimation characteristic. This filter can further be adapted to serve different purposes like satellite telephone communication [6].

If we dive into the world of electronics and communication more, the Wiener filter has a range of applications in signal processing, image processing, digital communication, etc. like system identification, deconvolution, noise reduction, and signal detection [7].

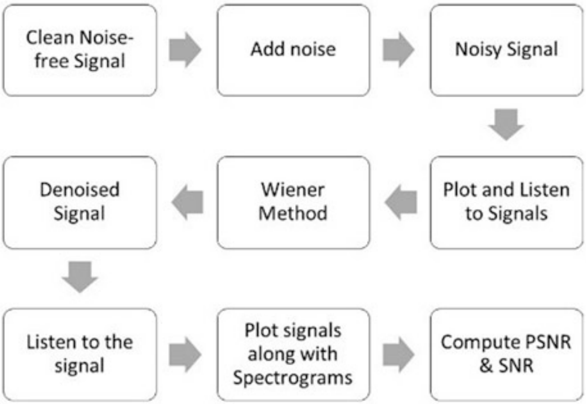
Specifically in image processing, Wiener filter is a quite popular technique used for deblurring, attributed to its least-mean-squares technique. The blurriness in images that is caused as a result of motion or unfocused lens is removed using this filter. Additionally, since it returns mathematically and theoretically the best results, it also has applications in other engineering fields [8].

2.1.1 Algorithm

To denoise a speech signal using the Wiener filtering technique, we first fetch a clean audio signal file and a noise signal file from the audio datastore in MATLAB. We then extract a segment from the noisy signal and add it to the clean signal to make it a noisy speech signal which is given as input to the Wiener filter. The Wiener filter performs denoising of the speech signal, and then we visualize the output signal. Flowchart for the algorithm can be seen in Fig. 1.

In order to compute peak SNR and SNR values, the output and input signals are given to PSNR function which is a built-in function in MATLAB. The frequency response of the Wiener filter is such that, at frequencies where SNR is low, that is, noise power is high, the gain of the filter decreases, and the output is limited, causing

Fig. 1 Wiener method flowchart



noise reduction. Correspondingly, for high SNR, that is, when signal power is high, the gain becomes nearly one (∼1), and output sought is very close to input. Another drawback is that at all given frequencies, the Wiener filter requires a fixed frequency response. One more shortcoming in the Wiener filter is that before filtering, the power spectral density of both clean and noise signals has to be estimated. Noise amplification is also a problem [6, 9–11].

2.2 Deep Neural Networks

Deep learning is part of machine learning with an algorithm inspired by the structure and function of the brain, which is called an artificial neural network. Artificial neural networks are the statistical model inspired by the functioning of human brain cells called neurons. Deep learning is used in many fields such as computer vision, speech recognition, natural language processing, etc. [12].

A neural network mimics the human brain and consists of artificial neurons, also known as nodes. Group of nodes make a layer. There are three types of layers: the input layer, the hidden layer(s), and the output layer. There can be multiple hidden layers and it depends on the model. All the nodes are provided with information in the form of input. At each node, the inputs are multiplied with some random weights and are computed, and then a bias value is added to it. Finally, activation functions, such as rectified linear unit (ReLU) function, are applied to determine which neuron to eliminate.

While deep learning algorithms feature self-learning representations, they depend upon neural networks that mimic the way the brain processes the information. During the training process, algorithms use random unknown elements in the input to extract features, segregate objects, and find useful data patterns. Much like training machines for self-learning, this occurs at multiple levels, using the algorithms to build the models. Deep learning models utilize several algorithms. Although none of the networks is considered perfect, some algorithms are preferred to perform specific tasks. Some commonly used artificial neural networks are feedforward neural network, convolutional neural network, recurrent neural network, and autoencoders [13].

There are also some disadvantages of deep learning. Very large amount of time is required to execute a deep learning model. Depending upon the complexity, sometimes, it may take several days to execute one model. Also, for small datasets, the deep learning model is not suitable. There are various applications of deep learning such as computer vision, natural language processing and pattern recognition, image recognition and processing, machine translation, sentiment analysis, question answering system, object classification and detection, automatic handwriting generation, automatic text generation, etc.

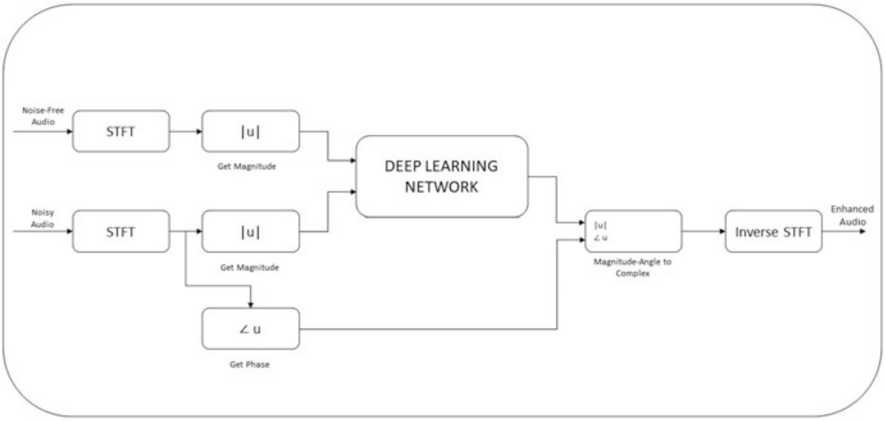


Fig. 2 Neural network block diagram

2.2.1 Algorithm

We first fetch clean and noisy audio files from the audio datastore in MATLAB, and then we extract a segment from the noisy audio and add it to the clean audio signal. This will be the input given to the deep learning network. Neural network block diagram is shown in Fig. 2.

We utilize short-time Fourier transform (STFT) to transform the audio signals from time domain to frequency domain. The magnitudes are extracted and then fed to the neural network. Then the output signal which is the denoised and enhanced version of the input noisy signal is converted back into the time domain using the inverse STFT.

An exemplary speech signal is shown in Fig. 3. Clearly, it can be seen that the amplitudes vary significantly with time, i.e., there will be huge variations frequently in the signals like music and speech. This is the reason we utilize the short-time Fourier transform technique.

We have utilized two models of deep learning networks: fully connected and convolutional neural network. For any model, the network first needs to be trained so that it learns its function to segregate the noise segments from the audio segments. For training the model, we consider a sample signal and then set the required parameters such as learning rate, number of epochs, batch size, etc. Once the model completes its training, it has to be tested. For the testing phase, we feed the model with another set of samples which were not given in the training phase and observe the outputs. The flowchart for the artificial neural network is depicted in Fig. 4.

To compare the efficiency of the two models, we compute PSNR and SNR values using psnr function which is a built-in function in MATLAB. We also use another in-built function, sound(), to listen to the audio signals. Besides this, we also represent the signals with timing plots and spectrogram.

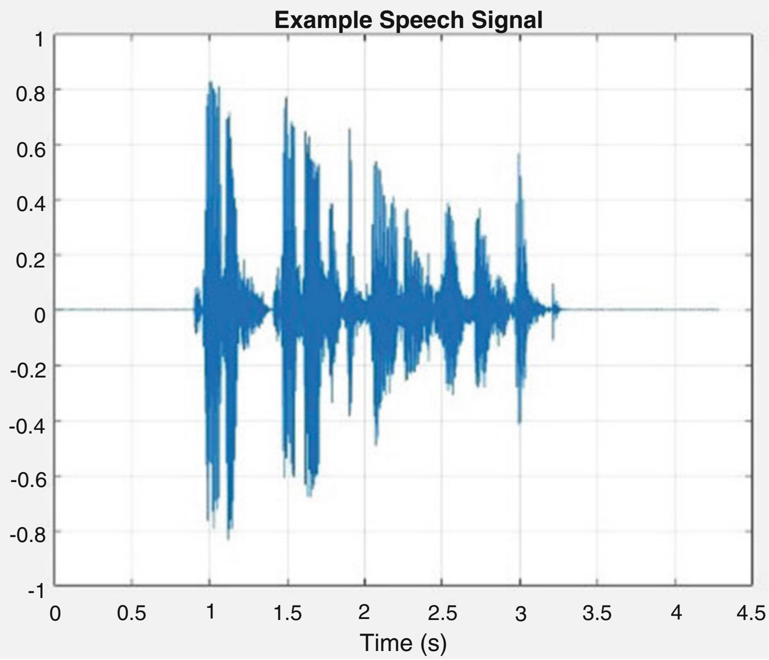


Fig. 3 Sample audio signal

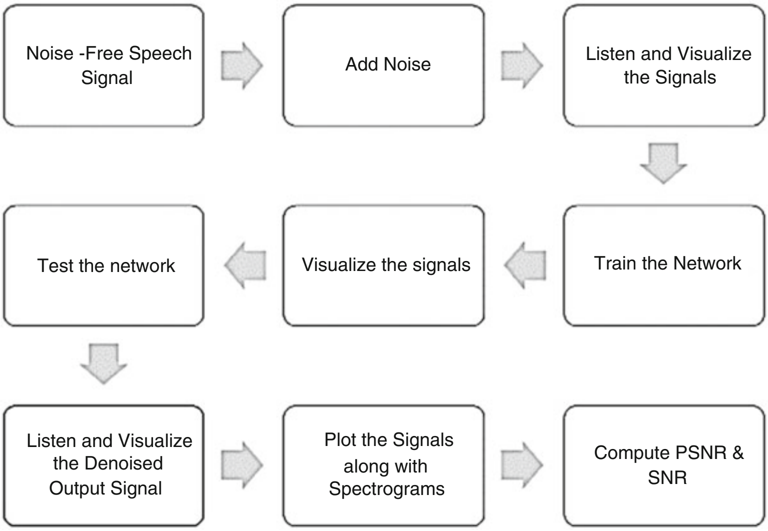

Noise -Free Speech Signal
Add Noise
Listen and Visualize the Signals
Train the Network
Visualize the signals
Test the network
Listen and Visualize the Denoised Output Signal
Plot the Signals along with Spectrograms
Compute PSNR & SNR

Fig. 4 Neural network flowchart

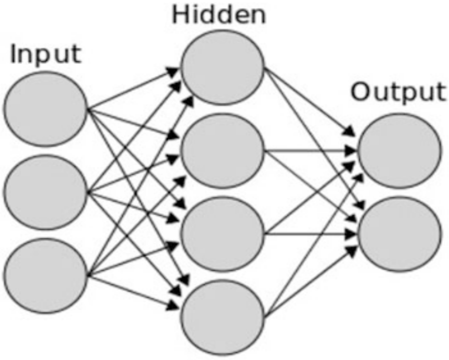


Fig. 5 Fully connected network

2.2.2 Fully Connected Network

A fully connected neural network consists of a series of fully connected layers that connect every neuron in one layer to every neuron in the next layer. For any network, there are three types of layers: input, hidden, and output layers. The information received from the input is given to the model, and then the model is trained using this data [14]. A fully connected network model is shown in Fig. 5.

We define the number of hidden layers in the model. For our model, we have 2 hidden layers with 1024 neurons each. The model is trained on the training dataset. Each of the hidden layers is followed by ReLU layers and batch normalization layers.

A clean audio file fetched from the audio datastore is corrupted with a noisy segment extracted from the noise signal. These signals are plotted in Fig. 6. Then these signals are passed to the network model, and the model is trained. The training process involves learning the model function by passing the model through the given dataset for 3 epochs (in order to avoid overfitting, we have limited to 3 epochs) with a batch size of 128 at an initial learn rate of 10^{-5}, and for every epoch, the learning rate decreases by a factor of 0.9.

2.2.3 Convolutional Neural Network (CNN)

Convolutional neural networks can be differentiated from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers: convolutional layer, pooling layer, and fully connected layer.

The first layer of a convolutional network is the convolutional layer. After the convolutional layer, we can have the additional convolutional layers or the pooling layers. The final layer is the fully connected layer. The CNN complexity increases with each layer, but the model gets more accurate outputs. Therefore, there should be some optimality.

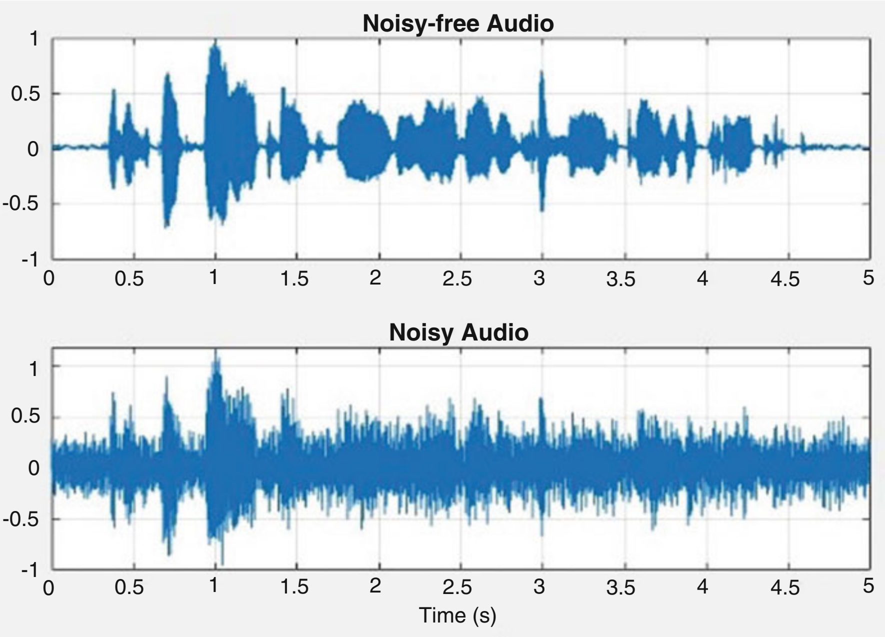


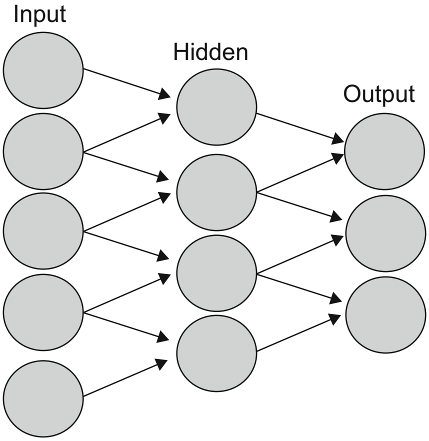
Fig. 6 Clean and noisy audio

The convolutional layer is the core building block of a CNN. It is where the major part of the computation occurs. It requires a few components: input data, a filter, and a feature map. Pooling layers, also known as downsampling, conduct dimensionality reduction, reducing the number of parameters in the input. It is similar to the convolutional layer in processing and filtering the input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. A basic convolutional layer model is shown in Fig. 7.

There are two main types of pooling: max pooling which selects the maximum value and average pooling which computes the average value of data. A lot of information is lost in the pooling layer, but it also has a number of advantages to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

In the fully connected layer, each node in the output layer connects directly to a node in the previous layer. This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLU functions, FC layers usually use a SoftMax activation function to classify inputs appropriately, producing a probability from 0 to 1 [15].

Fig. 7 Convolutional layer



For our convolutional network model, we have defined 16 layers. Similar to the fully connected network, convolutional layers are followed by ReLU and batch normalization layers.

3 Results

3.1 Wiener Filtering Technique

The Wiener model is provided with a clean and a noisy signal, and it yielded the output as shown in Fig. 8. We can clearly see that a lot of high-frequency components in the original audio signal are lost when denoised. This leads to reduced quality. Also, the model resulted in a PSNR 20.0589 dB and SNR of 2.4825 dB which are considered to be low compared to standards. Results for different samples are shown in Table 1.

3.2 Fully Connected Network

3.2.1 Training Stage

In the training stage, our model is provided with training dataset and is made to learn its function. The training progress is visualized in Fig. 9, where we can see the internal approximation process.

After the training is complete, the time and frequency plots are visualized as in Fig. 10. We can see that the denoised version is almost approximately equal to the original clean signal.

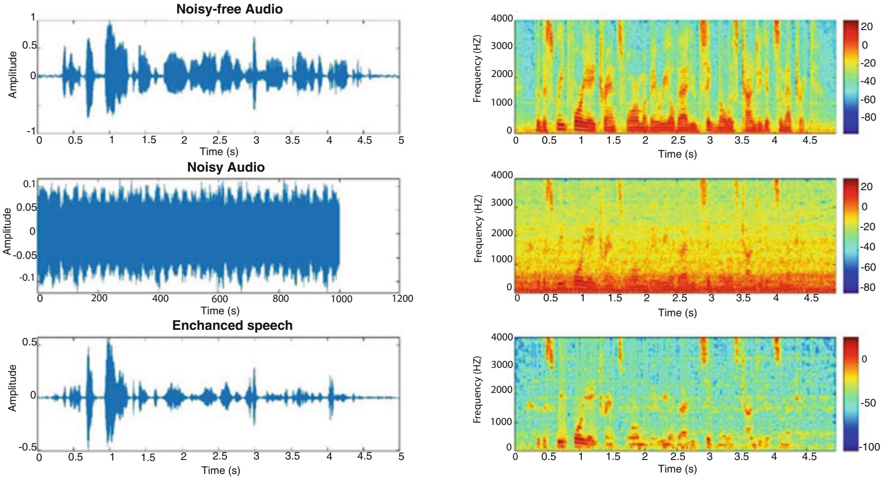


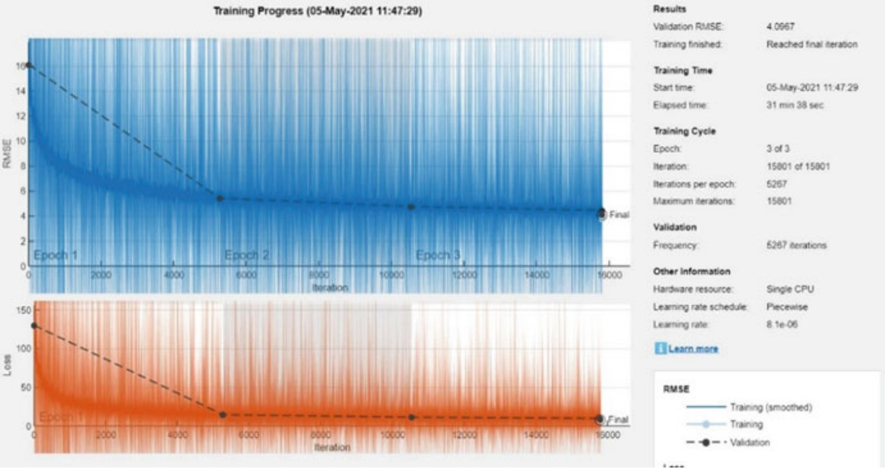
Fig. 8 Wiener method results

Table 1 Wiener method results

Sample	PSNR [dB]	SNR [dB]
1	20.0589	2.4825
2	20.7663	3.1900
3	20.4732	2.8969
4	20.8320	3.2557
5	21.2536	3.6772
6	20.6340	3.0577
7	20.2921	2.7157
8	20.6636	3.0872

3.2.2 Testing Stage

Once the model is trained, we test the model with a dataset which is not given in the training stage. When our fully connected model is tested, it resulted in a PSNR of 23.5416 dB and an SNR of 6.5651 dB, which are greater than those of the Wiener method, but yet they are lower than the acceptable standard values. The output time and spectrogram plots are visualized as in Fig. 11. The original and enhanced versions are nearly equal but not exactly equal, but when compared to the Wiener filtering technique, it can be seen that the fully connected network method yielded better results. Results for different samples are shown in Table 2.



Results

Validation RMSE: 4.0967

Training finished: Reached final iteration

Training Time

Start time: 05-May-2021 11:47:29

Elapsed time: 31 min 38 sec

Training Cycle

Epoch: 3 of 3

Iteration: 15801 of 15801

Iterations per epoch: 5267

Maximum iterations: 15801

Validation

Frequency: 5267 iterations

Other Information

Hardware resource: Single CPU

Learning rate schedule: Piecewise

Learning rate: 8.1e-06

RMSE

Training (smoothed)

Training

Validation

Fig. 9 Fully connected network training progress

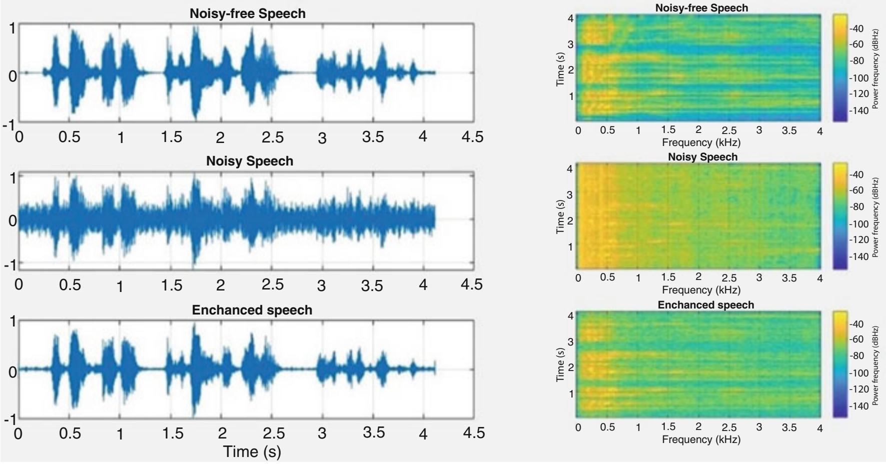


Fig. 10 Fully connected network result in the training stage

3.3 *Convolutional Neural Network*

3.3.1 Training Stage

Similar to that of the fully connected model, our convolutional model is first trained with a dataset. The training progress window is visualized in Fig. 12. It should be noted that the training process took longer time compared to fully connected model due to the number of layers and the complexity of the layers. Once training is completed, the time and frequency plots are visualized as in Fig. 13.

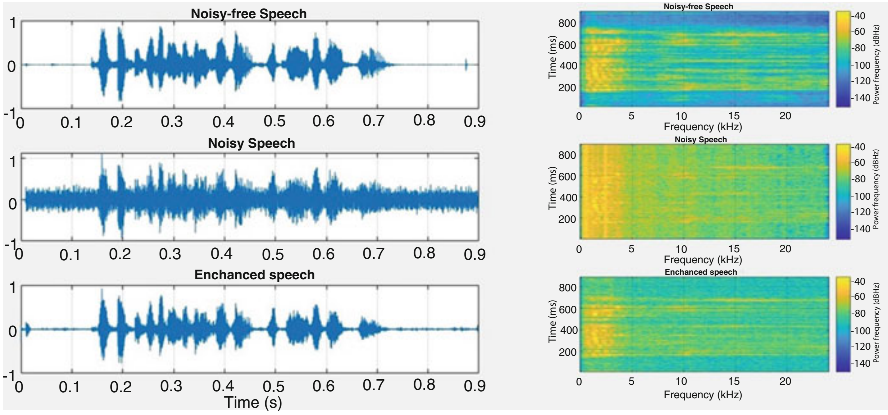


Fig. 11 Fully connected network result in the testing stage

Table 2 FC model results

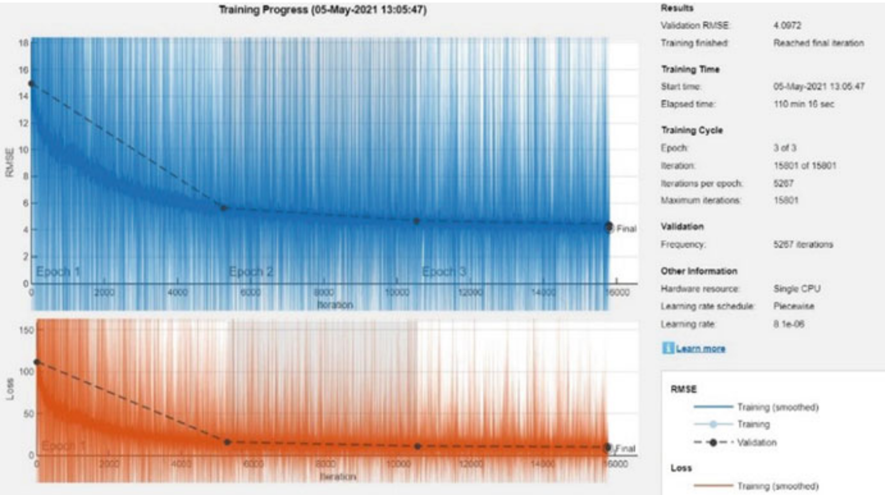
Sample	PSNR [dB]	SNR [dB]
1	23.5416	6.5651
2	24.1829	7.2064
3	24.0188	7.0423
4	26.5571	7.7258
5	23.7959	6.8194
6	23.8697	6.8932
7	23.9390	6.9625
8	23.8521	6.8756

3.3.2 Testing Stage

After successfully training the model, we test the model with a testing dataset. The testing dataset is a dataset which is not provided in the training dataset. This helps in the evaluation of the model.

In the testing stage, the time and frequency plots are visualized as in Fig. 14 where the original clean audio signal, the corrupted noisy signal, and the enhanced speech signals are plotted. It can be clearly seen that our model effectively denoises the noisy signal and enhances its quality. When compared to the previous technique, i.e., Wiener filtering technique, where we have removed all the high-frequency components, we have removed only the noisy components and retained the original high-frequency components in the output signal. So, this model is effective compared to the Wiener method.

The convolutional model resulted in a SNR of 7.6137 dB and a PSNR of 26.4451 dB. When compared to Wiener and fully connected models, these values are higher. However, they are still lower than the acceptable standard values. Results for different samples are shown in Table 3.



Results

Validation RMSE: 4.0972

Training finished: Reached final iteration

Training Time

Start time: 05-May-2021 13:05:47

Elapsed time: 110 min 16 sec

Training Cycle

Epoch: 3 of 3

Iteration: 15801 of 15801

Iterations per epoch: 5267

Maximum iterations: 15801

Validation

Frequency: 5267 iterations

Other Information

Hardware resource: Single CPU

Learning rate schedule: Piecewise

Learning rate: 8.1e-06

Learn more

RMSE

Training (smoothed)

Training

Validation

Loss

Training (smoothed)

Fig. 12 Convolutional model training progress

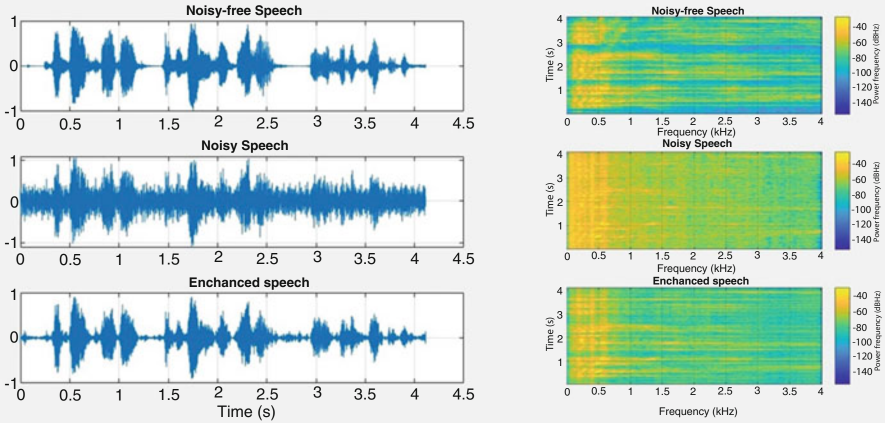


Fig. 13 Convolutional model result in the training stage

4 Conclusion

We have built three models to apply the Wiener filtering technique and neural networks for speech enhancement. The results from models for different samples are shown in Tables 1, 2, and 3. From the results obtained, we can clearly see that convolutional network performs better when compared to the other two models, but it requires very large amount of time for training and computation. We know that the resources are very limited and expensive to process such models. Besides this, requiring a very large computational time is a big disadvantage when it comes

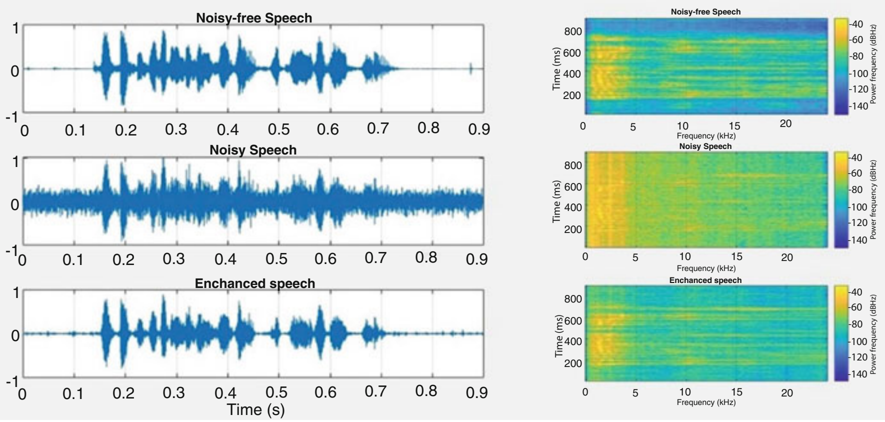


Fig. 14 Convolutional model result in the testing stage

Table 3 CNN model results

Sample	PSNR [dB]	SNR [dB]
1	26.4451	7.6137
2	25.9861	7.1547
3	26.1930	7.3617
4	26.3692	7.5378
5	26.3822	7.5508
6	25.9799	7.1485
7	26.2276	7.3962
8	26.5732	7.7461

to real-time applications. Also, when the model requires such huge resources, the model must also be very efficient, but the results obtained from the convolutional network model are not highly satisfactory. Therefore, we still need to optimize the model for better results.

References

1. G. K Rajini, V. Harikrishnan, Jasmin. M, S. Balaji, "A Research on Different Filtering Techniques and Neural Networks Methods for Denoising Speech Signals", IJITEE, ISSN: 2278-3075, Volume-8, Issue- 9S2, July 2019.
2. D. Liu, P. Smaragdis, M. Kim, "Experiments on Deep Learning for Speech Denoising", Interspeech, 2014.
3. F. G. Germain, Q. Chen, and V. Koltun, "Speech Denoising with Deep Feature Losses", arXiv:1806.10522v2 [eess.AS], 14 Sep 2018.
4. M. A. Abd El-Fattah, M. I. Dessouky, S. M. Diab and F. E. Abd El-samie, "Speech enhancement using an adaptive wiener filtering approach", Progress in Electromagnetics Research M, Vol. 4, 167–184, 2008.

5. A. Klos, M. S. Bos, R. M. S. Fernandes, “Noise-Dependent Adaption of the Wiener Filter for the GPS Position Time Series.” Math Geosci 51, 53–73 (2019). Available at https://doi.org/10.1007/s11004-018-9760-z.
6. S. China Venkateswarlu, K. Satya Prasad, A. Subbarami Reddy, “Improve Speech Enhancement Using Weiner Filtering”, Global Journal of Computer Science and Technology Volume 11 Issue 7 Version 1.0 May 2011.
7. This is from an online internet source which is available at https://en.wikipedia.org/wiki/Wiener_filter#Applications, as on 28th May 2021.
8. This is from an online internet source which is available at https://www.clear.rice.edu/elec431/projects95/lords/wiener.html, as on 28th May 2021.
9. Chavan, Karishma and Gawande, Ujwalla. (2015), “Speech recognition in noisy environment, issues and challenges: A review”, 1-5.10.1109/ICSNS.2015.7292420.
10. Article from an online internet source available here, as on 28th May 2021.
11. A. Dubey and M. Galley, “Experiments with Speech Enhancement Techniques”, available here, as on 28th May 2021.
12. This is from an online internet source which is available here, as on 28th May 2021.
13. This is from an online internet source which is available at https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm, as on 28th May 2021.
14. This is from an online internet source which is available here, as on 28th May 2021.
15. This is from an online internet source which is available at https://www.ibm.com/cloud/learn/convolutional-neural-networks, as on 28th May 2021.