SEMI-SUPERVISED DEEP LEARNING FOR SPEECH DENOISING

by Sherief Helwa and Josiah Smith

November 22, 2019

EESC6364
The University of Texas at Dallas

Contents

Table of Figures

Abstract

In this project, we demonstrate the efficacy of the semi-supervised deep learning technique for speech denoising. Using a mid-side microphone to capture two independent channels of an identical speech signal with independent additive noise signals, we train two fully convolutional neural networks (FCNNs) to perform real-time speech denoising in several noise environments. A thorough investigation is provided into the optimization of each FCNN's hyperparameters, training dataset size, training computational efficiency, and testing latency. After extensive data collection and network training, two highly robust FCNNs are produced, capable of speech denoising in machinery noise and babble noise environments. Lastly, a real-time platform is designed for audio capture, denoising, and playback of denoised audio signals for mobile testing.

1. Data Collection Setup and Hardware

To collect data for training, we considered several different environments and collection setups. We began by collecting speech and noise data simultaneously in the student union. While this approach gave us notably realistic data, we had limited control over the noise level and signal to noise ratio (SNR) to sufficiently support conclusions. As such, we moved to two different approaches. First, we developed a more controlled environment wherein we position the mid-side microphone directly in between two speakers and artificially finely control the SNR while speaking into the mic. This sterile environment we will refer to as the lab setup. The second alternative requires recording clean, noiseless voice signals and raw noise signals with no voice. Once we recorded several variants of noise environment types, we were able to digitally manipulate the power levels of the speech and noise signals independently to produce the desired SNRs. Throughout this report, we will refer to this method as the additive technique. Two microphones were used, the TASCAM TM-ST1 and the Shure MOTIV MV88.

2. Network Architecture and Common Hyperparameters

For every noise environment and setup, the following architecture is employed. First, an image input layer takes in the framed noisy signal one 20ms frame at a time. Next, we repeat a three-layer sequence of 2D convolution layer, batch normalization layer, and ReLU layer N times depending on the network. We will call this sequence of three layers a convolutional triple. The 2D convolution layer is of size [30,1], with 55 neurons, stride of [1,1], and zeros pads the signal to ensure the same input and output sizes from this layer. We also experimented with different non-linear layers including leaky ReLU and tanh but noticed minimal differences in performance. Finally, the last two layers are a 2D convolution layer of size [1,1] with a single neuron and a regression output layer.

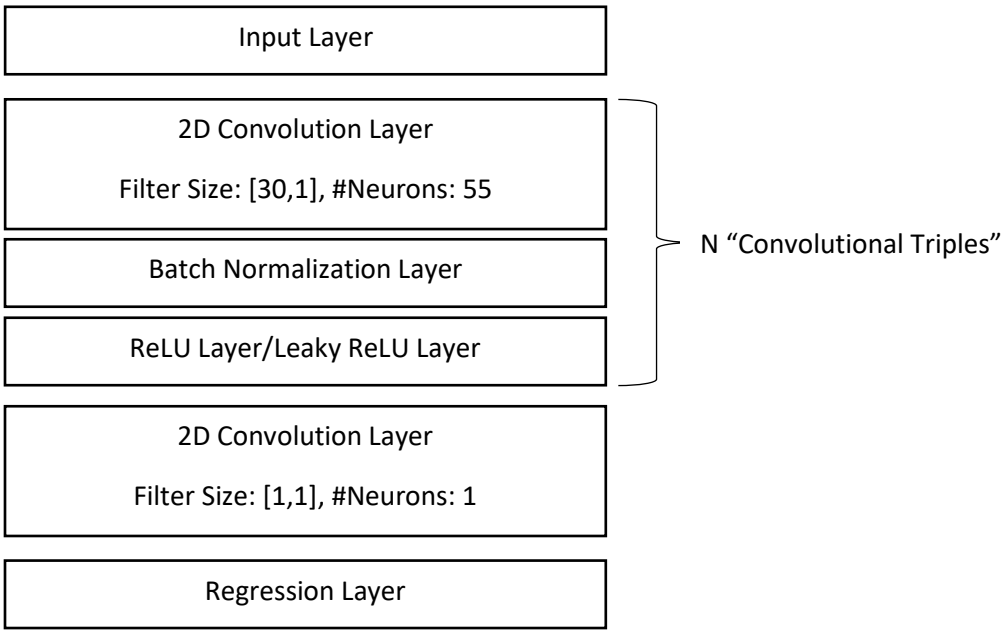

Figure 1: Network Architecture

For every network, we maintain the use of the Adam Optimization Algorithm, a mini-batch size of 128, and shuffling of the training data every epoch. Our investigation into the optimal hyperparameters is

centered around the learning rate and number of training epochs. The performance of our network in successfully denoising an audio signal seems to heavily rely on finely tuning these parameters.

3. Lab Setup Discussion and Results

Using the lab setup, with the TASCAM mid-side microphone positioned with a speaker on each side, we captured speech plus noise data with machinery background noise at various SNRs. However, likely due to the correlation of the noise channels, the results from our networks were not ideal. While we notice an audible reduction in the noise volume, especially during quiet, non-speech frames, the network appears also to distort the voice signal deeming this technique marginally unusable. For this analysis, we trained a network with the aforementioned architecture. Additionally, we used a piecewise learning rate with an initial learning rate of 0.001, a drop factor of 0.5, and a drop period of 10 epochs. We began by training the network for several epochs and noticed a strong correlation between the number epochs and the clarity of the signal and noise reduction. We settled on using 50 epochs for training and independently changing the number of convolutional triples. With these settings, using an NVIDIA GTX1050TI, the training for each case took approximately 30 minutes.

As we increase the number of convolutional triples from 5 to 20, we attempt to denoise the same sample data using each network. After feeding the sample through the network, we plot some of the data for examination and approximate the SNR before and after denoising. The difference between these SNRs is deemed the SNR improvement by each network and is a useful figure only in understanding the relationship between the number epochs and relative success of reducing noise power. However, for each of these networks, audible distortion is noticeable to the speech signal and these networks do not subjectively enhance the signal according to our subjective testing.

Figures 2-5 show the speech plus noise signals with some noise isolated. The control signal is the noisy signal prior to denoising. In these figures, it appears visibly that the networks are successfully removing noise from the sample data; however, this comes at the cost of distorting the speech signal. Figure 6 presents the SNR improvement across the number of convolutional triples. From 5 to 15 triples, we notice a directly proportional relationship between the number of convolutional triples and the SNR improvement.

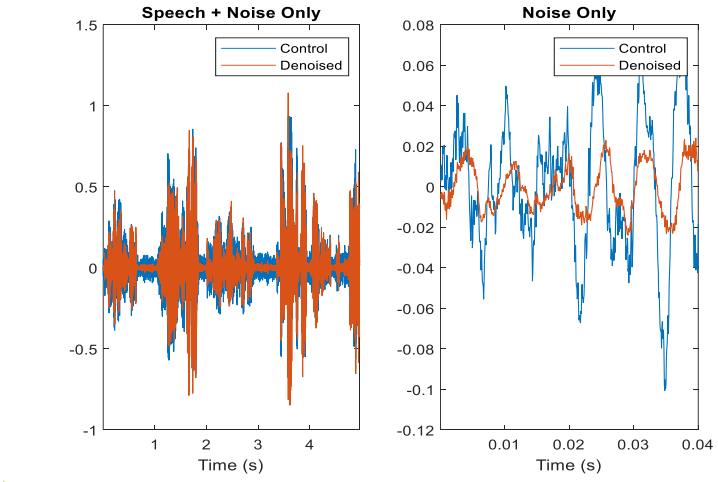

Figure 2: Speech and Noise from 5 Convolutional Triple Network

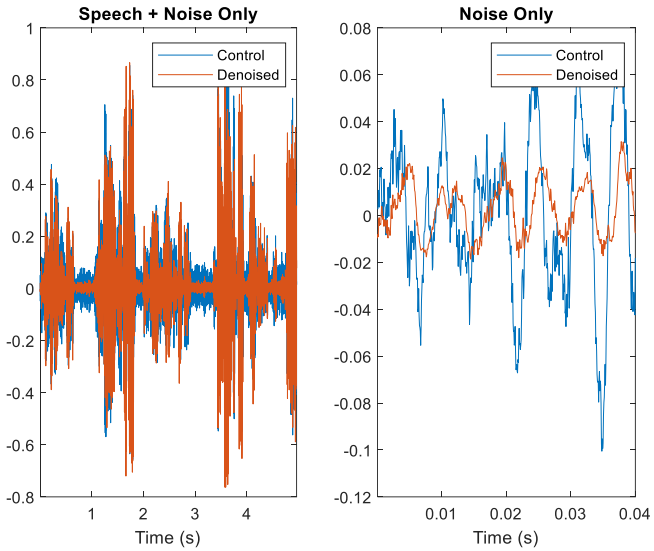

Figure 3: Speech and Noise from 10 Convolutional Triple Network

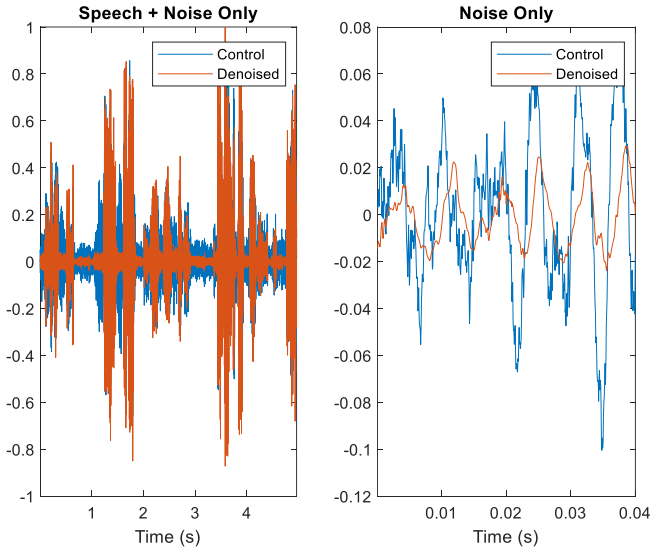

Figure 4: Speech and Noise from 15 Convolutional Triple Network

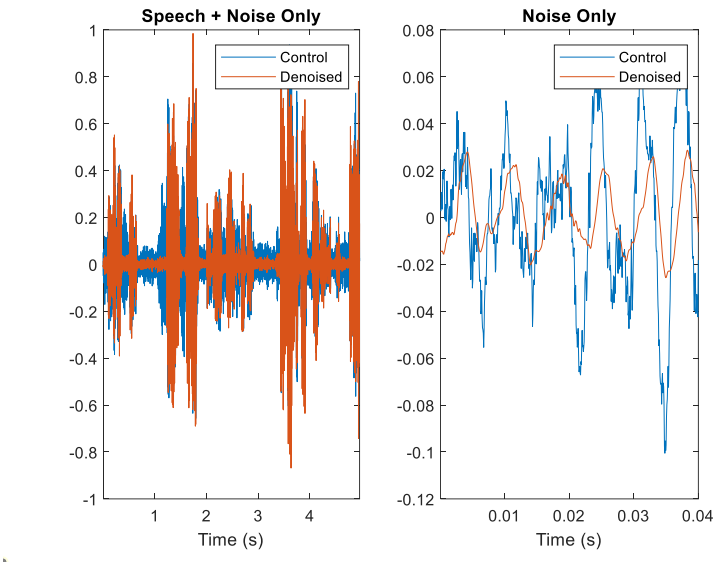

Figure 5: Speech and Noise from 20 Convolutional Triple Network

However, interestingly, while keeping the number of epochs constant (50), the distortion of the speech signal noticeably increases as we increase the number of convolutional triples. In a tangent study, we examined this relationship and found that an increase in epochs resulted in less distortion when increasing the number of convolutional triples. These observations are integral to the success of later networks further discussed in later sections.

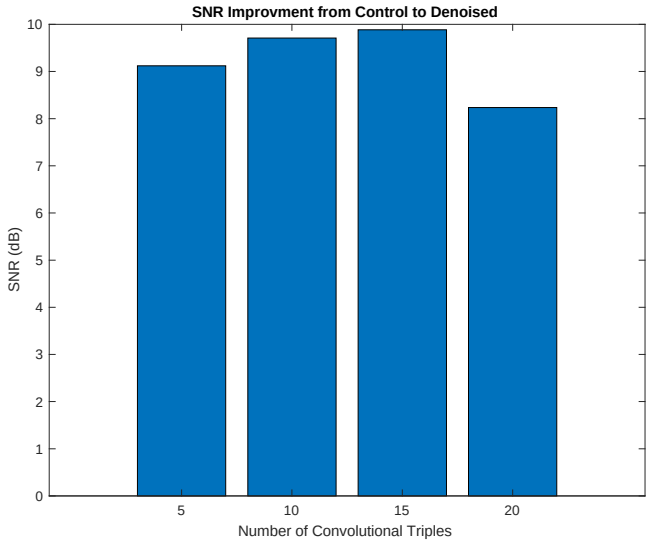

Figure 6: SNR Improvement Across the Number of Convolutional Triples

In further investigation, we considered the spectrum of the control noisy signal and the denoised signal. As expected, the spectrum of the control noisy signal includes high frequency noise appearing somewhat similar to white noise. As visible in figures 7-10, increasing the number of convolutional triples appears to have a denoising effect on the higher frequency noise, likely resulting in the SNR improvement previously discussed. However, there are noticeable increases in the frequency ranges around 600Hz, well outside

the fundamental frequency ranges of male subjects. The presence of these peaks is the likely cause of the aforementioned distortion. Upon close inspection, we notice the density of these high magnitude peaks around 600Hz as we increase the number of convolutional triples. While these networks have not provided sufficient denoising capability, although they visually appear to reduce the level of the noise, observing the relationship between the number of convolutional triples and the required number of epochs to avoid undesired distortion of the speech signal proves essential to the success of our final denoising networks.

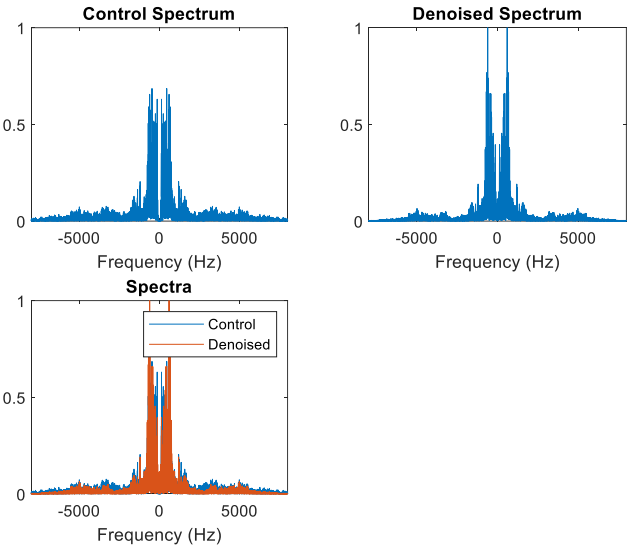

Figure 7: Spectra of Control and Denoised Signal for 5 Convolutional Triple Network

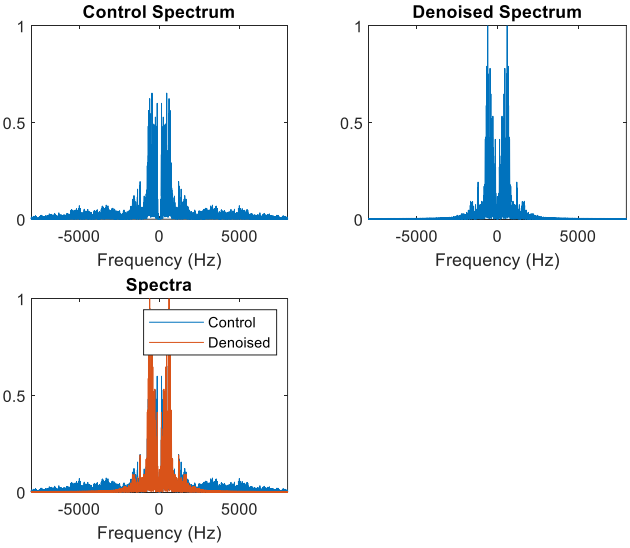

Figure 8: Spectra of Control and Denoised Signal for 10 Convolutional Triple Network

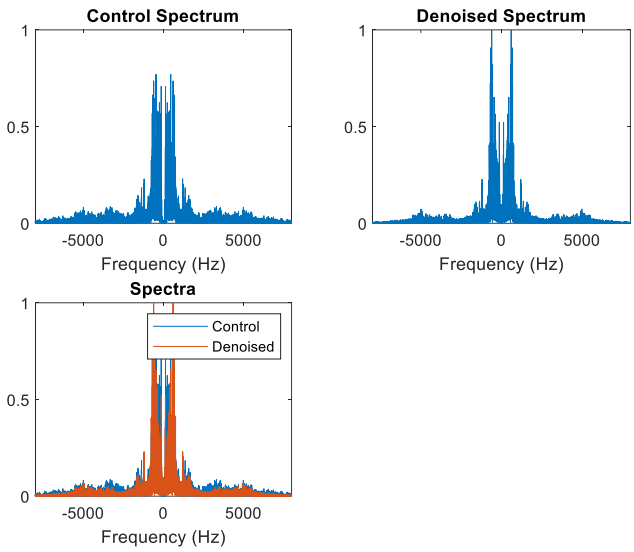

Figure 9: Spectra of Control and Denoised Signal for 15 Convolutional Triple Network

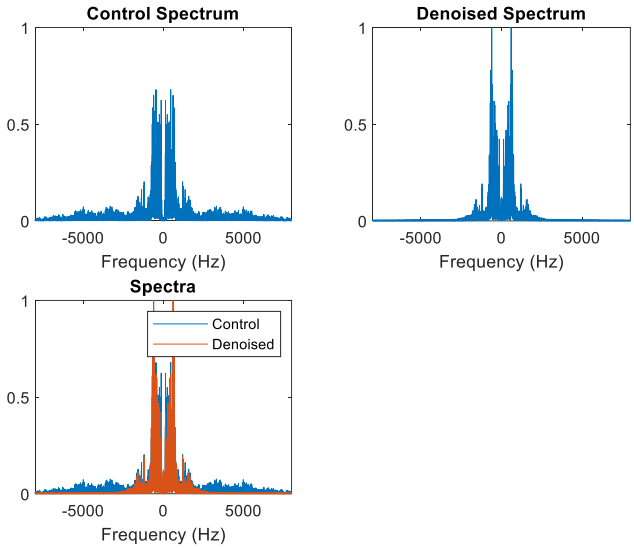

Figure 10: Spectra of Control and Denoised Signal for 20 Convolutional Triple Network

4. Additive Technique Discussion and Results

Through thorough investigation of the lab setup, the importance of the data became quite evident. In order to capture data more suitable for effectively training the network, we purchased the Shure MOTIV MV88 mid-side microphone and noticed significantly better results. In closely examining the noise across time, the Shure microphone offered much more uncorrelated channels than the TASCAM microphone. For the remainder of our data collection, we used the Shure microphone to capture sound samples. We captured three different noise environments using this microphone: fan noise from a spectrum analyzer,

car engine noise, and babble noise. In this section, we discuss the performance of our networks applied to each noise environment type and analyze the computational requirements of the networks.

a. Machinery Noise – Fan

One hour of noise was collected by recording a spectrum analyzer's fan. A 56-minute audio clip with a single, male subject is artificially contaminated by the fan noise captured by the Shure microphone. The average power of the noise and the clean speech signal are first normalized and then added in precise ratios to control the SNR of data being fed to the network for training. Using the fan noise, we considered three networks with the following parameters. Initial learning rate: 0.001, learning drop factor: 0.5, learning drop period: 10 epochs, total number of epochs: 75, number of convolutional triples: 3. Three networks are trained with input SNR values of 3dB, 5dB, and 10dB.

i. Computational Efficiency of Training

Using an NVIDIA GTX1050TI GPU for training, the training times for each network are given by table 1 below. Since we are using the same data and only scaling the power of the signal and noise signals, we expect to see nearly identical training times for all three networks. Computationally, maintaining the same network architecture and dataset size make each network essentially identical in computational efficiency of testing.

Network Input SNR	Training Time
3dB	193 min
5dB	187 min
10dB	184 min

Table 1: Fan Machinery Noise Training Times

ii. Computational Latency of Testing

Using the same device, the latency for each network is tested. Similarly, since the architectures of each network are identical, as expected, the average latencies are the same across all the networks. With framing done every 20ms, these latencies are well within the bounds of successful real-time denoising. Our provided script demonstrates this accordingly, being able to capture and denoise samples in real-time without missing or skipping frames. (Extensive work was done to ensure that the frame skipping would not occur. Under typical environments, i.e. no heavy CPU load, the machine can easily handle the audio capture and denoising process using these networks without loss.

Network Input SNR	Average Latency
3dB	5ms
5dB	5ms
10dB	5ms

Table 2: Fan Machinery Noise Latencies

iii. Subjective Evaluation of Denoising

Using a five-grade user preference rating specified in the project description (-2: much worse, -1: worse, 0: no difference, 1: better, 2: much better), we asked various participants to rate the denoising capabilities of these networks. The overwhelming response was that the network did indeed reduce the noise volume, but the reduction was not significant enough to be considered "much better."

Network Input SNR	Subjective Test Results
3dB	1
5dB	1.2
10dB	1

Table 3: Fan Machinery Noise Subjective Ratings

iv. SNR of Testing Audio Signals

After denoising the input signals with each network, a simple objective SNR calculation is performed across the speech and noise portions of the sample independently. This output SNR is compared to the input SNR calculated in the same manner to obtain the SNR gain. As shown in table 3 below, the largest SNR gain is produced using the 5dB network. This result is to be expected and aligns with the subjective testing indicating that the best performing network is the 5dB network.

Network Input SNR	SNR Gain
3dB	1.118dB
5dB	1.7141dB
10dB	1.3028dB

Table 4: Fan Machinery Noise SNR Gains

b. Machinery Noise – Car Engine

One hour of noise was collected by recording a running car engine. A 38-minute audio clip with a single, male subject is artificially contaminated by the babble noise captured by the Shure microphone. The average power of the noise and the clean speech signal are first normalized and then added in precise ratios to control the SNR of data being fed to the network for training. Using the captured babble noise, we considered networks with the following parameters. Initial learning rate: 0.01, learning drop factor: 0.5, learning drop period: 10 epochs. The number of convolutional triples and input SNR are varied and discussed in this section.

i. Key Observations

While considering the car engine noise environment, we made some key observations. While the RMSE appears to saturate around 5-10 epochs, training with fewer than 30 epochs results in a highly distorted network when denoising is attempted. Due to this distortion, not knowing that increasing the number of epochs would solve this issue, we first examined the architecture of the network. Thinking that perhaps we were experience some rectification issues due to the non-linearities of the network, we removed all non-linear layers (batch normalization and ReLU layers). As a result, we essentially constructed a purely linear system. As a result, the RSME reached significantly low values, about two orders of magnitude less than with the non-linear layers. However, in listening to the denoised signals from this network, the noise power does not audibly seem to have been reduced at all. This was when we noticed that there is no significant relationship between the RSME and the denoising performance. Removing the non-linear layers resulted in a much reduced RSME but did not result in denoising. Realizing this lack of correlation, we attempted to increase the number of epochs. Training with around 50 epochs yielded much better denoising even though the RSME appears to saturate and does not decrease after 5-10 epochs. Despite the fact the RMSE stays nearly constant, the network is learning the characteristics of the noise even during these valuable later epochs. Eventually, we changed the learning rate to higher values around 0.01 from the order of 0.0001 for the initial learning rate and noticed improved denoising capability from the network. Through an extensive trial and error process, we were able to make these key observations in

successfully producing an effective denoising network: non-linear layers are crucial, use 50 or more epochs, and use a large initial learning rate.

ii. Computational Efficiency of Training

Network	Training Time
10dB SNR, 3 Convolutional Triples, 80 Epochs	236 min
10dB SNR, 5 Convolutional Triples, 50 Epochs	140 min
15dB SNR, 5 Convolutional Triples, 50 Epochs	140 min

Table 5: Car Engine Machinery Noise Training Times

iii. Computational Latency of Testing

Consistent with our previous results, the latencies of these networks stay within range to provide real-time denoising without skipping or losing frames.

Network	Average Latency
10dB SNR, 3 Convolutional Triples, 80 Epochs	3.5ms
10dB SNR, 5 Convolutional Triples, 50 Epochs	5ms
15dB SNR, 5 Convolutional Triples, 50 Epochs	5ms

Table 6: Car Engine Machinery Noise Latencies

iv. Subjective Evaluation of Denoising

The trained networks performed quite well in audibly reducing the noise without heavily distorting the speech signal. All of our subjective test subjects rated the network with 2, on the previously discussed rating scale.

Network	Subjective Test Results
10dB SNR, 3 Convolutional Triples, 80 Epochs	2
10dB SNR, 5 Convolutional Triples, 50 Epochs	2
15dB SNR, 5 Convolutional Triples, 50 Epochs	2

Table 7: Car Engine Machinery Noise Subjective Ratings

v. SNR of Testing Audio Signals

Network	SNR Gain
10dB SNR, 3 Convolutional Triples, 80 Epochs	4.9dB
10dB SNR, 5 Convolutional Triples, 50 Epochs	4.8dB
15dB SNR, 5 Convolutional Triples, 50 Epochs	6.7dB

Table 8: Car Engine Machinery Noise SNR Gains

c. Babble Noise

One hour of noise was collected by recording the UTD student union. A 38-minute audio clip with a single, male subject is artificially contaminated by the babble noise captured by the Shure microphone. The average power of the noise and the clean speech signal are first normalized and then added in precise ratios to control the SNR of data being fed to the network for training. Using the captured babble noise, we considered networks with the following parameters. Initial learning rate: 0.01, learning drop factor: 0.5, learning drop period: 10 epochs, and input SNR of 10dB. The number of convolutional triples and

number of epochs are both discussed throughout this section. Table 9 shows the 6 different networks and their corresponding parameters.

# of Conv Triples	# Epochs
1	50
1	100
3	50
3	90
5	50
10	20

Table 9: Babble Noise Networks

i. Computational Efficiency of Training

For the babble noise environment case, the networks under consideration range from 1 convolutional triple to 10 convolutional triples. Thus, as expected, the training times vary significantly among the networks. Looking ahead the subjective testing results, all the networks except for the 10 convolutional triple network performed exceptionally well. This is to be expected as this network was only trained for 20 epochs rather than 50 or 100 epochs.

Network	Training Time
1 Convolutional Triple, 50 Epochs	16 min
1 Convolutional Triple, 100 Epochs	33 min
3 Convolutional Triple, 50 Epochs	60 min
3 Convolutional Triple, 90 Epochs	198 min
5 Convolutional Triple, 50 Epochs	150 min
10 Convolutional Triple, 20 Epochs	70 min

Table 10: Babble Noise Training Times

ii. Computational Latency of Testing

Varying the number of convolutional triples again results in computational variance among the networks, for networks with more convolutional triples, the average latency increases accordingly. As mentioned in the previous sections, the latencies of all these networks remain low enough to provide real-time denoising capabilities as demonstrated by our real-time platform.

Network	Average Latency
1 Convolutional Triple, 50 Epochs	2.5ms
1 Convolutional Triple, 100 Epochs	2.5ms
3 Convolutional Triple, 50 Epochs	3.5ms
3 Convolutional Triple, 90 Epochs	3.5ms
5 Convolutional Triple, 50 Epochs	5ms
10 Convolutional Triple, 20 Epochs	8ms

Table 11: Babble Noise Latencies

iii. Subjective Evaluation of Denoising

The babble denoising networks performed quite well for denoising with the proper hyperparameters and number of epochs. Not shown in this report are countless of failed attempts using smaller learning rates, less epochs, and smaller datasets. From these failures, we learned to keep the number of epochs at or above 50 epochs. As seen in the 10 convolutional triple network, only run for 20 epochs, the denoising

performs suffers greatly due to the lower number of epochs. Interestingly, the number of convolutional layers does not appear to have a drastic impact on the denoising effect if the network is provided enough data and proper hyperparameters. Even the networks with as few as 1 and 3 convolutional triples experience similar subjective test results. Some participants noted that the 3 convolutional triple networks audibly provided slightly better denoising and a cleaner speech signal.

Network	Subjective Test Results
1 Convolutional Triple, 50 Epochs	2
1 Convolutional Triple, 100 Epochs	2
3 Convolutional Triple, 50 Epochs	2
3 Convolutional Triple, 90 Epochs	2
5 Convolutional Triple, 50 Epochs	2
10 Convolutional Triple, 20 Epochs	0

Table 12: Babble Noise Subjective Ratings

iv. SNR of Testing Audio Signals

Considering the SNR gain for each of the networks, we see a similar SNR gain for most of the networks, with some increase in SNR gain as we increase the number of convolutional triples. The best sounding network with the cleanest speech signal and most denoising capabilities audibly is the 3 convolutional triple 90 epoch network. The speech and noise reduction is shown below in figure 11.

Network	SNR Gain
1 Convolutional Triple, 50 Epochs	3.2138dB
1 Convolutional Triple, 100 Epochs	3.1804dB
3 Convolutional Triple, 50 Epochs	3.97dB
3 Convolutional Triple, 90 Epochs	3.422dB
5 Convolutional Triple, 50 Epochs	4.321dB
10 Convolutional Triple, 20 Epochs	4.4222dB

Table 13: Babble Noise SNR Gains

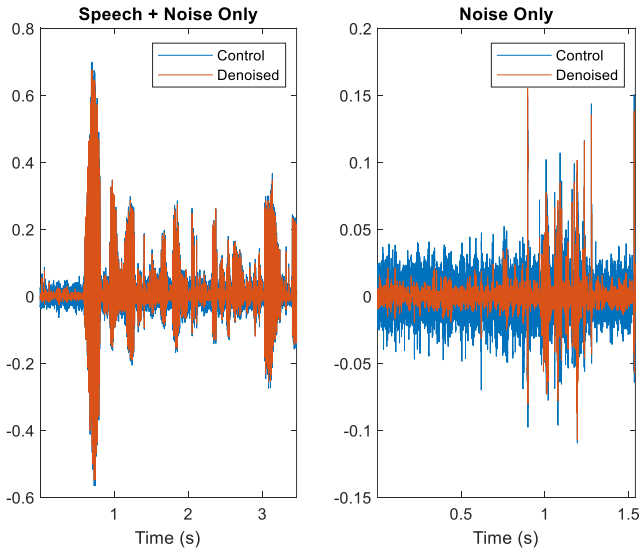

Figure 11: Speech and Noise from 3 Convolutional Triple 90 Epoch Network

Conclusion

The success of this project demonstrates the efficacy of semi-supervised deep learning for speech denoising in multiple noise environments. Without access to the clean speech signal, successful denoising of both babble noise and machinery noise contaminated speech signals using fully convolutional neural networks is achieved. Through a lengthy process, several key observations led to these successful networks. Namely, properly capturing uncorrelated noise channels for training, providing the network with enough training data, careful selection of learning rate and number of epochs, and suitable network architecture selection play a crucial role in the performance of the trained networks. As expected, a strong correlation is noted between the number of layers in a given network and its computational expense during training. Similarly, networks with fewer layers perform more efficiently in real-time scenarios providing less computational latency than more complex networks. Finally, a real-time platform is developed in MATLAB for real-time audio capture and denoising. A user may select between a machinery denoising network and a babble denoising network as desired.