Comparative Study of Speech Denoising Techniques

ECE 6255 Project

Aanish Nair, Ningyuan Yang, Prithwijit Chowdhury, Sumedh Ravi

Department of Electrical and Computer Engineering
Georgia Institute of Technology

Contents

Abstract

In many speech communication settings, the presence of background noise would degrade the quality or intelligibility of speech. For instance, the quality of speech signal may be influenced in the process of data conversion (microphone), transmission (noisy data channels) or reproduction (loudspeakers and headphones). Over the decades, various speech enhancement algorithms were proposed to improve the intelligibility and overall perceptual quality of the degraded speech signal. In this project, we concentrate on removing different types of additive background noise to enhance the speech quality. We make a careful comparison of two signal processing algorithms including Spectral Subtraction and Minima Controlled Recursive Averaging (MCRA) as well as one deep learning-based model called NSNet2. The Short-time objective intelligibility (STOI) is used for evaluating the performance of speech enhancement algorithms, and experimental results show that NSNet2 model is the best approach to suppress the background noise and the enhance speech quality.

1 Introduction

When a speaker communicates in a quiet environment, information exchange is easy and accurate and the listener can understand it quite easily. However, a noisy environment reduces the listener's ability to understand what is said. Noise can be introduced at the source of speech (background noise) or during transmission. This degradation of speech by noise creates problems not only for just interpersonal communication but more serious problems in applications in which decision or control is made on the basis of speech signal.

Intelligibility refers to the ability to recognize the actual content of speech, and quality refers to the aspect of speech that determines the ease with which one can understand the speech. The main objective of speech enhancement is to improve the perceptual aspects of speech such as overall quality, intelligibility, or degree of listener fatigue. Speech enhancement is becoming increasingly important in applications such as voice calling, recording and video conferencing, as there are background noises from different sources in our environment. The objective of achieving higher quality and intelligibility of noisy speech may also contribute to improved performance in other speech applications, such as speech compression, speech recognition, or speaker verification. Speech enhancement can be of different types:

1. Enhancement of speech affected by background additive noise.
2. Suppression of distortion from voice coding algorithms.
3. Suppression of competing speakers in a multi-speaker setting.
4. Enhancement of speech for hearing-impaired listeners.

In this project, we consider speech enhancement algorithms that fall under the first category.

2 Methodology

This section explains the methodology of different methods used in the project.

2.1 Spectral Subtraction

Spectral subtraction is a method to restore the power spectrum or the magnitude spectrum from the signal observed in additive noise, through subtraction of an estimate of the average noise spectrum from the noisy signal spectrum. Noise that accompanies electronic voice signals might be added as background noise or as part of the data compression or transmission process.

When the speech signal is absent and only the noise is present, the noise spectrum is usually approximated and updated. The key assumption is that noise is a stationary or slow-moving process with little variation in the noise spectra between updating periods. Thus, it is possible to retrieve the speech signal by removing an estimate of the noise from the noisy signal in cases where the noise is accessible on a separate channel in addition to the noisy signal.

The only signal accessible in many situations, such as at the receiver of a noisy communication channel, is the noisy signal. Although it is impossible to eliminate random noise in certain instances, it may be able to lessen the average impacts of noise on the signal spectrum. The magnitude spectrum of a signal is affected by additive noise, which increases the mean and variance of the spectrum.

The random fluctuations of the noise cause an increase in the variance of the signal spectrum that cannot be wiped entirely. By subtracting an estimate of the mean of the noise spectrum from the noisy signal spectrum, the increase in the mean of the signal spectrum can be reduced.

The following sub-section describes the principle of the spectral subtraction method. [1]

2.1.1 Principle of Spectral Subtraction Method

Let's consider a noisy signal which consists of clean speech degraded by statistically independent additive noise as

$$y[n] = s[n] + d[n] \tag{1}$$

where $y[n]$, $s[n]$ and $d[n]$ are sampled noisy speech, clean speech and additive noise, respectively. It is assumed that the additive noise is zero mean and uncorrelated with clean speech.

Because the speech signal is non-stationary and time variant, the noisy speech signal is processed frame by frame. Their representation in the short-time Fourier Transform (STFT) domain is given by

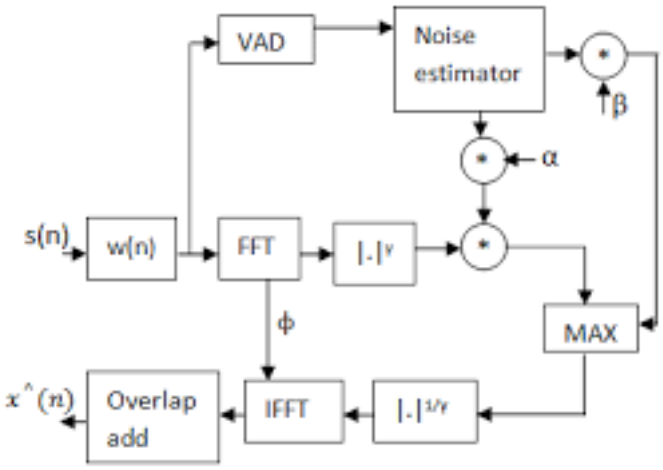

Figure 1: Block diagram representing spectral subtraction

$$Y(\omega,k) = S(\omega,k) + D(\omega,k) \tag{2}$$

where k is the frame index and ω is the frequency bin. It is assumed that the speech signal is segmented into frames, so we drop k for simplicity.

Since speech signal is assumed to be uncorrelated with the background noise, the short-term power spectrum of $y[n]$ has no cross-terms. Hence,

$$|Y(\omega)|^2 = |S(\omega)|^2 + |D(\omega)|^2 \tag{3}$$

The speech can be estimated by subtracting the noise estimate from the received signal.

$$|\hat{(}S(\omega)|^2 = |Y(\omega)|^2 - |\hat{D}(\omega)|^2 \tag{4}$$

The estimation of the noise spectrum $|\hat{D}(\omega)|^2$ is obtained by averaging recent speech pauses frames:

$$|\hat{D}(\omega)|^2 = \frac{1}{M}\sum_{j=0}^{M-1} |Y_{SP_j}(\omega)|^2 \tag{5}$$

where M denotes the number of consecutive frames of speech pauses (SP). If the background noise is stationary, (5) converges to the optimal noise power spectrum estimate as longer average is taken. [1]

Spectral subtraction can also be viewed as a filter, and by manipulating (4) it can be expressed as the product of the noisy speech spectrum and the spectral subtraction filter (SSF) as:

$$|\hat{S}(\omega)|^2 = \left(1 - \frac{|\hat{D}(\omega)|^2}{|Y(\omega)|^2}\right)|Y(\omega)|^2 \tag{6}$$

$$= H^2(\omega)|Y(\omega)|^2 \tag{7}$$

where $H(\omega)$ is the gain function of the spectral subtraction filter (SSF). $H(\omega)$ is a zero phase filter, with its magnitude response in the range of $0 \leq H(\omega) \leq 1$.

$$H(\omega) = \left(max\left(0, 1 - \frac{|\hat{D}(\omega)|^2}{|Y(\omega)|^2}\right)\right)^{\frac{1}{2}} \tag{8}$$

To reconstruct the desired signal, the phase estimate of the speech is also needed. A common phase estimation method is to adopt the phase of the noisy signal as the phase of the estimated clean speech signal, based on the notion that short-term phase is relatively unimportant to human ears. [2]

Then, the speech signal in one frame is estimated as

$$\hat{S}(\omega) = |\hat{S}(\omega)|e^{j<Y(\omega)} = H(\omega)Y(\omega) \tag{9}$$

The estimated speech waveform is recovered in the time domain by inverse Fourier transforming $\hat{S}(\omega)$ using an overlap and add approach. [1]

I now proceed to explain the Algorithmic Design for the Spectral Subtraction method.

The spectral subtraction task is to deliver a buffer of noise-suppressed speech to the vocoder analyzer over a time interval. This time interval is not only shorter than the buffer length duration, but it is also short enough to allow the analyzer to compute and the vocoder channel parameters to be transmitted. This is a limitation in the algorithm's implementation.

2.1.2 Data Segmentation

Because the length of speech compression analyzer buffer differs, the simplest method is to match the noise suppression analysis buffer to the vocoder's. However, this strategy results in two trade-offs.

1. Zeros must be appended before the transformation if the buffer is not a power of two.
2. No overlapping is allowed if the buffer lengths are to be matched with minimum delay.

Padding with zeros leads to lower efficiency since there are a fewer number of points that are processed per FFT. However, this also has the positive effect of reducing the amount of temporal aliasing due to spectral modification [3]. While having no overlap doubles the processing speed, it has the detrimental effect of introducing discontinuities at the buffer boundaries. This can be minimized if the data is weighted by half-overlapped hanning windows.

2.1.3 Bidirectional Biplexed DFT

Spectral subtraction requires two DFTs to be performed.

1. A forward transform of the noisy signal $x[n]$.
2. An inverse transform of the noise suppressed spectrum, S[k] = X[k]H[k].

A biplexed DFT is developed, which can simultaneously compute the forward transform of $x[n]$ and the inverse transform of S[k] and is given by:

$$Re(i) = x_o(i) + SR(i)/N \tag{10}$$

$$Im(i) = x_e(i) - SI(i)/N \tag{11}$$

where,

$$x_e(i) = (x(i) + x(N-i))/2 \tag{12}$$

$$x_o(i) = (x(i) - x(N-i))/2 \tag{13}$$

$SR(i)$ = Real part of $S(k)$
$SI(i)$ = Imaginary part of $S(k)$
N = DFT size

Let $C(k) + jD(k) = \text{DFT}[RE(i) + jIM(i)]$

Then,

$$s(k) = C(k) \tag{14}$$

$$Re[X(k)] = (D(k) + D(N-k))/2 \tag{15}$$

$$Im[X(k)] = (D(k) - D(N-k))/2 \tag{16}$$

where
$s(K)$ equals the inverse DFT of $S(k)$
$Re[X(k)]$ = Real part of $X(k)$
$Im[X(k)]$ = Imaginary part of $X(k)$

2.1.4 Speech Activity Detection

An accurate estimation of the average noise bias B(k) is required for effective noise suppression. The bias should be updated during the next interval of non-speech activity if the ambient noise becomes either louder or softer.

The estimated signal-to-noise ratio can be used for detecting the absence of speech activity during a stationary noise interval and/or detecting a decrease in the noise bias.

$$\mathrm{SNR(k)} = \mathrm{H(k)}/(1-\mathrm{H(k)}) = \mathrm{S(k)}/\mathrm{B(k)} \tag{17}$$

Computing the average SNR over all frequency bins provides a measure of the relative energy of the signal as compared to the average noise bias.This measure can also detect when the ambient noise becomes less. In that instance, more values of X(K) will lie below B(k) and thus more values of H(k) will be zero driving the average value down.

2.2 Minima Controlled Recursive Averaging (MCRA)

2.2.1 Noise Spectrum Estimation

The key of the MCRA algorithm is to use recursive formula to estimate the noise spectrum [4]. Let $x(n)$ and $d(n)$ denote speech signal and uncorrelated additive noise , respectively. The observed signal $y(n)$ given by $y(n) = x(n) + d(n)$, is divided into overlapping frames by the application of a window function and analyzed using the short-time Fourier transform (STFT). Specifically,

$$Y(k,l) = \sum_{n=0}^{N-1} y(n+lM)h(n)e^{-j(2\pi/N)nk} \tag{18}$$

where k is the frequency bin, l is the frame index, h is an analysis window of size N, and M is the frame update step in time.

Two hypotheses, $H_0(k,l)$ and $H_1(k,l)$[4], are defined as speech absence and presence in the lth frame and kth sub-band according to (19) and (20),

$$H_0(k,l) : Y(k,l) = D(k,l) \tag{19}$$

$$H_1(k,l) : Y(k,l) = X(k,l) + D(k,l) \tag{20}$$

where $X(k,l)$ and $D(k,l)$ represent the STFT of clean signal and noisy signal, respectively. Let $\lambda_d(k,l) = E[|D(k,l)|^2]$ denote the variance of the noise in the kth sub-band. Then, a common technique to obtain its estimate is to apply a temporal recursive smoothing to the noisy measurement during periods of speech absence. In particular,

$$H_0(k,l) : \lambda_d(k,l+1) = \alpha_d \lambda_d(k,l) + (1-\alpha_d)|Y(k,l+1)|^2 \tag{21}$$

$$H_1(k,l) : \lambda_d(k,l+1) = \lambda_d(k,l) \tag{22}$$

where $\alpha_d (0 < \alpha_d < 1)$ is a smoothing parameter.

Let $p(k,l) = P(H_1(k,l)|Y(k,l))$ denote the conditional signal presence probability, then we update $\lambda_d(k,l)$ by

$$\lambda_d(k,l+1) = \lambda_d(k,l) * p(k,l) + [\alpha_d \lambda_d(k,l) + (1-\alpha_d)|Y(k,l+1)|^2] * (1-p(k,l)) \tag{23}$$

Then, we get the following equation:

$$\lambda_d(k,l+1) = \tilde{\alpha}_d(k,l) * \lambda_d(k,l) + [1-\tilde{\alpha}_d(k,l)] * |Y(k,l+1)|^2 \tag{24}$$

where

$$\tilde{\alpha}_d(k,l) = \alpha_d + (1-\alpha_d)p(k,l) \tag{25}$$

is a time-varying smoothing parameter. Based on (24) and (25), the noise spectrum can be estimated by averaging past spectral power values, using a smoothing parameter that is adjusted by the signal presence probability.

2.2.2 Signal Presence Probability

The second part describes some steps to compute $p(k,l)$.

Speech presence in a given frame of a sub-band is determined by the ratio between the local energy of the noisy speech and its minimum within a specified time window. Let the local energy of the noisy speech be obtained by smoothing the magnitude squared of its STFT in time and frequency. In frequency, we use a window function b whose length is $2w+1$

$$S_f(k,l) = \sum_{i=-w}^{w} b(i)|Y(k-i,l)|^2. \tag{26}$$

In time, the smoothing is performed y a first order recursive averaging, given by

$$S(k,l) = \alpha_s S(k,l-1) + (1-\alpha_s)S_f(k,l) \tag{27}$$

$\alpha_s (0 < \alpha_s < 1)$ is a parameter.

The minimum of the local energy, $S_{\min}(k,l)$, is searched according to the following procedure[5]. First of all, the minimum and a temporary variable $S_{\text{tmp}}(k,l)$ are initiated by $S_{\min}(k,0) = S(k,0)$ and $S_{\text{tmp}}(k,0) = S(k,0)$. Then, we use (28) and (29) to update $S_{\min}(k,l)$ and $S_{\text{tmp}}(k,l)$.

$$S_{\min}(k,l) = \min\{S_{\min}(k,l-1), S(k,l)\} \tag{28}$$

$$S_{\text{tmp}}(k,l) = \min\{S_{\text{tmp}}(k,l-1), S(k,l)\} \tag{29}$$

Whenever L frames have been read (now l is divisible by L), the temporary variable is employed and initialized by

$$S_{\min}(k,l) = \min\{S_{\text{tmp}}(k,l-1), S(k,l)\} \tag{30}$$

$$S_{\text{tmp}}(k,l) = S(k,l) \tag{31}$$

and search for the minimum continues with (28) and (29).

Finally, let $S_r(k,l) = S(k,l)/S_{\min}(k,l)$ denote the ratio between the local energy of the noisy speech and its derived minimum. The proposed estimator for $p(k,l)$ is given as follows:

$$p(k,l) = \alpha_p * p(k,l-1) + (1-\alpha_p) * I(k,l) \tag{32}$$

where $\alpha_p (0 < \alpha_p < 1)$ is a smoothing parameter and $I(k,l)$ is given by

$$I(k,l) = \begin{cases} 1 & S_r(k,l) > \delta \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

and δ is a threshold.

2.2.3 Flow Diagram

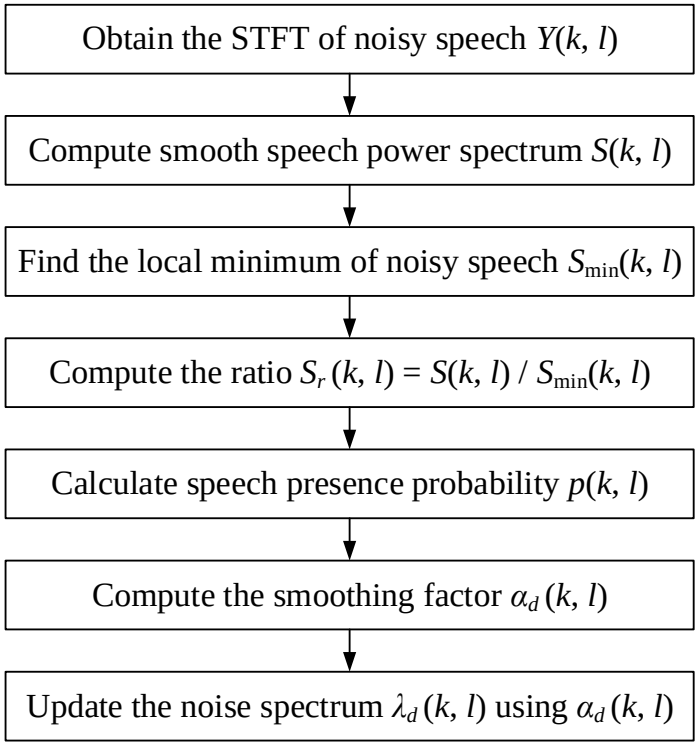

Figure 2: Flow diagram of the noise estimation algorithm

2.3 Deep Learning-based Noise Suppression (NSNet2)

Speech enhancement using deep neural networks(DNNs) has recently received a lot of attention and success in research, and it's also being used in commercial applications that are aimed at real-time communication. Deep learning-based noise suppression [6] has additional benefits of reducing extremely non-stationary noise and background noise.

The dataset is an essential component in data-driven learning techniques, particularly supervised learning. It is difficult to create a dataset that is large enough to generalize successfully while yet adequately representing expected real-world data.

NSNet2 [7] was released as a baseline for the Deep Noise Suppression (DNS) Challenge hosted yearly by Microsoft and is the current state-of-the-art (STOA) for noise suppression.

2.3.1 Principle

It is assumed that the observed signal in a pure noise reduction job is an additive combination of the intended speech and noise. The observed signal is denoted as $X(k,n)$ in the STFT domain, which is given in (33),

$$X(k,n) = S(k,n) + N(k,n) \tag{34}$$

where $S(k,n)$ represents speech signal, $N(k,n)$ represents the disruptive noise signal, k is the frequency index and n is the time frame index. It should be noted that the speech signal $S(k,n)$ might be reverberant, and our goal is just to reduce additive noise. The goal is to recover an approximation of the speech signal $widehatS(k,n)$ by applying a filter $G(k,n)$ to the observed signal according to (34).

$$\widehat{S}(k,n) = G(k,n)X(k,n) \tag{35}$$

The filter $G(k,n)$ might be a real-valued or complex-valued suppression gain. While the first technique (also known as masking) just recovers the voice amplitude, a complicated filter might theoretically restore the signal phase as well. A suppression gain is used in this study.

2.3.2 Algorithmic Design

A rather straightforward recurrent network architecture based on gated recurrent units (GRUs) and feed forward (FF) layers, similar to the core architecture of [8] without convolutional encoder layers have been used to construct NSNet2.

2.3.3 Network and Training

Input features are the logarithmic power spectrum $P = \log_{10}\left(|X(k,n)|^2 + \varepsilon\right)$, normalized by the global mean and variance of the training set. An STFT size of 512 with 32 ms square-root Hanning windows and 16 ms frame shift is used; but only the relevant 255 frequency bins

have been fed into the network, omitting 0th and highest (Nyquist) bins, which do not contain useful information. The network also consists of a FF embedding layer, two GRUs, and three FF mapping layers. All FF layers use rectified linear unit (ReLU) activations, except for the last output layer. When estimating a real-valued suppression gain, a Sigmoid activation is used to ensure positive output. The network architecture is shown in Fig. 3, and has 2.8M parameters.

The network was trained using the AdamW optimizer [9] with an initial learning rate of 10^{-4}, which was dropped by a factor of 0.9 if the loss plateaued for 5 epochs.

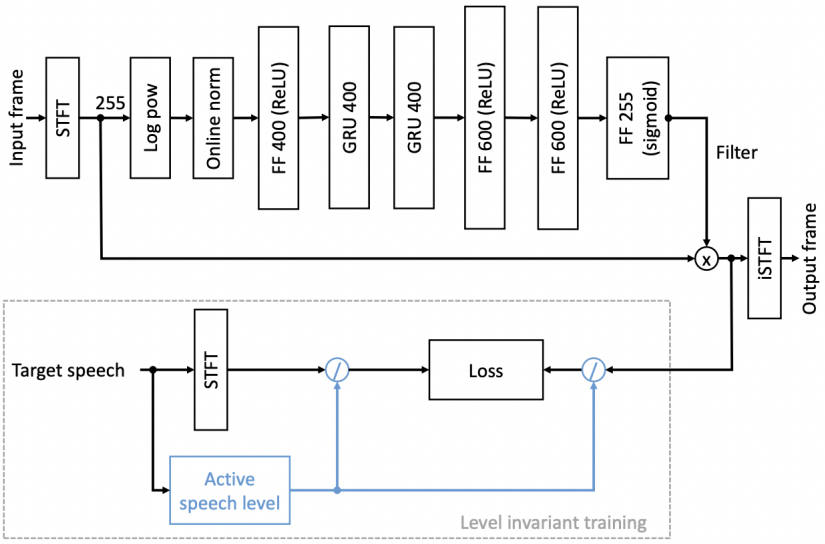

Figure 3: Network architecture and enhancement system and training procedure

2.3.4 Level Invariant Normalized Loss Function

The speech enhancement loss function is typically a distance metric between the enhanced and target spectral representations. The dynamically compressed loss proposed in [10] has been shown to be very effective. A compression exponent of $0 < c \leq 1$ is applied to the magnitudes, while the compressed magnitudes are combined with the phase factors again. Furthermore, the magnitude only loss is blended with the complex loss with a factor $0 \leq \alpha \leq 1$

$$\mathscr{L} = \alpha \sum_{k,n} ||S|^c e^{j\varphi_S} - |\widehat{S}|^c e^{j\varphi_{\widehat{S}}}|^2 + (1-\alpha) \sum_{k,n} ||S|^c - |\widehat{S}|^c|^2 \tag{36}$$

A value of $c = 0.3$ and $\alpha = 0.3$ were chosen

3 Experimental Results

Different types of background additive noise have different properties. Some noise is predictable and unchanging, which is called stationary noise. The other noise that is continuously changing is termed non-stationary noise. The difficulty for a listener to understand what a speech is depends on the signal-to-noise ratio (SNR) and the type of noise. In our experiments, we choose three kinds of noise including white noise, babble noise and factory noise, where the first kind is stationary noise and the others are non-stationary noise.

With many speech enhancement methods available today, we can choose the optimal method for a given application if we are able to evaluate performance of a speech enhancement technique accurately. The traditional SNR metric is not appropriate for comparing the denoised speech with the noisy speech. This is because the traditional SNR does not take into account perpetual effects or correlate with speech intelligibility. Instead, we use Short-Time Objective Intelligibility (STOI) score as the evaluation metric to compare the performance of different approaches. STOI is defined as a correlation of short-time temporal envelopes between clean and noisy speech, which has been shown to be positively correlated to human speech intelligibility score.

3.1 Spectral Subtraction

STOI for Female Speech (Spectral Subtraction)					
Type of speech	**SNR Value in dB**				
	0	5	10	15	20
Noisy (White)	0.7760	0.8555	0.9195	0.9627	0.9855
Denoised (White)	0.7827	0.8628	0.9250	0.9654	0.9860
STOI Gain	**+0.0067**	**+0.0073**	**+0.0055**	**+0.0027**	**+0.0005**
Noisy (Babble)	0.7295	0.8337	0.9113	0.9588	0.9830
Denoised (Babble)	0.7258	0.8362	0.9143	0.9605	0.9834
STOI Gain	**-0.0037**	**+0.0025**	**+0.0030**	**+0.0017**	**+0.0004**
Noisy (Factory)	0.7180	0.8286	0.9113	0.9604	0.9847
Denoised (Factory)	0.7206	0.8340	0.9150	0.9621	0.9848
STOI Gain	**+0.0026**	**+0.0054**	**+0.0037**	**+0.0017**	**+0.0001**

Table 1: STOI score for noisy and denoised female speech of different SNR values

Table 1 and Table 2 show the STOI score for female and male speech, respectively. However, what we really care about is the STOI gain, i.e. the difference in the score between the denoised and the noised signals, which is shown in Fig. 4 and Fig. 5 for female and male speech, respectively. Some of the conclusions that can be made are given below:

1. For female speech, the STOI gain is maximum at 5 dB for white and factory noise but STOI gain is maximum at 10 dB for babble noise.

STOI for Male Speech (Spectral Subtraction)					
Type of speech	**SNR Value in dB**				
	0	5	10	15	20
Noisy (White)	0.7340	0.8116	0.8740	0.9195	0.9496
Denoised (White)	0.7330	0.8137	0.8767	0.9212	0.9498
STOI Gain	**-0.001**	**+0.0021**	**+0.0027**	**+0.0017**	**+0.0002**
Noisy (Babble)	0.6646	0.7774	0.8654	0.9211	0.9519
Denoised (Babble)	0.6541	0.7738	0.8639	0.9197	0.9502
STOI Gain	**-0.0105**	**-0.0036**	**-0.0015**	**-0.0014**	**-0.0017**
Noisy (Factory)	0.6666	0.7767	0.8615	0.9171	0.9494
Denoised (Factory)	0.6575	0.7732	0.8604	0.9195	0.9476
STOI Gain	**-0.0091**	**-0.0035**	**-0.0011**	**+0.0024**	**-0.0018**

Table 2: STOI score for noisy and denoised male speech of different SNR values

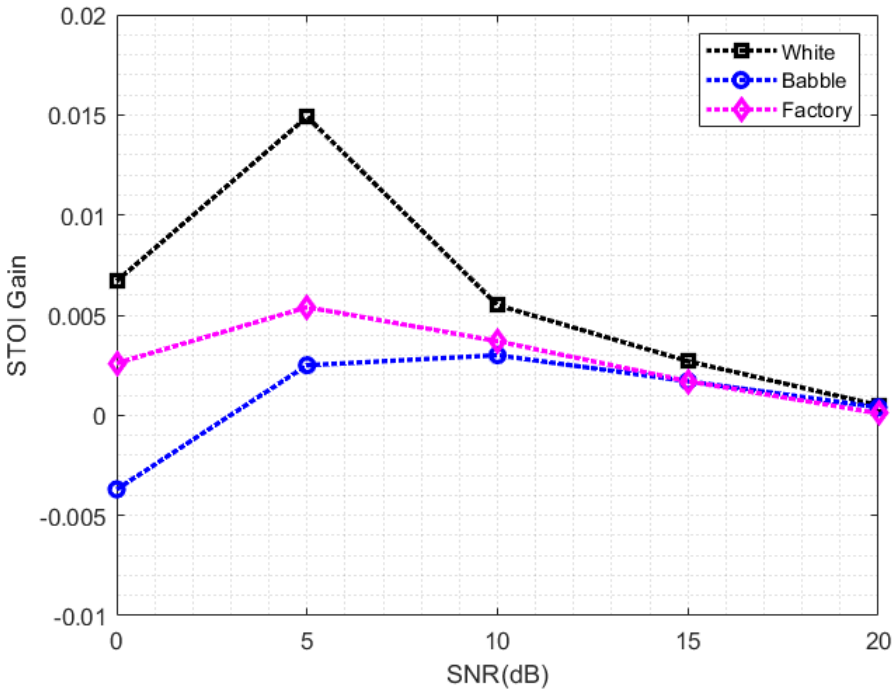

Figure 4: STOI gain for female speech of different types of noise and SNR (Spectral Subtraction)

2. For female speech, the STOI gain is ≈ 0 across all three types of noises.
3. For male speech, a majority of the values have negative STOI gain.
4. For male speech, the STOI gain for babble and factory noise take on approximately similar values.

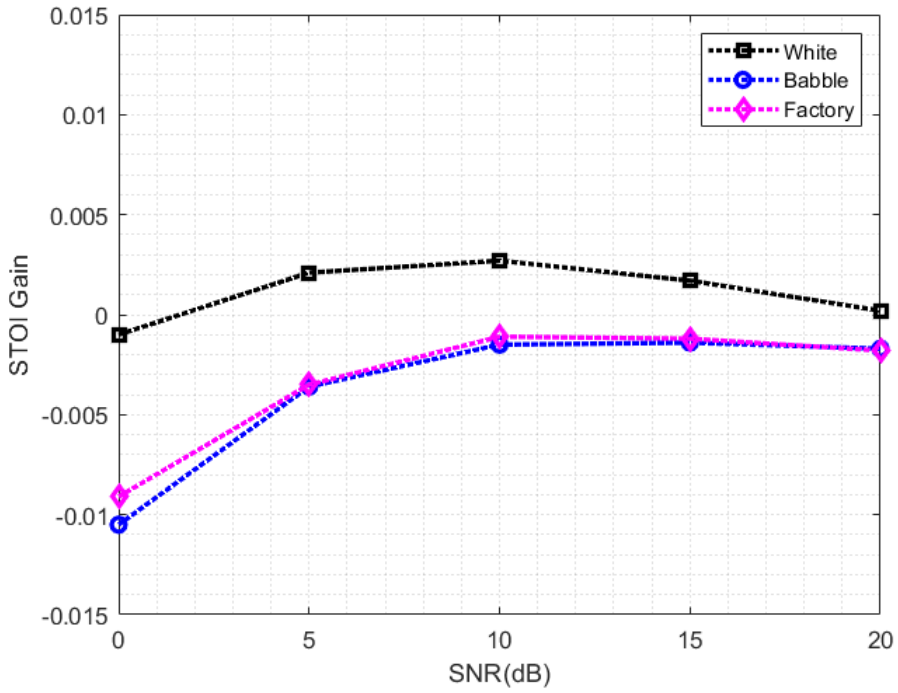

Figure 5: STOI gain for male speech of different types of noise and SNR (Spectral Subtraction)

5. For male speech, the STOI gain for white noise is higher than any other noise.

3.2 Minima Controlled Recursive Averaging (MCRA)

STOI for Female Speech (MCRA)					
Type of speech	**SNR Value in dB**				
	0	5	10	15	20
Noisy(White)	0.7760	0.8555	0.9195	0.9627	0.9855
Denoised(White)	0.7896	0.8704	0.9314	0.9673	0.9860
STOI Gain	**+0.0136**	**+0.0149**	**+0.0119**	**+0.0046**	**+0.0005**
Noisy(Babble)	0.7295	0.8337	0.9113	0.9588	0.9830
Denoised(Babble)	0.7300	0.8423	0.9179	0.9620	0.9843
STOI Gain	**+0.0005**	**+0.0086**	**+0.0066**	**+0.0032**	**+0.0013**
Noisy(Factory)	0.7180	0.8286	0.9113	0.9604	0.9847
Denoised(Factory)	0.7225	0.8409	0.9196	0.9638	0.9855
STOI Gain	**+0.0045**	**+0.0123**	**+0.0083**	**+0.0034**	**+0.0008**

Table 3: STOI score for noisy and denoised female speech of different SNR values

STOI for Male Speech (MCRA)					
Type of speech	**SNR Value in dB**				
	0	5	10	15	20
Noisy(White)	0.7340	0.8116	0.8740	0.9195	0.9496
Denoised(White)	0.7258	0.8162	0.8807	0.9245	0.9525
STOI Gain	**-0.0082**	**+0.0046**	**+0.0067**	**+0.0050**	**+0.0029**
Noisy(Babble)	0.6646	0.7774	0.8654	0.9211	0.9519
Denoised(Babble)	0.6505	0.7791	0.8703	0.9248	0.9535
STOI Gain	**-0.0141**	**+0.0017**	**+0.0049**	**+0.0037**	**+0.0016**
Noisy(Factory)	0.6666	0.7767	0.8615	0.9171	0.9494
Denoised(Factory)	0.6541	0.7778	0.8664	0.9206	0.9508
STOI Gain	**-0.0125**	**+0.0011**	**+0.0049**	**+0.0035**	**+0.0014**

Table 4: STOI score for noisy and denoised male speech of different SNR values

Table 3 and Table 4 show the results of STOI score and its gain. Fig. 6 and Fig.7 show the STOI gain for female speech and male speech, respectively. Here are some conclusions we could draw from the two figures:

1. For female speech, the STOI gain reaches its maximum when SNR = 5dB for three kinds of noise.
2. For female speech, the STOI gain: White $>$ Factory $>$ Babble when SNR = 0, 5, 10 and 15dB.
3. For female speech, the STOI gain for factory noise is the largest compared with white noise and babble noise when SNR = 20dB.
4. For male speech, the STOI gain: White $>$ Babble $>$ Factory for different SNR.
5. For male speech, the STOI gain is less than 0 when SNR = 0dB for three kinds of noise.
6. For male speech, the STOI gain reaches its maximum when SNR = 10dB for three kinds of noise.

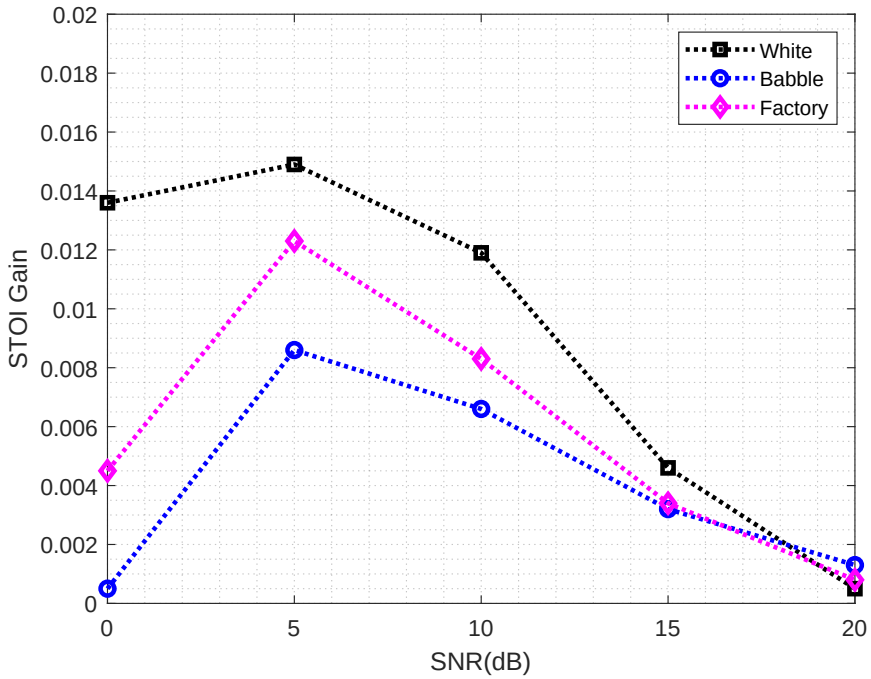

Figure 6: STOI gain for female speech of different types of noise and SNR (MCRA)

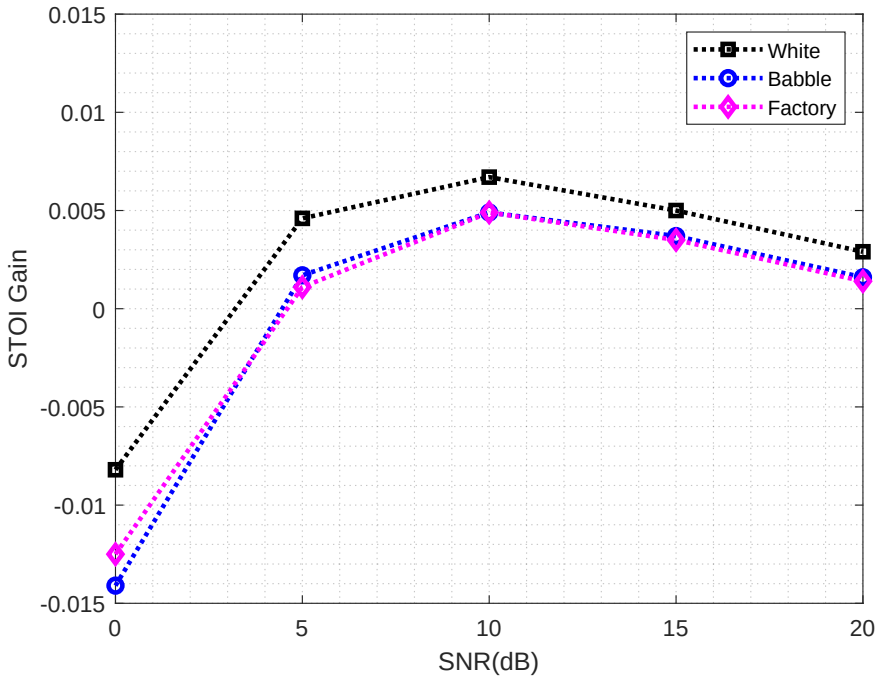

Figure 7: STOI gain for male speech of different types of noise and SNR (MCRA)

3.3 Deep Learning based Noise Suppression

Table 5 and Table 6 show results of STOI score and its gain. Fig. 8 and Fig.9 show the STOI gain for female speech and male speech, respectively. Here are some conclusions we draw:

1. STOI gain is max when SNR = 0dB for three kinds of noise, and for both male and female speech.
2. For female speech, STOI gain for factory noise is the largest when SNR = 0, 5 and 10dB, for both male and female voices.
3. For male speech, STOI gain: Factory $>$ Babble $>$ White for all SNR values till is almost converges at SNR = 20dB.
4. For both female and male speech, the STOI gain is positive for three kinds of noise.
5. For the model NSNet2, STOI gain is much more linear that the other two models.

STOI for Female Speech (NSNet2)					
Type of Speech	**SNR Value in dB**				
	0	5	10	15	20
Noisy(White)	0.7760	0.8555	0.9195	0.9627	0.9855
Denoised(White)	0.8545	0.9115	0.9504	0.9735	0.9865
STOI Gain	**+0.0785**	**+0.0560**	**+0.0309**	**+0.0107**	**+0.0010**
Noisy(Babble)	0.7295	0.8337	0.9113	0.9588	0.9830
Denoised(Babble)	0.7856	0.8862	0.9452	0.9734	0.9865
STOI Gain	**+0.0561**	**+0.0525**	**+0.0339**	**+0.0146**	**+0.0035**
Noisy(Factory)	0.7180	0.8286	0.9113	0.9604	0.9847
Denoised(Factory)	0.8101	0.8957	0.9461	0.9735	0.9874
STOI Gain	**+0.0921**	**+0.0671**	**+0.0348**	**+0.0131**	**+0.0027**

Table 5: STOI score for noisy and denoised female speech of different SNR values

STOI for Male Speech (NSNet2)					
Type of Speech	**SNR Value in dB**				
	0	5	10	15	20
Noisy(White)	0.7340	0.8116	0.874	0.9195	0.9496
Denoised(White)	0.8033	0.8671	0.9122	0.9415	0.9590
STOI Gain	**+0.0693**	**+0.0555**	**+0.0382**	**+0.0220**	**+0.0094**
Noisy(Babble)	0.6646	0.7774	0.8654	0.9211	0.9519
Denoised(Babble)	0.7477	0.8441	0.9057	0.941	0.9604
STOI Gain	**+0.0832**	**+0.0667**	**+0.0403**	**+0.0199**	**+0.0085**
Noisy(Factory)	0.6666	0.7767	0.8615	0.9171	0.9494
Denoised(Factory)	0.7696	0.8582	0.9108	0.9420	0.9605
STOI Gain	**+0.1030**	**+0.0815**	**+0.0493**	**+0.0249**	**+0.0111**

Table 6: STOI score for noisy and denoised male speech of different SNR values

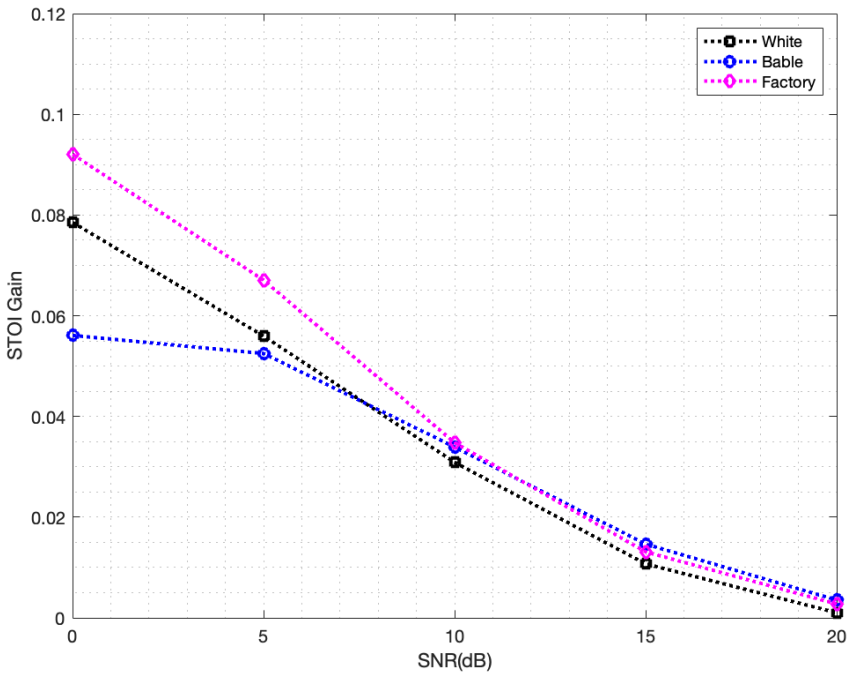

Figure 8: STOI gain for female speech of different types of noise and SNR (NSNet2)

3.4 Comparsion of three methods

3.4.1 Wave and Spectrum plots

First of all, we make a comparison of these three methods when female speech is mixed with white noise with SNR = 0 dB. Fig. 10 to Fig. 14 show the results of noisy speech, denoised speech using Spectral Subtraction, denoised speech using MCRA, denoised speech using

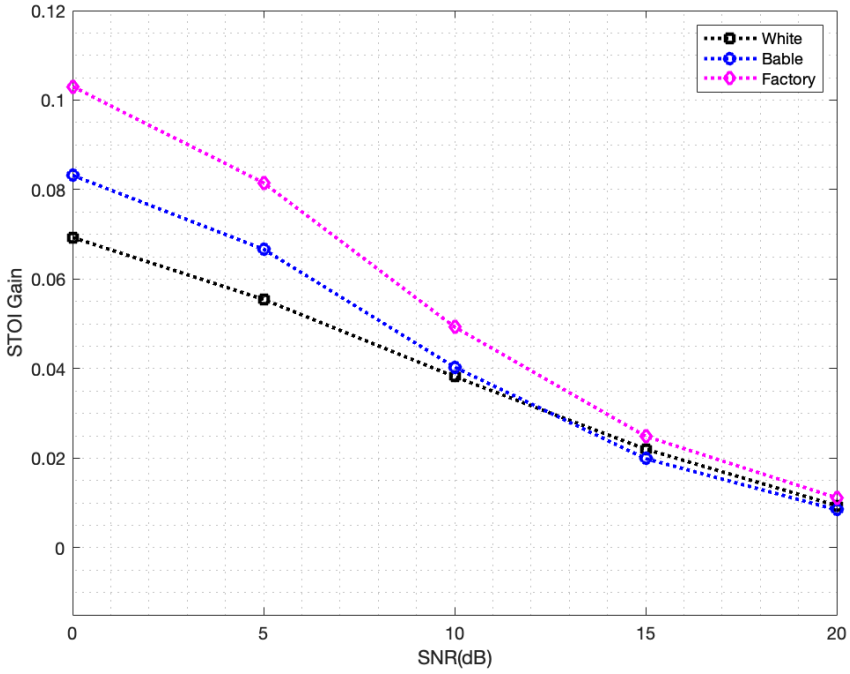

Figure 9: STOI gain for male speech of different types of noise and SNR (NSNet2)

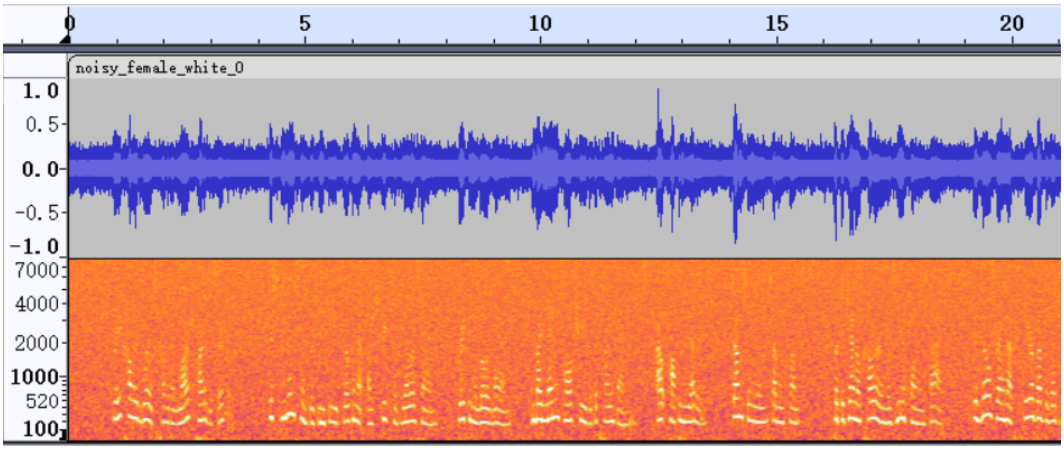

Figure 10: Noisy female speech with white noise of 0 dB

NSNet2, repsectively. For each figure, the upper part is the plot of wave and the lower part is the plot of spectrum.

From the figures we see that the amount of white noise that has been suppressed is ranked as: NSNet2 $>$ MCRA $>$ Spectral Subtraction. Since the NSNet2 model is trained with various kinds of noise data, it achieves the best performance among the three methods. However, from Fig. 13 and Fig.14 we see that the NSNet2 not only suppresses the noise, but also suppresses the speech signal a little bit, which would be verified by listening to the

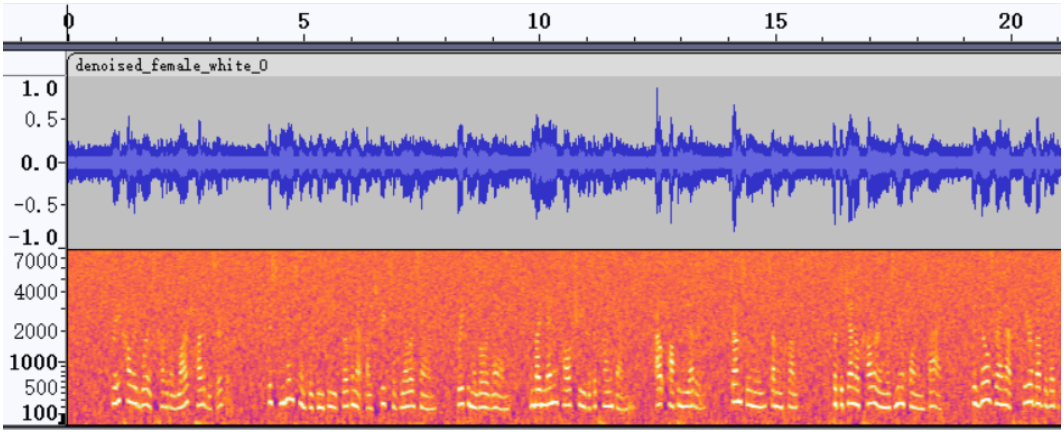

Figure 11: Denoised female speech with white noise of 0 dB using Spectral Subtraction

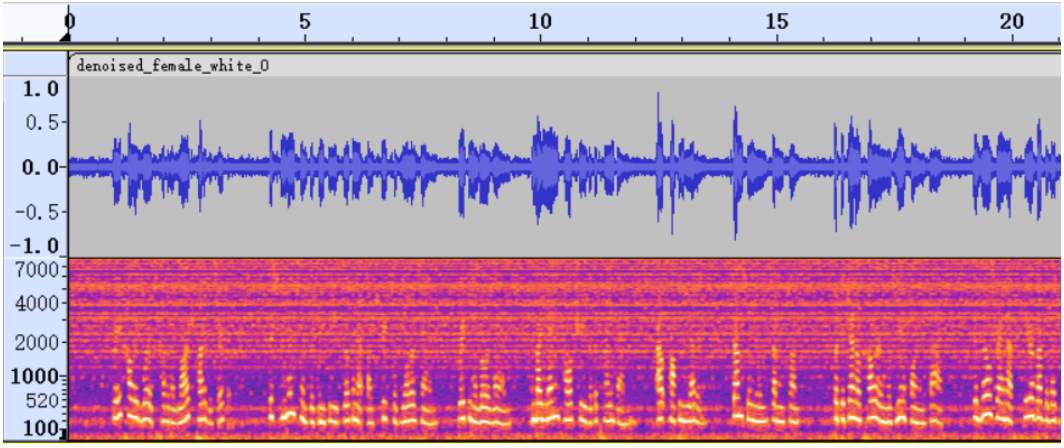

Figure 12: Denoised female speech with white noise of 0 dB using MCRA

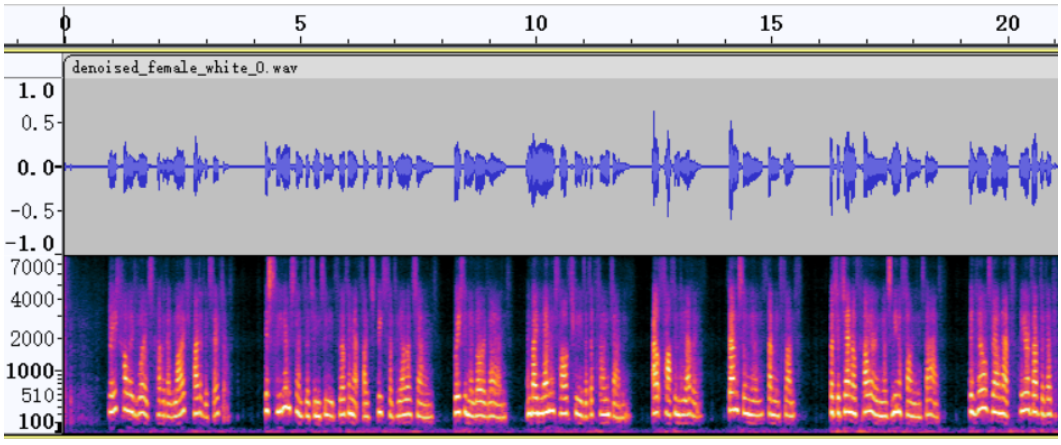

Figure 13: Denoised female speech with white noise of 0 dB using NSNet2

audio waves. By only looking at the wave and spectrum, it is hard to decide whether this

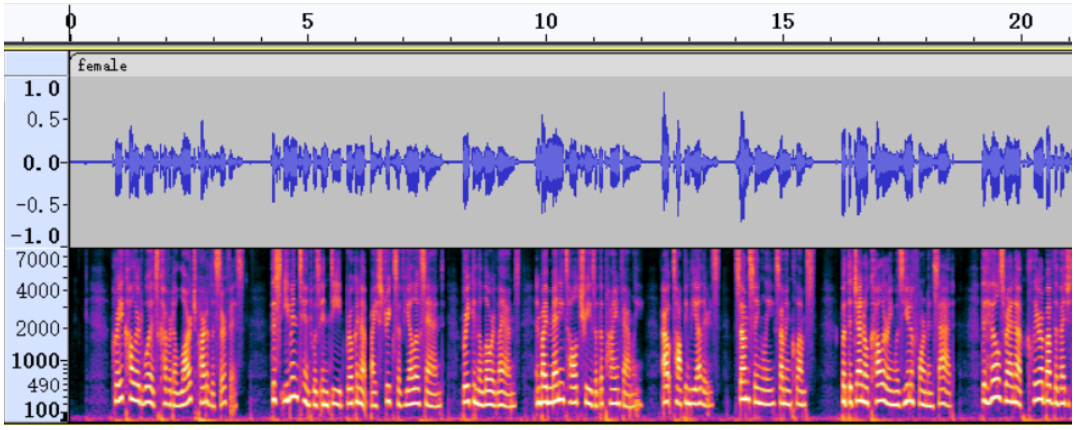

Figure 14: Clean female speech

phenomenon happens when using Spectral Subtraction and MCRA. Thus, STOI score is needed to make a more careful comparison of these three methods.

3.4.2 STOI score

Fig. 15 to 17 present the STOI gain for denoised female speech with white noise, babble noise and factory noise, respectively. Here are some conclusions we would draw from these plots:

1. For three kinds of noise, the amount of STOI gain achieved is ranked as: NSNet2 $>$ MCRA $>$ Spectral Subtraction, which indicate that the NSNet2 model is the best method to perform speech enhancement.

2. For three kinds of noise, the STOI gain using the NSNet2 is monotonically decreasing with SNR.

3. For three kinds of noise, the STOI gain using MCRA reaches its maximum when SNR = 5 dB, and it firstly increases with SNR before 5 dB and then decreases with SNR after 5 dB.

4. For white noise, the STOI gain using Spectral Subtraction reaches its maximum when SNR = 5 dB, while for babble noise and factory noise, the STOI gain using Spectral Subtraction reaches its maximum when SNR = 10 dB.

5. For three kinds of noise and three methods, the STOI gain at SNR = 20 dB is almost zero.

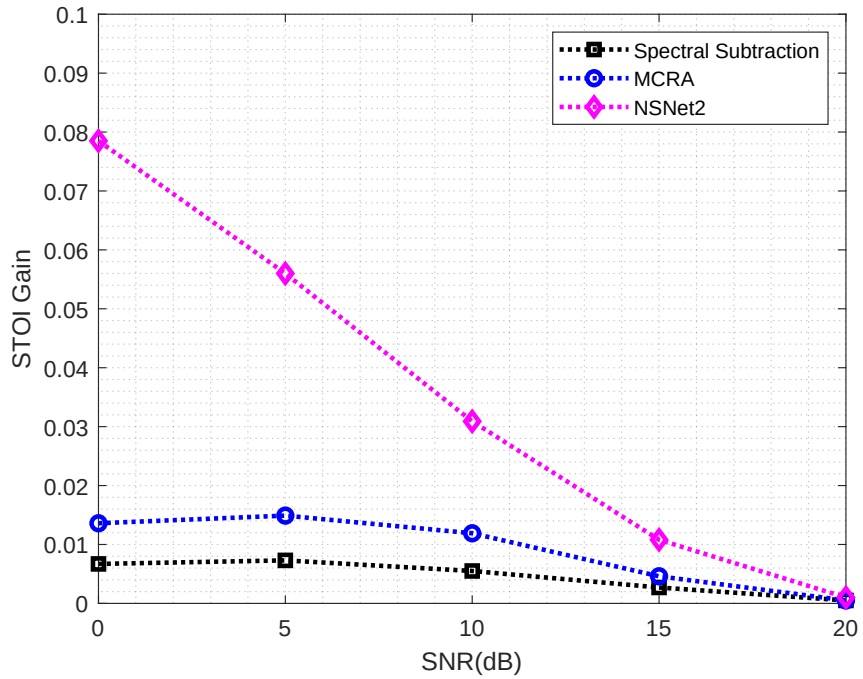

Figure 15: STOI gain for denoised female speech with white noise using three methods

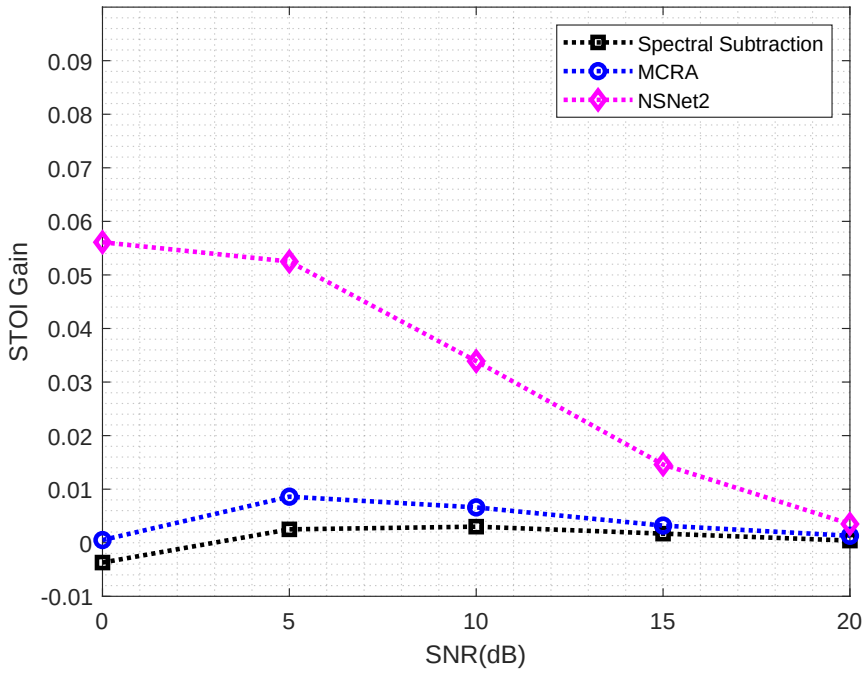

Figure 16: STOI gain for denoised female speech with babble noise using three methods

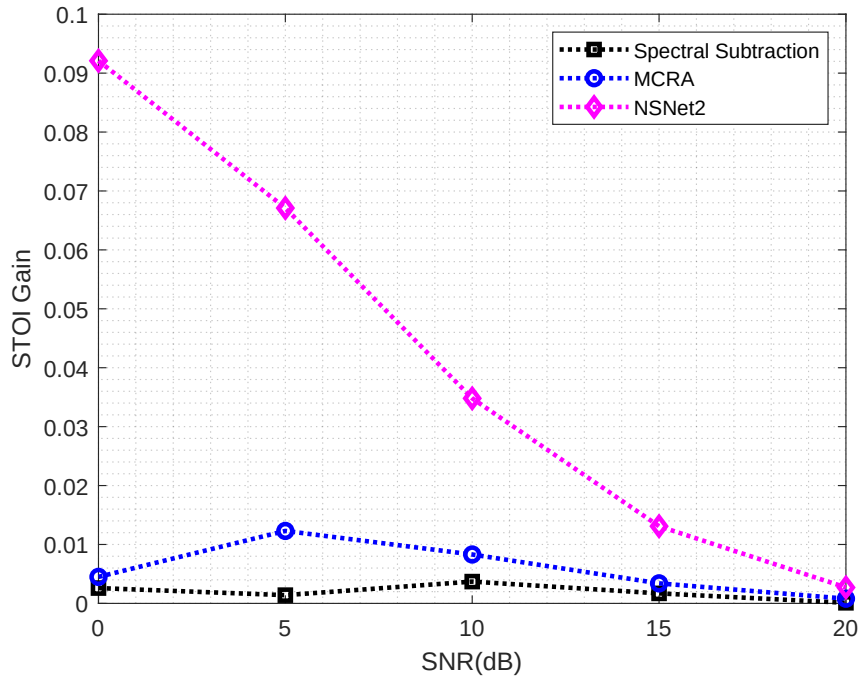

Figure 17: STOI gain for denoised female speech with factory noise using three methods

4 Conclusion

In this project, we have implemented three algorithms for speech enhancement, including two traditional methods called spectral subtraction and MCRA as well as a deep learning-based method called NSNet2. We run our experiments with three types of noise: White, Factory and Babble at different levels of SNR for both male and female speech samples.

Spectral subtraction and MCRA are both computationally efficient speech enhancement approaches and thus they are ideal for embedded real-time systems where there are limited computing resources. As we have discussed in the earlier sections, these approaches do improve the quality of the speech by suppress the background noise. However, we also observe that the STOI gain varies with SNR. Besides, for signals with very low SNR, the STOI gain might even be negative, which indicate that the speech signal is also suppressed.

The deep learning-based approach NSNet2 provides better improvement in the intelligibility of noisy signals. We observe that the STOI gain is high even for speech samples with low SNR across different types of noise. However, training such a model requires a large amount of noise data and time before it learns to suppress noise effectively. Nowadays, with an increasing availability of training data and decreasing cost of computing resources, deep learning-based methods for speech enhancement are gaining more attraction because they have a better performance compared with various traditional methods.

References

[1] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[2] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[3] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235–238, 1977.

[4] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE signal processing letters*, vol. 9, no. 1, pp. 12–15, 2002.

[5] S. Rangachari and P. C. Loizou, "A noise-estimation algorithm for highly non-stationary environments," *Speech communication*, vol. 48, no. 2, pp. 220–231, 2006.

[6] S. Braun and I. Tashev, "Data augmentation and loss normalization for deep noise suppression," 2020. [Online]. Available: https://arxiv.org/abs/2008.06412

[7] H. Dubey, V. Gopal, R. Cutler, S. Matusevych, S. Braun, E. S. Eskimez, M. Thakker, T. Yoshioka, H. Gamper, and R. Aichner, "Icassp 2022 deep noise suppression challenge," in *ICASSP*, 2022.

[8] S. Wisdom, J. R. Hershey, K. W. Wilson, J. Thorpe, M. Chinen, B. Patton, and R. A. Saurous, "Differentiable consistency constraints for improved deep speech enhancement," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 900–904, 2019.

[9] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[10] K. W. Wilson, M. Chinen, J. Thorpe, B. Patton, J. R. Hershey, R. A. Saurous, J. Skoglund, and R. F. Lyon, "Exploring tradeoffs in models for low-latency speech enhancement," *CoRR*, vol. abs/1811.07030, 2018. [Online]. Available: http://arxiv.org/abs/1811.07030

Appendices

.1 Spectral Subtraction

.1.1 Code Implementation

The code to implement the Spectral Subtraction algorithm is obtained from `https://github.com/Gauri-Prajapati/Speech_Enhancement` some key functions are

```
% Calculation of power spectral density
for b = 1:L;
    p = [0.15*abs(Y(b)).^2,zeros(1,k)];
    a = 0.85;
    for  d = 1:k-1
         p(d+1) = a*p(d)+(1-a)*abs(Y(b+d*L)).^2;
    for  e = 1:k-95
         actmin(e) = min(p(e:95+e));
    end
    for  l = k-94:k
         m(l-(k-95)) = min(p(l:k));
    end
    actmin = [actmin(1:k-95),m(1:95)];
    c(1+(b-1)*k:b*k) = actmin(1:k);
end
```

```
% Framing of the data
for r = 1:k
    y = testsignal(1+(r-1)*R:L+(r-1)*R);
    y = y.*h;
    w = fft(y);
    Y(1+(r-1)*L:r*L) = w(1:L);
end
```

```
% Spectral subtraction task
for     t = 1:k
         X = abs(Y).^2;
         S = X(1+(t-1)*L:t*L)-NOISE(1+(t-1)*L:t*L); %
         S = sqrt(S);
         A = Y(1+(t-1)*L:t*L)./abs(Y(1+(t-1)*L:t*L));
         S = S.*A;
         s = ifft(S);
         s = real(s);
         spectruesub_enspeech(1+(t-1)*L/2:L+(t-1)*L/2) = ...
             spectruesub_enspeech(1+(t-1)*L/2:L+(t-1)*L/2)+s;
```

```
            win(1+(t-1)*L/2:L+(t-1)*L/2) = ...
                win(1+(t-1)*L/2:L+(t-1)*L/2)+h;
end
spectruesub_enspeech = spectruesub_enspeech./win;
```

.2 MCRA

.2.1 Code Implementation

The codes for implementing the MCRA algorithm are derived from `https://github.com/Gauri-Prajapati/Speech_Enhancement`, which was pubished on Github for the public to use. Here are annotations of the key lines in `improved_mcra_test.m`.

```
gamma=noisy_psd./noise_cap;
%To compute a posteriori SNR.
eps_cap=alpha*(GH1.^2).*gamma_old+(1-alpha)*max(gamma-1,0);
%To compute a priori SNR.
S=alpha_s*S+(1-alpha_s)*Sf;
%For computing S(k,l).
I(index)=1;
%To compute I(k,l).
S_tild=alpha_s*S_tild+(1-alpha_s)*Sf_tild;
%For updating S_tild.
alpha_d_tild=alpha_d+(1-alpha_d)*p;
%To update the time and frequency dependent smoothing factor.
noise_tild=alpha_d_tild.*noise_tild+(1-alpha_d_tild).*noisy_psd;
%To update noise estimate.
noise_cap=beta*noise_tild;
%To update noise estimate for correction factor.
```

.3 NSNet2

.3.1 Dataset and Experimental Setup

For training, the CHIME-2 WSJ-20k dataset has been used, which is currently, while only being of medium size, the only realistic self-contained public dataset including matching reverberant speech and noise conditions. The dataset contains 7138, 2418, and 1998 utterances for training, validation and testing, respectively. The utterances are reverberant using binaural room impulse responses, and noise from the same rooms was added with SNRs in the range of -6 to 9 dB in the validation and test sets.

.3.2 Datset Augmentation

Especially for small and medium-scale datasets, augmentation is a powerful tool to improve the results. In the case of supervised speech enhancement training, where the actual noisy

audio training data is generated synthetically by mixing speech and noise, there are some augmentation steps, which are essential to mimic effects on data encountered in the wild.

The augmentation pipeline is shown in Fig.18. Before mixing speech with noise, random biquad filters are applied to each noise sequence and speech sequences separately to mimic different acoustic transmission effects. From these signals, active speech and noise levels are computed using a level threshold-based voice activity detector (VAD). Speech and noise sequences are then mixed with a given SNR on-the-fly during training.

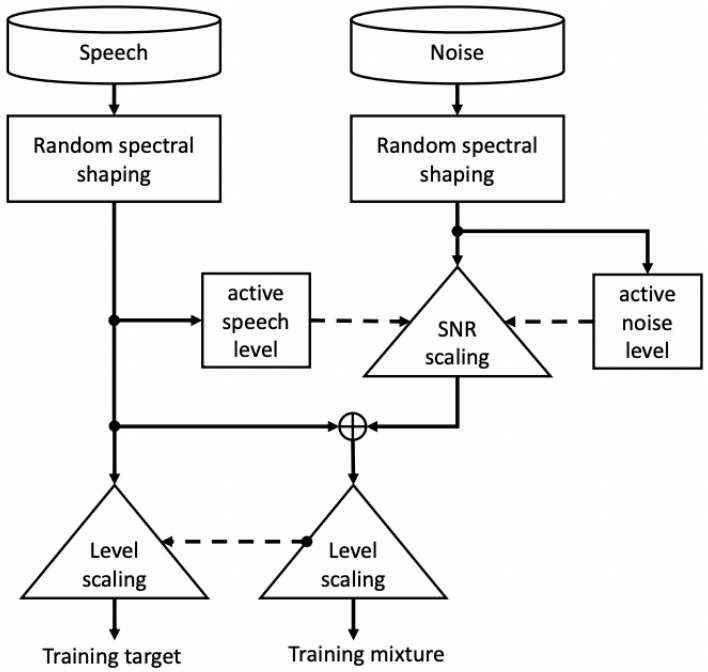

Figure 18: On-the-fly training augmented data generation.

.3.3 Code description

Individual Function headings are mentioned below along with their descriptions.

```
def calcFeat(Spec, cfg):
    """compute spectral features"""
```

`Snippet1:calcFeat` function to compute spectral feature extraction.

```
def calcSpec(y, params, channel=None):
    """compute complex spectrum from audio file"""
```

`Snippet2:calcSpec` function to compute the spectrum of the `.wav` files.

```
def spec2sig(Spec, params):
    """convert spectrum to time signal"""
```

Snippet3:spec2sig function to convert the spectrum back into a time-domain audio signal.

```
def stft(x, N_fft, win, N_hop, nodelay=True):
        """
        short-time Fourier transform
                x = time domain signal [samples x channels]
                N_fft = FFT size (samples)
                win = wiindow,  len(win) <= N_fft
                N_hop = hop size (samples)
                nodelay = [True,False]: do not introduce delay (visible
 ↪  windowing effects in first frames)
        """
```

Snippet4:stft function to compute the a windowed Short Time Fourier Transform.

```
class NSnet2Enhancer(object):
    """NSnet2 enhancer class."""
    #
    #
    #
    #
     def enhance(self, x):
```

Snippet5:NSnet2Enchancer class to compute the denoising of the noisy signal x.

```
model = '$CHECKPOINT DIRECTORY$'
input = '$NOISY .WAV FILE$'
output = '$OUTPUT DIRECTORY$'
```

Snippet6: Input Parameters. The model takes the input of the pretrainned .onnx checkpoint.