A New Framework for CNN-Based Speech Enhancement in the Time Domain

Ashutosh Pandey, DeLiang Wang

Supervisor：Prof. 陳自強 Oscal T.-C. Chen
Student：蔡承宏(Tsai Cheng Hong)
Student ID：609415074

Outline

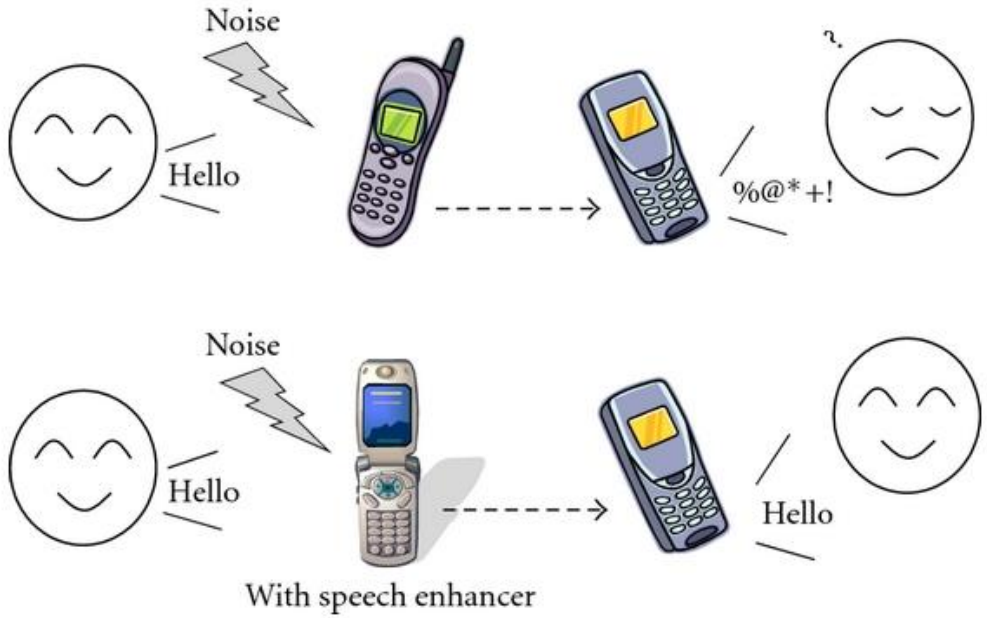
- Abstract
- Introduction
- Frequency Domain Loss Function
- Model Architecture
- Invalid Short-Time Fourier Transform
- Experimental Settings
- Results and Discussions & Conclusion
- Reference

Abstract

- This paper proposes a new learning mechanism for a fully convolutional neural network (CNN) to address speech enhancement in the time domain.
- The CNN takes as input the time frames of noisy utterance and outputs the time frames of the enhanced utterance.
- At the training time, we converts the time domain to the frequency domain. This conversion corresponds to simple matrix multiplication, and is hence differentiable implying that a frequency domain loss can be used for training in the time domain.
- We use mean absolute error loss between the enhanced short-time Fourier transform (STFT) magnitude and the clean STFT magnitude to train the CNN.
- the model can exploit the domain knowledge of converting a signal to the frequency domain for analysis. Moreover, this approach avoids the invalid STFT problem since the proposed CNN operates in the time domain.
- Experimental results demonstrate that the proposed method substantially outperforms the other methods of speech enhancement.

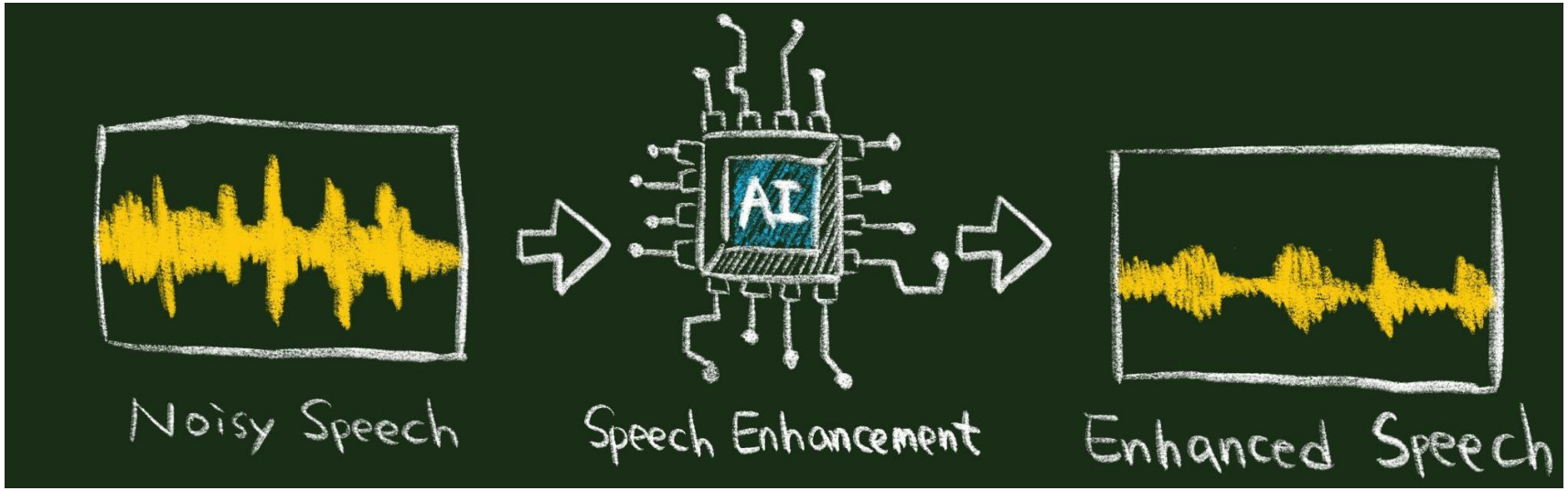
Introduction

- Speech enhancement is the task of removing or attenuating additive noise from a speech signal.
- Speech enhancement is employed as a preprocessor in many applications such as robust automatic speech recognition, teleconferencing and hearing aids design



Introduction

- The purpose of monaural (single-channel) speech enhancement is to provide a versatile and cost-efficient approach to the problem that utilizes recordings from only a single microphone. Single-channel speech enhancement is considered a very challenging problem especially at low signal-to-noise ratios (SNRs). This study focuses on single-channel speech enhancement in the time domain.



Introduction

- Traditional monaural speech enhancement approaches include statistical enhancement methods and computational auditory scene analysis . In the last few years, supervised methods for speech enhancement using deep neural networks (DNNs) have become the mainstream. Among the most popular deep learning methods are denoising autoencoders, feedforward neural networks and CNNs.

Traditional method:

a. statistical enhancement methods : Increase contrast.
b. computational auditory scene analysis : Several sounds separate to one sound.

Introduction

- Most frequently employed methods for supervised speech enhancement use T-F masking or spectral mapping [3]. Both of these approaches reconstruct the speech signal in the time domain from the frequency domain using the phase of the noisy signal.
- Both of these approaches reconstruct the speech signal in the time domain from the frequency domain using the phase of the noisy signal. It means that the learning machine learns a mapping in the frequency domain but the task of going from the frequency domain to the time domain is not subject to the learning process.

a. T-F masking : When the noise is within the range, it has no effect on hearing.
Application : voice compression communications.
b. Spectral mapping : Make ambiguous time-frequency data points are repositioned and clearly.

Introduction

- We design a fully convolutional neural network that takes as input the noisy speech signal in the time domain and outputs the enhanced speech signal in the time domain.
- One way to train this network is to minimize the mean squared error (MSE) or the MAE loss between the clean speech signal and the enhanced speech signal.
- Our experiments show that, using a time domain loss, some of the phonetic information in the estimated speech is distorted probably because the underlying phones are difficult to distinguish from the background noise.
- We believe that it is important to use a frequency domain loss, which can discriminate speech sounds from nonspeech noises and produce speech with high quality.

Introduction

- A model is employed in the time domain and trained using a loss function in the frequency domain, so the generated signal is always a valid signal and we do not need to use the phase of the noisy signal. The neural network learns a phase structure itself in the process of optimizing the proposed loss.
- Other researchers have explored speech enhancement in the time domain using deep learning.Other authors explore CNNs for speech enhancement and claim that fully connected layers inside a DNN are not suitable for the time domain enhancement and instead propose to use a fully-convolutional neural network.

Frequency Domain Loss Function

- Given a real-valued vector xt of size N in the time domain, we can convert it to the frequency domain by multiplying it with a complex-valued discrete Fourier transform (DFT) matrix D using the following equation $\boldsymbol{x}_f = \boldsymbol{D}\boldsymbol{x}_t$
- where xf is the DFT of xt and D is of size N × N. Since xt is real-valued, the relation in (1) can be rewritten as $\boldsymbol{x}_f = (\boldsymbol{D}_r + i\boldsymbol{D}_i)\boldsymbol{x}_t = \boldsymbol{D}_r\boldsymbol{x}_t + i\boldsymbol{D}_i\boldsymbol{x}_t$
- Where Dr and Di are real-valued matrices formed by taking the element-wise real and imaginary part of D and i denotes the imaginary unit. This relation can be separated into two Equations involving only real-valued vectors as given in the following Equation.

$$\boldsymbol{x}_{f_r} = \boldsymbol{D}_r\boldsymbol{x}_t$$
$$\boldsymbol{x}_{f_i} = \boldsymbol{D}_i\boldsymbol{x}_t$$

- Here, xfr and xfi are real-valued vectors formed by taking element-wise real and imaginary part of xf. A frequency domain loss can thus be defined using xfr and xfi. One such loss defined as the average of the MSE losses on the real and imaginary part of xf is:

$$L(\hat{\boldsymbol{x}}_f, \boldsymbol{x}_f) = \frac{1}{N}\sum_{n=1}^{N}((\hat{x}_{f_r}(n) - x_{f_r}(n))^2 + (\hat{x}_{f_i}(n) - x_{f_i}(n))^2)$$

Frequency Domain Loss Function

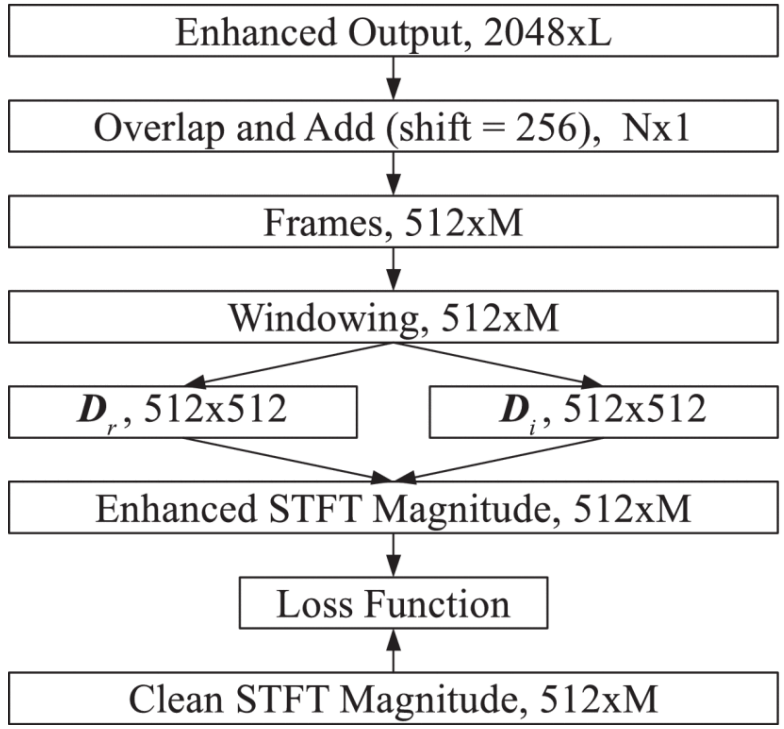
- where x^f is an estimate of xf. x(n) denotes the nth component of x. It should be noted that this loss function has both magnitude and phase because it uses the real as well as the imaginary part. However, we find that using both the magnitude and the phase does not give as good performance as using only the magnitude. So, we use the following loss function defined using only the magnitudes.

$$L(\hat{\boldsymbol{x}}_f, \boldsymbol{x}_f) = \frac{1}{N}\sum_{n=1}^{N} |(|\hat{x}_{f_r}(n)| + |\hat{x}_{f_i}(n)|) - (|x_{f_r}(n)| + |x_{f_i}(n)|)|$$

- In the present study we have confirmed that the MAE loss performs better than the MSE loss for objective intelligibility and quality.

Frequency Domain Loss Function

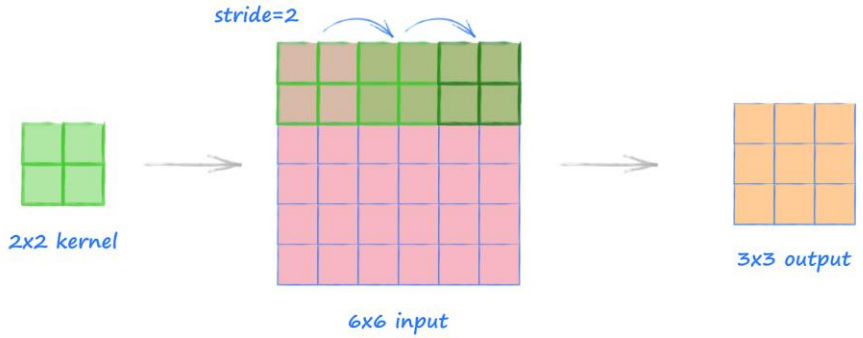
a. model operates on the frame size of 2048 samples
b. it takes as input a frame of duration 128 ms with the sampling frequency of 16 kHz, and outputs a frame of the same length.
c. All the enhanced frames of an utterance at the network output are combined using the overlap-and -add (OLA) method to obtain the enhanced utterance.
d. A frame shift of 256 samples is used for OLA. The enhanced utterance is then divided into frames of size 512. The obtained frames are multiplied by the Hamming window and then separately with two matrices Dr and Di, each of size 512×512.
e. The matrix multiplication gives the real and imaginary part of the STFT. Next, the real and imaginary part of the STFT are combined to get the STFT magnitude. The computed STFT magnitude is compared with the clean STFT magnitude to obtain a frequency domain loss.



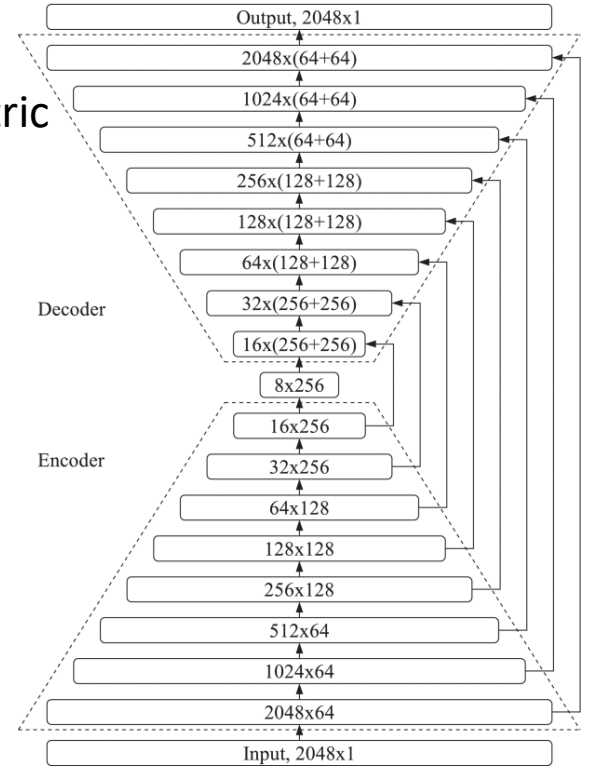
shows a schematic diagram for computing a frequency domain loss from enhanced time domain frames.

Model Architecture

- We use a fully convolutional neural network that is comprised of a series of convolutional and deconvolutional layers.
- A deconvolution layer, also known as transposed convolution, is a convolution meant to increase the size at the output,make the input & output is 2N.
- The final output of the encoder is of size 8 with 256 channels.
- Each layer in the network uses the activation of parametric ReLU non-linearity



CNN Stride = 2



autoencoder based FCN

Invalid Short-Time Fourier Transform

- The STFT of a signal is obtained by taking the DFT of overlapped frames of a signal. The overlap between consecutive frames causes the adjacent frames to have common samples at the boundary of frames. This correlation between adjacent frames appears in the frequency domain as well and results in a certain relationship between the STFT magnitude and the STFT phase.
- ISTFT denotes inverse STFT, m frame number, and k is frequency index. An STFT obtained by taking the STFT of a real signal in the time domain is always a valid STFT. It means that given a real signal x(t) in the time domain, the following relations will always hold.

$$\mathrm{ISTFT}(\mathrm{STFT}(x(t))) = x(t)$$
$$\mathrm{STFT}(\mathrm{ISTFT}(\mathrm{STFT}(x(t)))) = \mathrm{STFT}(x(t))$$

- The proposed framework can be thought of as a supervised way of resolving the invalid STFT problem by a CNN, which produces a speech signal in the time domain but is trained with a loss function which minimizes the distance measured in terms of the STFT magnitudes.

Experimental Settings

Datasets

- we evaluate and analyze the proposed framework on the TIMIT dataset which consists of utterances from many male and female speakers.
- We use 2000 randomly chosen utterances from the TIMIT training set as the training utterances, 192 utterances is used as the test set.
- Five noise-dependent models are trained on the following five noises: babble, factory1, oproom, engine, and speech-shaped noise (SSN).
- All noises are around 4 minutes long

Experimental Settings

System Setup

- All the utterances are resampled to 16 kHz.
- The noisy and the clean utterances are normalized to the value range [−1, 1] and frames are extracted from the normalized utterances
- A filter size of 11 is used.
- All the weights in CNNs are initialized using the Xavier initializer with normally distributed random initialization.
- Batch size of 4 utterances.
- Learning rate is set to 0.0002.
- Our model has around 6.4 million parameters.

※**Xavier initialization** is the weights initialization technique that tries to make the variance of the outputs of a layer to be equal to the variance of its inputs.

Experimental Settings

Baseline Models

- We use three baseline models:

1. We train a DNN model using the MAE loss to estimate the ideal ratio mask, This model is a 3-layered fully connected DNN that takes as input the noisy STFT magnitudes of five consecutive frames (centered at the current frame) concatenated together and outputs the IRM of the corresponding five frames together.
2. The second baseline is the SEGAN model,it's GAN's generator. We train two versions of this model,The first is trained using both the loss functions, adversarial loss and the MAE loss as in the original paper. We call this model SEGAN. The second is trained using only the loss on time domain samples. We call this model SEGAN-T in our experiments.
3. The third is gated residual network (GRN) model is a 62-layer deep fully convolutional network with residual connections. It takes as input the spectrogram of the whole utterance at once and outputs the phase-sensitive mask (PSM) of the whole utterance.

Experimental Settings

Evaluation Metrics and Comparisons

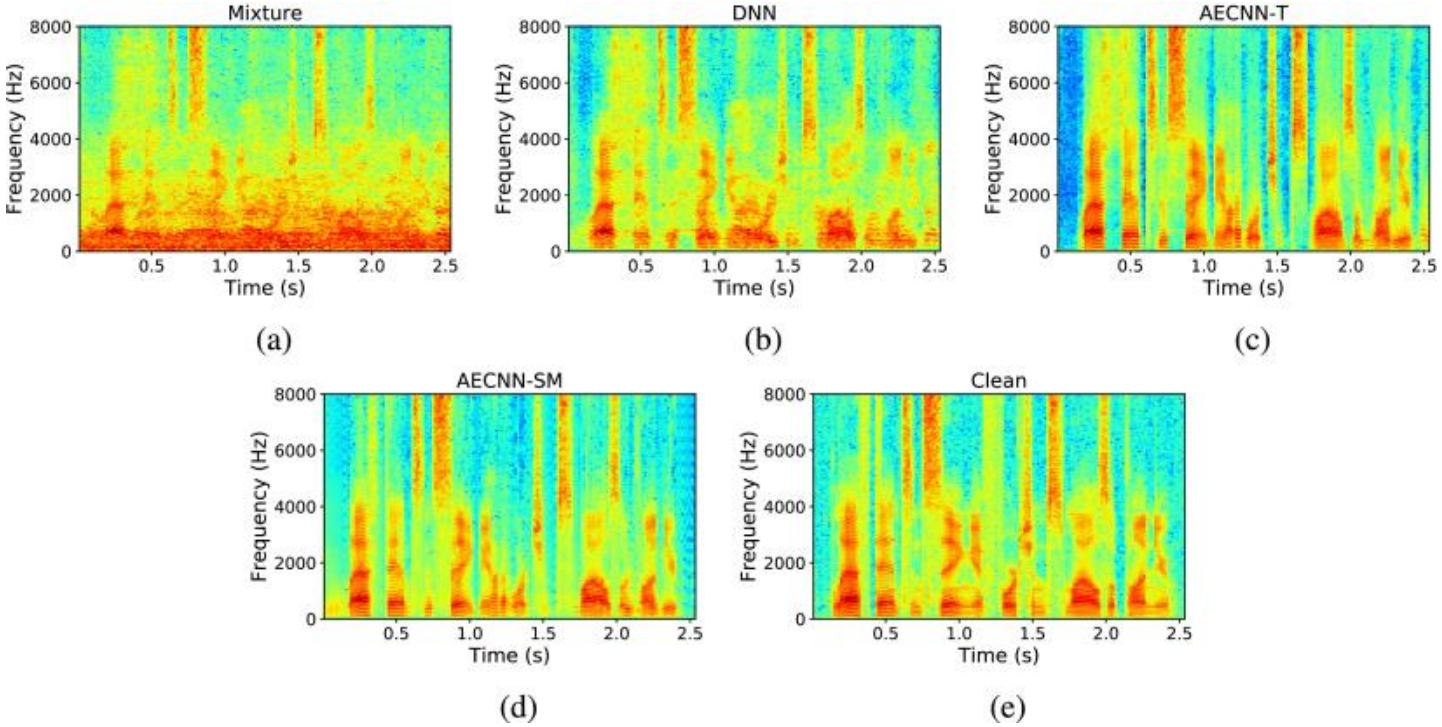
- In our experiments, models are compared using short-term objective intelligibility (STOI), perceptual evaluation of speech quality (PESQ), and scale-invariant signal-to-distortion ratio (SI-SDR) scores, all of which represent the standard metrics for speech enhancement
- STOI has a typical value range from 0 to 1, which can be roughly interpreted as percent correct. PESQ values range from −0.5 to 4.5.
- The STOI, PESQ and SI-SDR scores are compared at the SNRs of −5, 0, and 5 dB.

Results and Discussions

- We observe that the models trained with a loss with phase perform better using MSE whereas the models trained with a loss without phase perform better using MAE.
- The SEGAN-T, AECNN-T, and AECNN-RI, have better STOI and SI-SDR scores with MAE but significantly worse PESQ
- AECNN-SM1 and AECNN-SM2 produce better scores with the MAE loss.
- AECNN-SM1 and AECNN-SM2 have similar STOI and PESQ scores but AECNN-SM1 is consistently better in terms of SI-SDR.

SNR		-5 dB			0 dB			5 dB		
Metric		STOI (%)	PESQ	SI-SDR	STOI (%)	PESQ	SI-SDR	STOI (%)	PESQ	SI-SDR
Mixture		56.5	1.41	-4.8	68.2	1.72	0.1	78.9	2.05	5.1
DNN		69.7	1.88	2.8	80.4	2.34	7.5	87.7	2.74	11.9
SEGAN		76.8	1.77	7.1	86.0	2.28	10.3	90.2	2.60	12.5
SEGAN-T	MAE	77.7	1.73	7.6	87.2	2.22	11.2	91.3	2.57	13.5
	MSE	77.9	2.01	7.7	87.3	2.46	11.2	91.5	2.78	13.6
AECNN-T	MAE	78.9	1.88	8.2	88.2	2.41	11.7	92.3	2.80	**14.3**
	MSE	78.6	2.00	8.0	88.0	2.60	11.5	92.2	2.90	13.9
AECNN-RI	MAE	80.0	1.92	**8.6**	**89.0**	2.48	**12.0**	**92.8**	2.86	**14.3**
	MSE	78.6	2.00	8.1	88.1	2.56	11.6	92.2	2.90	14.1
AECNN-SM1	MAE	**80.3**	**2.20**	8.0	**89.0**	2.68	11.4	**92.8**	3.01	13.7
	MSE	78.9	**2.20**	7.5	88.0	2.60	11.1	92.2	2.90	13.6
AECNN-SM2	MAE	80.2	**2.20**	7.8	**89.0**	**2.70**	10.8	92.7	**3.02**	12.7
	MSE	78.9	2.20	7.3	88.1	2.60	10.7	92.1	2.9	12.9

Results and Discussions



Concluding Remarks

- In this paper, proposed a novel approach to train a fully convolutional neural network for speech enhancement in the time domain.
- The key idea is to use a frequency domain loss to train the CNN.
- We have investigated different types of loss function in the frequency domain. Our main observation is that frequency domain loss is better than a time domain loss.
- Using a frequency domain loss helps to improve objective quality and intelligibility
- In all the cases, the proposed method substantially outperforms the current state-of-the-art methods.

Reference

A. Pandey and D. Wang, "A New Framework for CNN-Based Speech Enhancement in the Time Domain," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 7, pp. 1179-1188, July 2019, doi: 10.1109/TASLP.2019.2913512.

END